

- Preface p. vii
- 1 Introduction to Computer Graphics p. 1
- 1.1 What is Computer Graphics? p. 1
- 1.2 Where Computer Generated Pictures are Used p. 3
- 1.2.1. Art, Entertainment, and Publishing p. 3
- 1.2.2. Computer Graphics and Image Processing p. 3
- 1.2.3. Monitoring a Process p. 5
- 1.2.4. Displaying Simulations p. 6
- 1.2.5. Computer-aided Design p. 6
- 1.2.6. Scientific Analysis and Visualization p. 7
- 1.3 Elements of Pictures created in Computer Graphics p. 10
- 1.3.1. Polylines p. 10
- 1.3.2. Text p. 12
- 1.3.3. Filled Regions p. 13
- 1.3.4. Raster Images p. 15
- 1.3.5. Representation of Gray Shades and Color for Raster Graphics p. 18
- 1.4 Graphics Display Devices p. 22
- 1.4.1. Line Drawing Displays p. 22
- 1.4.2. Raster Displays p. 23
- 1.4.3. Indexed Color and the LUT p. 26
- 1.4.4. Other Raster Display Devices p. 29
- 1.4.5. Hard Copy Raster Devices p. 30
- 1.5 Graphics Input Primitives and Devices p. 32
- 1.5.1. Types of Input Graphics Primitives p. 32
- 1.5.2. Types of Physical Input Devices p. 32
- 1.6. Summary p. 35
- 1.7. Further Reading p. 36
- 2 Getting Started Drawing Figures p. 37
- 2.1 Getting Started Making Pictures p. 37
- 2.1.1. Device-independent Programming, and OpenGL p. 39
- 2.1.2. Windows-based Programming p. 40
- 2.1.3. Opening a Window for Drawing p. 41
- 2.2 Drawing Basic Graphics Primitives p. 42
- 2.2.1. Examples of Drawing Dot Constellations p. 47
- 2.3 Making Line Drawings p. 51
- 2.3.1. Drawing Polylines and Polygons p. 53
- 2.3.2. Line Drawing using moveto() and lineto() p. 58
- 2.3.3. Drawing Aligned Rectangles p. 59
- 2.3.4. Aspect Ratio of an Aligned Rectangle p. 60
- 2.3.5. Filling Polygons p. 61
- 2.3.6. Other Graphics Primitives in OpenGL p. 62
- 2.4 Simple Interaction with the Mouse and Keyboard p. 63
- 2.4.1. Mouse Interaction p. 63
- 2.4.2. Keyboard Interaction p. 66

- 3 More Drawing Tools p. 81
- 3.1. Introduction
- 3.2. World Windows and Viewports p. 82
- 3.2.1. The Mapping from the Window to the Viewport p. 83
- 3.2.2. Setting the Window and Viewport Automatically p. 92
- 3.3. Clipping Lines p. 95
- 3.3.1. Clipping a Line p. 96
- 3.3.2. The Cohen-Sutherland Clipping Algorithm p. 96
- 3.4. Developing the Canvas Class p. 100
- 3.4.1. Some useful Supporting Classes p. 100
- 3.4.2. Declaration of Class Canvas p. 102
- 3.4.3. Implementation of Class Canvas p. 103
- 3.5. Relative Drawing p. 105
- 3.5.1. Developing moveRel() and lineRel() p. 105
- 3.5.2. TurtleGraphics p. 106
- 3.6. Figures Based on Regular Polygons p. 110
- 3.6.1. The Regular Polygons p. 110
- 3.6.2. Variations on n-Gons p. 112
- 3.7. Drawing Circles and Arcs p. 116
- 3.7.1. Drawing arcs p. 116
- 3.8. Using the Parametric Form of a Curve p. 119
- 3.8.1. Parametric Forms for Curves p. 120
- 3.8.2. Drawing Curves Represented Parametrically p. 123
- 3.8.3. Superellipses p. 124
- 3.8.4. Polar Coordinate Shapes p. 126
- 3.8.5. 3D Curves p. 127
- 4 Vector Tools for Graphics p. 144
- 4.1. Introduction p. 145
- 4.2. Review of Vectors p. 147
- 4.2.1 Operations with Vectors p. 148
- 4.2.2. Linear Combinations of Vectors p. 149
- 4.2.3. The Magnitude of a Vector, and Unit Vectors p. 151
- 4.3. The Dot Product p. 152
- 4.3.1. Properties of the Dot Product p. 153
- 4.3.2. The Angle Between two Vectors p. 154
- 4.3.3. The Sign of $b \cdot c$; Perpendicularity p. 154
- 4.3.4. The 2D "Perp" Vector p. 156
- 4.3.5. Orthogonal Projections, and the Distance from a Point to a Line p. 157
- 4.3.6. Applications of Projection: Reflections p. 159
- 4.4. The Cross Product of Two Vectors p. 160
- 4.4.1. Geometric Interpretation of the Cross Product p. 162
- 4.4.2. Finding the Normal to a Plane p. 163
- 4.5. Representations of Key Geometric Objects p. 164
- 4.5.1. Coordinate Systems and Coordinate Frames p. 165

