TO MY FAMILY

Shelly, Catherine, and Max
# Brief Contents

1. A Survey of Computer Graphics 1
2. Computer Graphics Hardware 9
4. Graphics Output Primitives 51
5. Attributes of Graphics Primitives 105
7. Two-Dimensional Geometric Transformations 195
8. Two-Dimensional Viewing 233
9. Three-Dimensional Geometric Transformations 279
10. Three-Dimensional Viewing 307
11. Hierarchical Modeling 359
12. Computer Animation 369
13. Three-Dimensional Object Representations 393
14. Spline Representations 411
15. Other Three-Dimensional Object Representations 467
16. Visible-Surface Detection Methods 479
17. Illumination Models and Surface-Rendering Methods 507
18. Texturing and Surface-Detail Methods 555
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>Color Models and Color Applications</td>
<td>579</td>
</tr>
<tr>
<td>20</td>
<td>Interactive Input Methods and Graphical User Interfaces</td>
<td>599</td>
</tr>
<tr>
<td>21</td>
<td>Global Illumination</td>
<td>639</td>
</tr>
<tr>
<td>22</td>
<td>Programmable Shaders</td>
<td>665</td>
</tr>
<tr>
<td>23</td>
<td>Algorithmic Modeling</td>
<td>695</td>
</tr>
<tr>
<td>24</td>
<td>Visualization of Data Sets</td>
<td>725</td>
</tr>
</tbody>
</table>
Contents

Preface xxi
About the Authors xxvi

1 A Survey of Computer Graphics 1
  1-1 Graphs and Charts 2
  1-2 Computer-Aided Design 2
  1-3 Virtual-Reality Environments 4
  1-4 Data Visualizations 4
  1-5 Education and Training 5
  1-6 Computer Art 5
  1-7 Entertainment 6
  1-8 Image Processing 7
  1-9 Graphical User Interfaces 7
  1-10 Summary 8

2 Computer Graphics Hardware 9
  2-1 Video Display Devices 10
     Refresh Cathode-Ray Tubes 10
     Raster-Scan Displays 13
     Random-Scan Displays 15
     Color CRT Monitors 16
     Flat-Panel Displays 18
     Three-Dimensional Viewing Devices 20
     Stereoscopic and Virtual-Reality Systems 21
  2-2 Raster-Scan Systems 22
     Video Controller 22
     Raster-Scan Display Processor 24
  2-3 Graphics Workstations and Viewing Systems 25
  2-4 Input Devices 26
     Keyboards, Button Boxes, and Dials 26
     Mouse Devices 26
     Trackballs and Spaceballs 27
     Joysticks 27

3 Computer Graphics Software 35
  3-1 Coordinate Representations 36
  3-2 Graphics Functions 37
  3-3 Software Standards 38
  3-4 Other Graphics Packages 39
  3-5 Introduction to OpenGL 40
     Basic OpenGL Syntax 40
     Related Libraries 40
     Header Files 41
     Display-Window Management Using GLUT 42
     A Complete OpenGL Program 43
     Error Handling in OpenGL 47
  3-6 Summary 48

4 Graphics Output Primitives 51
  4-1 Coordinate Reference Frames 52
     Screen Coordinates 52
     Absolute and Relative Coordinate Specifications 53
  4-2 Specifying a Two-Dimensional World-Coordinate Reference Frame in OpenGL 54