- 4.5.2. Affine Combinations of Points p. 167
- 4.5.3. Linear Interpolation of Two Points p. 170
- 4.5.4. "Tweening" for Art and Animation p. 170
- 4.5.5. Preview: Quadratic and Cubic Tweening, and Bezier Curves p. 172
- 4.5.6. Representing Lines and Planes p. 173
- 4.6. Finding the Intersection of Two Line Segments p. 181
- 4.6.1. Application of Line Intersections: the Circle Through Three Points p. 184
- 4.7. Intersections of Lines with Planes, and Clipping p. 186
- 4.8. Polygon Intersection Problems p. 188
- 4.8.1. Working with Convex Polygons and Polyhedra p. 188
- 4.8.2. Ray Intersections and Clipping for Convex Polygons p. 189
- 4.8.3. The Cyrus-Beck Clipping Algorithm p. 192
- 4.8.4. Clipping against Arbitrary Polygons p. 194
- 4.8.5. More Advanced Clipping p. 196
- 5 Transformations of Objects p. 209
- 5.1 Introduction p. 210
- 5.2 Introduction to Transformations p. 211
- 5.2.1. Transforming Points and Objects p. 214
- 5.2.2. The Affine Transformations p. 216
- 5.2.3. Geometric Effects of elementary 2D Affine Transformations p. 217
- 5.2.4. The Inverse of an Affine Transformation p. 222
- 5.2.5. Composing Affine Transformations p. 223
- 5.2.6. Examples of Composing 2D Transformations p. 224
- 5.2.7. Some Useful Properties of Affine Transformations p. 228
- 5.3. 3D Affine Transformations p. 233
- 5.3.1. The Elementary 3D Transformations p. 234
- 5.3.2. Composing 3D Affine Transformations p. 238
- 5.3.3. Combining Rotations p. 238
- 5.3.4. Summary of Properties of 3D Affine Transformations p. 243
- 5.4. Changing Coordinate Systems p. 244
- 5.5. Using Affine Transformations in a Program p. 247
- 5.5.1. Saving the CT for Later Use p. 254
- 5.6. Drawing 3D Scenes with OpenGL p. 258
- 5.6.1. An Overview of the Viewing Process and the Graphics Pipeline p. 259
- 5.6.2. Some OpenGL tools for Modeling and Viewing p. 262
- 5.6.3. Drawing Elementary Shapes Provided by OpenGL p. 265
- 5.6.4. Reading a Scene Description from a File p. 273
- 6 Modeling Shapes with Polygonal Meshes p. 287
- 6.1. Introduction p. 288
- 6.2. Introduction to Solid Modeling with Polygonal Meshes p. 288
- 6.2.1. Defining a Polygonal Mesh p. 290
- 6.2.2. Finding the Normal Vectors p. 292
- 6.2.3. Properties of Meshes p. 294
- 6.2.4. Mesh Models for Nonsolid Objects p. 295

- 6.2.5. Working with Meshes in a Program p. 296
- 6.3. Polyhedra p. 299
- 6.3.1. Prisms and Antiprisms p. 300
- 6.3.2. The Platonic Solids p. 302
- 6.3.3. Other Interesting Polyhedra p. 307
- 6.4. Extruded Shapes p. 310
- 6.4.1. Creating Prisms p. 310
- 6.4.2. Arrays of Extruded Prisms: "Bricklaying" p. 311
- 6.4.3. Extrusions with a "Twist" p. 313
- 6.4.4. Building Segmented Extrusions: Tubes and Snakes p. 315
- 6.4.5. "Discretely" Swept Surfaces of Revolution p. 320
- 6.5. Mesh Approximations to Smooth Objects p. 321
- 6.5.1. Representations for Surfaces p. 322
- 6.5.2. The Normal Vector to a Surface p. 323
- 6.5.3. The Effect of an Affine Transformation p. 325
- 6.5.4. Three "Generic" Shapes: the Sphere, Cylinder, and Cone p. 326
- 6.5.5. Forming a Polygonal Mesh for a Curved Surface p. 329
- 6.5.6. Ruled Surfaces p. 331
- 6.5.7. Surfaces of Revolution p. 337
- 6.5.8. The Quadric Surfaces p. 339
- 6.5.9. The Superquadrics p. 342
- 6.5.10. Tubes Based on 3D Curves p. 344
- 6.5.11. Surfaces Based on Explicit Functions of Two Variables p. 345
- 7 Three-Dimensional Viewing p. 358
- 7.1 Introduction p. 359
- 7.2. The Camera Revisited p. 359
- 7.2.1. Setting the View Volume p. 360
- 7.2.2. Positioning and Pointing the Camera p. 361
- 7.3. Building a Camera in a Program p. 366
- 7.3.1. "Flying" the Camera p. 368
- 7.4. Perspective Projections of 3D Objects p. 371
- 7.4.1 Perspective Projection of a Point p. 372
- 7.4.2 Perspective Projection of a Line p. 375
- 7.4.3. Incorporating Perspective in the Graphics Pipeline p. 379
- 7.5. Producing Stereo Views p. 392
- 7.6. Taxonomy of Projections p. 394
- 7.6.1. One-, Two-, and Three-Point Perspective p. 394
- 7.6.2. Types of Parallel Projections p. 398
- 8 Rendering Faces for Visual Realism p. 408
- 8.1. Introduction p. 409
- 8.2. Introduction to Shading Models p. 413
- 8.2.1. Geometric Ingredients for Finding Reflected Light p. 414
- 8.2.2. Computing the Diffuse Component p. 415
- 8.2.3. Specular Reflection p. 416

- 8.2.4. The Role of Ambient Light p. 419
- 8.2.5. Combining Light Contributions p. 420
- 8.2.6. Adding Color p. 421
- 8.2.7. Shading and the Graphics Pipeline p. 422
- 8.2.8. Using Light Sources in OpenGL p. 425
- 8.2.9. Working with Material Properties in OpenGL p. 429
- 8.2.10. Shading of Scenes Specified by SDL p. 430
- 8.3. Flat Shading and Smooth Shading p. 430
 - 8.3.1. Flat Shading p. 432
 - 8.3.2. Smooth Shading p. 433
- 8.4. Removing Hidden Surfaces p. 436
 - 8.4.1. The Depth Buffer Approach p. 437
- 8.5. Adding Texture to Faces p. 439
 - 8.5.1. Pasting the Texture onto a Flat Surface p. 442
 - 8.5.2. Rendering the Texture p. 444
 - 8.5.3. What Does a Texture Modulate? p. 451
 - 8.5.4. A Texture Example Using OpenGL p. 453
 - 8.5.5. Wrapping Texture on Curved Surfaces p. 457
 - 8.5.6. Reflection Mapping p. 461
- 8.6. Adding Shadows of Objects p. 465
 - 8.6.1. Shadows as Texture p. 465
 - 8.6.2. Creating Shadows with the Use of a Shadow Buffer p. 467
- 9 Approaches to Infinity p. 472
 - 9.1. Introduction p. 473
 - 9.2 Fractals and Self-Similarity p. 473
 - 9.2.1. Successive Refinement of Curves p. 474
 - 9.2.2. Drawing Koch Curves and Snowflakes p. 475
 - 9.2.3. Fractional Dimension p. 477
 - 9.3. String Production and Peano Curves p. 478
 - 9.3.2. Producing Recursively and Drawing in a Program p. 480
 - 9.3.3. Allowing Branching p. 483
 - 9.4. Tiling the Plane p. 485
 - 9.4.1. Monohedral Tilings p. 486
 - 9.4.2. Dihedral Tilings p. 488
 - 9.4.3. Drawing Tilings p. 491
 - 9.4.4. Reptiles p. 491
 - 9.5. Creating an Image by Means of Iterated Functions Systems p. 494
 - 9.5.1. An Experimental Copier p. 494
 - 9.5.2. Some Underlying theory of the Copying Process p. 496
 - 9.5.3. Drawing the k-th Iterate p. 497
 - 9.5.4. The Chaos Game p. 498
 - 9.5.5. Finding the IFS, Fractal Image Compression p. 501
 - 9.6. The Mandelbrot Set p. 504
 - 9.6.1. Mandelbrot Sets and Iterated Functions Systems p. 505