6-3 Setting Frame-Buffer Values 146
6-4 Circle-Generating Algorithms 147
   Properties of Circles 147
   Midpoint Circle Algorithm 149
6-5 Ellipse-Generating Algorithms 153
   Properties of Ellipses 153
   Midpoint Ellipse Algorithm 154
6-6 Other Curves 160
   Conic Sections 161
   Polynomials and Spline Curves 162
6-7 Parallel Curve Algorithms 163
6-8 Pixel Addressing and Object
   Geometry 163
   Screen Grid Coordinates 164
   Maintaining Geometric Properties
   of Displayed Objects 164
6-9 Attribute Implementations for
   Straight-Line Segments and Curves 166
   Line Width 166
   Line Style 168
   Pen and Brush Options 169
   Curve Attributes 170
6-10 General Scan-Line Polygon-Fill
    Algorithm 171
6-11 Scan-Line Fill of Convex Polygons 175
6-12 Scan-Line Fill for Regions with Curved
    Boundaries 176
6-13 Fill Methods for Areas with Irregular
    Boundaries 176
    Boundary-Fill Algorithm 176
    Flood-Fill Algorithm 180
6-14 Implementation Methods for Fill
    Styles 180
    Fill Styles 181
    Color-Blended Fill Regions 181
6-15 Implementation Methods for
    Antialiasing 183
    Supersampling Straight-Line Segments 184
    Subpixel Weighting Masks 186
    Area Sampling Straight-Line Segments 186
    Filtering Techniques 186
    Pixel Phasing 186
    Compensating for Line-Intensity
    Differences 187
    Antialiasing Area Boundaries 188
6-16 Summary 190

7 Two-Dimensional Geometric
   Transformations 195
7-1 Basic Two-Dimensional Geometric
   Transformations 196
   Two-Dimensional Translation 196
   Two-Dimensional Rotation 198
   Two-Dimensional Scaling 200
7-2 Matrix Representations and
   Homogeneous Coordinates 201
   Homogeneous Coordinates 202
   Two-Dimensional Translation Matrix 203
   Two-Dimensional Rotation Matrix 203
   Two-Dimensional Scaling Matrix 203
7-3 Inverse Transformations 204
7-4 Two-Dimensional Composite
   Transformations 204
   Composite Two-Dimensional
   Translations 205
   Composite Two-Dimensional Rotations 205
   Composite Two-Dimensional Scalings 205
   General Two-Dimensional Pivot-Point
   Rotation 206
   General Two-Dimensional Fixed-Point
   Scaling 206
   General Two-Dimensional Scaling
   Directions 207
   Matrix Concatenation Properties 208
   General Two-Dimensional Composite
   Transformations and Computational
   Efficiency 209
   Two-Dimensional Rigid-Body
   Transformation 210
   Constructing Two-Dimensional Rotation
   Matrices 211
   Two-Dimensional Composite-Matrix
   Programming Example 211
7-5 Other Two-Dimensional
   Transformations 216
   Reflection 216
   Shear 218
7-6 Raster Methods for Geometric
   Transformations 220
7-7 OpenGL Raster Transformations 221
7-8 Transformations between Two-
   Dimensional Coordinate Systems 222
7-9 OpenGL Functions for Two-Dimensional Geometric Transformations 224
Basic OpenGL Geometric Transformations 224
OpenGL Matrix Operations 226
7-10 OpenGL Geometric-Transformation Programming Examples 228
7-11 Summary 229

8 Two-Dimensional Viewing 233
8-1 The Two-Dimensional Viewing Pipeline 234
8-2 The Clipping Window 235
Viewing-Coordinate Clipping Window 236
World-Coordinate Clipping Window 236
8-3 Normalization and Viewport Transformations 237
Mapping the Clipping Window into a Normalized Viewport 237
Mapping the Clipping Window into a Normalized Square 239
Display of Character Strings 241
Split-Screen Effects and Multiple Output Devices 241
8-4 OpenGL Two-Dimensional Viewing Functions 241
OpenGL Projection Mode 241
GLU Clipping-Window Function 242
OpenGL Viewport Function 242
Creating a GLUT Display Window 243
Setting the GLUT Display-Window Mode and Color 244
GLUT Display-Window Identifier 244
Deleting a GLUT Display Window 244
Current GLUT Display Window 244
Relocating and Resizing a GLUT Display Window 245
Managing Multiple GLUT Display Windows 245
GLUT Subwindows 246
Selecting a Display-Window Screen-Cursor Shape 246
Viewing Graphics Objects in a GLUT Display Window 247
Executing the Application Program 247

Other GLUT Functions 248
OpenGL Two-Dimensional Viewing Program Example 248
8-5 Clipping Algorithms 250
8-6 Two-Dimensional Point Clipping 250
8-7 Two-Dimensional Line Clipping 251
Cohen-Sutherland Line Clipping 252
Liang-Barsky Line Clipping 257
Nicholl-Lee-Nicholl Line Clipping 260
Line Clipping Using Nonrectangular Polygon Clip Windows 262
Line Clipping Using Nonlinear Clipping-Window Boundaries 262
8-8 Polygon Fill-Area Clipping 263
Sutherland-Hodgman Polygon Clipping 265
Weiler-Atherton Polygon Clipping 269
Polygon Clipping Using Nonrectangular Polygon Clip Windows 271
Polygon Clipping Using Nonlinear Clipping-Window Boundaries 272
8-9 Curve Clipping 272
8-10 Text Clipping 273
8-11 Summary 274

9 Three-Dimensional Geometric Transformations 279
9-1 Three-Dimensional Translation 280
9-2 Three-Dimensional Rotation 281
Three-Dimensional Coordinate-Axis Rotations 282
General Three-Dimensional Rotations 284
Quaternion Methods for Three-Dimensional Rotations 289
9-3 Three-Dimensional Scaling 293
9-4 Composite Three-Dimensional Transformations 295
9-5 Other Three-Dimensional Transformations 298
Three-Dimensional Reflections 298
Three-Dimensional Shears 299
9-6 Transformations between Three-Dimensional Coordinate Systems 299
9-7 Affine Transformations 300
Three-Dimensional Viewing 307

10-1 Overview of Three-Dimensional Viewing Concepts 308
Viewing a Three-Dimensional Scene 308
Projections 308
Depth Cueing 309
Identifying Visible Lines and Surfaces 309
Surface Rendering 309
Exploded and Cutaway Views 309
Three-Dimensional and Stereoscopic Viewing 310

10-2 The Three-Dimensional Viewing Pipeline 310

10-3 Three-Dimensional Viewing-Coordinate Parameters 312
The View-Plane Normal Vector 312
The View-Up Vector 313
The uvn Viewing-Coordinate Reference Frame 313
Generating Three-Dimensional Viewing Effects 314

10-4 Transformation from World to Viewing Coordinates 314

10-5 Projection Transformations 316

10-6 Orthogonal Projections 316
Axonometric and Isometric Orthogonal Projections 317
Orthogonal Projection Coordinates 317
Clipping Window and Orthogonal-Projection View Volume 318
Normalization Transformation for an Orthogonal Projection 320

10-7 Oblique Parallel Projections 321
Oblique Parallel Projections in Drafting and Design 322
Cavalier and Cabinet Oblique Parallel Projections 323
Oblique Parallel-Projection Vector 324
Clipping Window and Oblique Parallel-Projection View Volume 325
Oblique Parallel-Projection Transformation Matrix 325
Normalization Transformation for an Oblique Parallel Projection 326

10-8 Perspective Projections 327
Perspective-Projection Transformation Coordinates 327
Perspective-Projection Equations: Special Cases 328
Vanishing Points for Perspective Projections 330
Perspective-Projection View Volume 331
Perspective-Projection Transformation Matrix 332
Symmetric Perspective-Projection Frustum 333
Oblique Perspective-Projection Frustum 335
Normalized Perspective-Projection Transformation Coordinates 338

10-9 The Viewport Transformation and Three-Dimensional Screen Coordinates 341

10-10 OpenGL Three-Dimensional Viewing Functions 341
OpenGL Viewing-Transformation Function 342
OpenGL Orthogonal-Projection Function 342
OpenGL Symmetric Perspective-Projection Function 344
OpenGL General Perspective-Projection Function 344
OpenGL Viewports and Display Windows 345
OpenGL Three-Dimensional Viewing Program Example 345

10-11 Three-Dimensional Clipping Algorithms 347
Clipping in Three-Dimensional Homogeneous Coordinates 348
Three-Dimensional Region Codes 349
Three-Dimensional Point and Line Clipping 350
Three-Dimensional Polygon Clipping 352
Three-Dimensional Curve Clipping 353
Arbitrary Clipping Planes 353
Three-Dimensional Object Representations 393

13-1 Polyhedra 394
13-2 OpenGL Polyhedron Functions 394
  OpenGL Polygon Fill-Area Functions 394
  GLUT Regular Polyhedron Functions 394
  Example GLUT Polyhedron Program 396
13-3 Curved Surfaces 397
13-4 Quadric Surfaces 398
  Sphere 398
  Ellipsoid 398
  Torus 399
13-5 Superquadrics 400
  Superellipse 400
  Superellipsoid 401
13-6 OpenGL Quadric-Surface and Cubic-Surface Functions 401
  GLUT Quadric-Surface Functions 401
  GLUT Cubic-Surface Teapot Function 402
  GLU Quadric-Surface Functions 403
  Example Program Using GLUT and GLU Quadric-Surface Functions 405
13-7 Summary 407

Spline Representations 411

14-1 Interpolation and Approximation Splines 412
14-2 Parametric Continuity Conditions 413
14-3 Geometric Continuity Conditions 414
14-4 Spline Specifications 415
14-5 Spline Surfaces 416
14-6 Trimming Spline Surfaces 416
14-7 Cubic-Spline Interpolation Methods 417
  Natural Cubic Splines 417
  Hermite Interpolation 418
  Cardinal Splines 419
  Kochanek-Bartels Splines 423
14-8 Bézier Spline Curves 423
  Bézier Curve Equations 424
  Example Bézier Curve-Generating Program 425
17-2 Surface Lighting Effects 512
17-3 Basic Illumination Models 513
   Ambient Light 513
   Diffuse Reflection 514
   Specular Reflection and the Phong Model 516
   Combined Diffuse and Specular Reflections 519
   Diffuse and Specular Reflections from Multiple Light Sources 519
   Surface Light Emissions 519
   Basic Illumination Model with Intensity Attenuation and Spotlights 520
   RGB Color Considerations 521
   Other Color Representations 522
   Luminance 522
17-4 Transparent Surfaces 522
   Translucent Materials 523
   Light Refraction 523
   Basic Transparency Model 524
17-5 Atmospheric Effects 525
17-6 Shadows 526
17-7 Camera Parameters 526
17-8 Displaying Light Intensities 526
   Distributing System Intensity Levels 527
   Gamma Correction and Video Lookup Tables 528
   Displaying Continuous-Tone Images 529
17-9 Halftone Patterns and Dithering Techniques 529
   Halftone Approximations 531
   Dithering Techniques 533
17-10 Polygon Rendering Methods 535
    Constant-Intensity Surface Rendering 536
    Gouraud Surface Rendering 536
    Phong Surface Rendering 538
    Fast Phong Surface Rendering 539
17-11 OpenGL Illumination and Surface-Rendering Functions 540
    OpenGL Point Light-Source Function 540
    Specifying an OpenGL Light-Source Position and Type 541
    Specifying OpenGL Light-Source Colors 542
    Specifying Radial-Intensity Attenuation Coefficients for an OpenGL Light Source 543
    OpenGL Directional Light Sources (Spotlights) 543
17-12 Summary 551

18 Texturing and Surface-Detail Methods 555
18-1 Modeling Surface Detail with Polygons 556
18-2 Texture Mapping 556
   Linear Texture Patterns 557
   Surface Texture Patterns 557
   Volume Texture Patterns 560
   Texture Reduction Patterns 561
   Procedural Texturing Methods 561
18-3 Bump Mapping 561
18-4 Frame Mapping 562
18-5 OpenGL Texture Functions 563
    OpenGL Line-Texture Functions 563
    OpenGL Surface-Texture Functions 566
    OpenGL Volume-Texture Functions 568
    OpenGL Color Options for Texture Patterns 568
    OpenGL Texture-Mapping Options 569
    OpenGL Texture Wrapping 569
    Copying OpenGL Texture Patterns from the Frame Buffer 570
    OpenGL Texture-Coordinate Arrays 570
    Naming OpenGL Texture Patterns 570
    OpenGL Texture Subpatterns 572
    OpenGL Texture Reduction Patterns 572
    OpenGL Texture Borders 573
    OpenGL Proxy Textures 573
    Automatic Texturing of Quadric Surfaces 574
    Homogeneous Texture Coordinates 574
    Additional OpenGL Texture Options 575
18-6 Summary 575
## 19 Color Models and Color Applications 579

### 19-1 Properties of Light 580
- The Electromagnetic Spectrum 580
- Psychological Characteristics of Color 581

### 19-2 Color Models 582
- Primary Colors 582
- Intuitive Color Concepts 583

### 19-3 Standard Primaries and the Chromaticity Diagram 583
- The XYZ Color Model 584
- Normalized XYZ Values 584
- The CIE Chromaticity Diagram 585
- Color Gamuts 585
- Complementary Colors 585
- Dominant Wavelength 586
- Purity 586

### 19-4 The RGB Color Model 586

### 19-5 The YIQ and Related Color Models 588
- The YIQ Parameters 588
- Transformations Between RGB and YIQ Color Spaces 588
- The YUV and Y/Cb/Cr Systems 589

### 19-6 The CMY and CMYK Color Models 589
- The CMY Parameters 589
- Transformations Between CMY and RGB Color Spaces 590

### 19-7 The HSV Color Model 590
- The HSV Parameters 590
- Selecting Shades, Tints, and Tones 591
- Transformations Between HSV and RGB Color Spaces 592

### 19-8 The HLS Color Model 594

### 19-9 Color Selection and Applications 595

### 19-10 Summary 595

## 20 Interactive Input Methods and Graphical User Interfaces 599

### 20-1 Graphical Input Data 600

### 20-2 Logical Classification of Input Devices 600
- Locator Devices 600

### 20-3 Input Functions for Graphical Data 603
- Input Modes 604
- Echo Feedback 604
- Callback Functions 604

### 20-4 Interactive Picture-Construction Techniques 605
- Basic Positioning Methods 605
- Dragging 605
- Constraints 605
- Grids 606
- Rubber-Band Methods 606
- Gravity Field 607
- Interactive Painting and Drawing Methods 607

### 20-5 Virtual-Reality Environments 608

### 20-6 OpenGL Interactive Input-Device Functions 608
- GLUT Mouse Functions 609
- GLUT Keyboard Functions 613
- GLUT Tablet Functions 617
- GLUT Spaceball Functions 618
- GLUT Button-Box Function 618
- GLUT Dials Function 618
- OpenGL Picking Operations 619

### 20-7 OpenGL Menu Functions 624
- Creating a GLUT Menu 624
- Creating and Managing Multiple GLUT Menus 626
- Creating GLUT Submenus 627
- Modifying GLUT Menus 629

### 20-8 Designing a Graphical User Interface 630
- The User Dialogue 630
- Windows and Icons 630
- Accommodating Multiple Skill Levels 631
- Consistency 631
- Minimizing Memorization 631
- Backup and Error Handling 632
- Feedback 632

### 20-9 Summary 633

## Global Illumination

### 21-1 Ray-Tracing Methods 640  
**Basic Ray-Tracing Algorithm** 640  
Ray–Surface Intersection Calculations 643  
Ray–Sphere Intersections 644  
Ray–Polyhedron Intersections 645  
Reducing Object-Intersection Calculations 646  
Space-Subdivision Methods 646  
Simulating Camera Focusing Effects 648  
Antialiased Ray Tracing 650  
Distributed Ray Tracing 651

### 21-2 Radiosity Lighting Model 654  
**Radiant-Energy Terms** 654  
The Basic Radiosity Model 655  
Progressive Refinement Radiosity Method 658

### 21-3 Environment Mapping 660

### 21-4 Photon Mapping 661

### 21-5 Summary 662

## Programmable Shaders

### 22-1 A History of Shading Languages 666  
**Cook’s Shade Trees** 666  
Perlin’s Pixel Stream Editor 668  
RenderMan 668

### 22-2 The OpenGL Pipeline 670  
The Fixed-Function Pipeline 670  
Changing the Pipeline Structure 671  
Vertex Shaders 672  
Fragment Shaders 672  
Geometry Shaders 672  
Tessellation Shaders 672

### 22-3 The OpenGL Shading Language 673  
**Shader Structure** 673  
Using Shaders in OpenGL 675  
Basic Data Types 679  
Vectors 679

## Algorithmic Modeling

### 23-1 Fractal-Geometry Methods 696  
**Fractal Generation Procedures** 697  
Classification of Fractals 697  
Fractal Dimension 698  
Geometric Construction of Deterministic Self-Similar Fractals 700  
Geometric Construction of Statistically Self-Similar Fractals 703  
Affine Fractal-Construction Methods 704  
Random Midpoint-Displacement Methods 704  
Controlling Terrain Topography 706  
Self-Squaring Fractals 708  
Self-Inverse Fractals 717

### 23-2 Particle Systems 719

### 23-3 Grammar-Based Modeling Methods 720

### 23-4 Summary 722

## Visualization of Data Sets

### 24-1 Visual Representations for Scalar Fields 726

### 24-2 Visual Representations for Vector Fields 728

### 24-3 Visual Representations for Tensor Fields 728

### 24-4 Visual Representations for Multivariate Data Fields 729

### 24-5 Summary 729

Mathematics for Computer Graphics 731

A-1 Coordinate Reference Frames 731
Two-Dimensional Cartesian Screen Coordinates 731
Standard Two-Dimensional Cartesian Reference Frames 732
Polar Coordinates in the xy Plane 732
Standard Three-Dimensional Cartesian Reference Frames 733
Three-Dimensional Cartesian Screen Coordinates 733
Three-Dimensional Curvilinear-Coordinate Systems 734
Solid Angle 735

A-2 Points and Vectors 735
Point Properties 735
Vector Properties 736
Vector Addition and Scalar Multiplication 737
Scalar Product of Two Vectors 738
Vector Product of Two Vectors 738

A-3 Tensors 739

A-4 Basis Vectors and the Metric Tensor 739
Determining Basis Vectors for a Coordinate Space 740
Orthonormal Basis 740
Metric Tensor 741

A-5 Matrices 742
Scalar Multiplication and Matrix Addition 742
Matrix Multiplication 743
Matrix Transpose 744
Determinant of a Matrix 744
Matrix Inverse 744

A-6 Complex Numbers 745
Basic Complex Arithmetic 745
Imaginary Unit 746
Complex Conjugate and Modulus of a Complex Number 746
Complex Division 746
Polar-Coordinate Representation for a Complex Number 747

A-7 Quaternions 747

A-8 Nonparametric Representations 748

A-9 Parametric Representations 749
A-10 Rate-of-Change Operators 750
Gradient Operator 750
Directional Derivative 751
General Form of the Gradient Operator 751
Laplace Operator 751
Divergence Operator 752
Curl Operator 752

A-11 Rate-of-Change Integral Transformation Theorems 752
Stokes’s Theorem 753
Green’s Theorem for a Plane Surface 753
Divergence Theorem 754
Green’s Transformation Equations 755

A-12 Area and Centroid of a Polygon 755
Area of a Polygon 755
Centroid of a Polygon 756

A-13 Calculating Properties of Polyhedra 757

A-14 Numerical Methods 758
Solving Sets of Linear Equations 758
Finding Roots of Nonlinear Equations 760
Evaluating Integrals 761
Solving Ordinary Differential Equations 763
Solving Partial Differential Equations 764
Least-Squares Curve-Fitting Methods for Data Sets 765

Graphics File Formats 767

B-1 Image-File Configurations 767
B-2 Color-Reduction Methods 768
Uniform Color Reduction 768
Popularity Color Reduction 768
Median-Cut Color Reduction 769

B-3 File-Compression Techniques 769
Run-Length Encoding 770
LZW Encoding 770
Other Pattern-Recognition Compression Methods 771
Huffman Encoding 771
Arithmetic Encoding 773
Discrete Cosine Transform 774

B-4 Composition of the Major File Formats 776
JPEG: Joint Photographic Experts Group 776
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

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.


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), Jorgen Schulze (University of California, San Diego), Soon Tee Teoh (San Jose State University), Iren Valova (University of Massachusetts, 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.