- 9.6.2. Defining the Mandelbrot Set p. 508
- 9.6.3. Computing whether the point c is in the Mandelbrot Set p. 510
- 9.6.4. Drawing the Mandelbrot Set p. 511
- 9.6.5. Some Notes on the Mandelbrot Set p. 512
- 9.7. Julia Sets p. 513
- 9.7.1. The Filled-in Julia Set K_c p. 513
- 9.7.2. Drawing Filled-in Julia Sets p. 513
- 9.7.3. Some Notes on Fixed points and Basins of Attraction p. 514
- 9.7.4. The Julia Set J_c p. 514
- 9.8. Random Fractals p. 519
- 9.8.1. Fractalizing a Segment p. 519
- 9.8.2. Controlling the Spectral Density of the Fractal Curve p. 521
- 10 Tools for Raster Displays p. 532
- 10.1. Introduction p. 533
- 10.2. Manipulating Pixmaps p. 534
- 10.2.1. Operations of Interest for Pixmaps p. 534
- 10.2.2. Useful Data Types for Pixmaps p. 535
- 10.2.3. Scaling and Rotating Images p. 540
- 10.3. Combining Pixmaps p. 543
- 10.3.1. The Read-Modify-Write Cycle p. 544
- 10.3.2. The Alpha Channel and Image Blending p. 545
- 10.3.3. Logical Combinations of Pixmaps p. 549
- 10.3.4. The BitBlt operation p. 553
- 10.4. Do-It-Yourself Line Drawing: Bresenham's Algorithm p. 554
- 10.4.1. Bresenham's Line-Drawing Algorithm p. 555
- 10.5 Defining and Filling Regions of Pixels p. 561
- 10.5.1. Defining Regions p. 561
- 10.5.2. Pixel-Defined Regions p. 561
- 10.5.3. A Recursive Flood-Fill Algorithm p. 562
- 10.5.4. Filling Regions with Patterns p. 563
- 10.5.5. Using Coherence: Region Filling Based on Runs of Pixels p. 565
- 10.6. Manipulating Symbolically Defined Regions p. 566
- 10.6.1. Rectangle-defined Regions p. 567
- 10.6.2. Path-defined Regions p. 568
- 10.7. Filling Polygon-Defined Regions p. 570
- 10.7.1. Which Pixels on an Edge Belong to a Polygon? p. 571
- 10.7.2 Improving the Algorithm's Performance p. 573
- 10.8. Aliasing; Antialiasing Techniques p. 577
- 10.8.1. Antialiasing Techniques p. 578
- 10.8.2. Antialiasing of Texture p. 583
- 10.8.3. Antialiasing Using OpenGL p. 585
- 10.9. Creating More Shades and Colors p. 587
- 10.9.1. Ordered Dither p. 587
- 10.9.2. Error Diffusion p. 592

- 11 Curve and Surface Design p. 597
- 11.1. Introduction p. 598
- 11.1.1 Parametric Curves as Trajectories p. 598
- 11.1.2. Smoothness of Motion p. 598
- 11.2. Describing Curves by Means of Polynomials p. 603
- 11.3. On Interactive Curve Design p. 607
- 11.4. Bezier Curves for Curve Design p. 609
- 11.4.1. The de Casteljau Algorithm p. 610
- 11.5. Properties of Bezier Curves p. 614
- 11.6. Finding Better Blending Functions p. 618
- 11.6.1. The Problem of Local Control p. 619
- 11.6.2. Wish List for a Set of Blending Functions p. 620
- 11.6.3. Piecewise Polynomial Curves and Splines p. 621
- 11.6.4. Building a set of Blending Functions Out of $g(t)$ p. 623
- 11.6.5. Spline Curves and Basis Functions p. 624
- 11.7. The B-Spline Basis Functions p. 627
- 11.7.1. Definition of B-Spline Functions p. 628
- 11.7.2. Using Multiple Knots in the Knot Vector p. 632
- 11.7.3. Open B-Spline Curves: Standard Knot Vector p. 633
- 11.8. Useful Properties of B-Spline Curves for Design p. 637
- 11.8.1. Using Multiple Control Points p. 638
- 11.9. Rational Splines and NURBS Curves p. 639
- 11.10. A Glimpse at Interpolation p. 643
- 11.10.1. Interpolation Using Piecewise Cubic Polynomials p. 643
- 11.10.2. Hermite Interpolation p. 644
- 11.10.3. The Natural Cubic Spline p. 648
- 11.10.4. Computing the Slopes in Cubic Interpolation p. 649
- 11.10.5. Specifying the Tangent Vectors Interactively p. 653
- 11.11. Modeling Curved Surfaces p. 654
- 11.11.1. Ruled Surfaces Based on B-Splines p. 654
- 11.11.2. Surfaces of Revolution Based on B-Splines p. 655
- 11.11.3. Bezier Surface Patches p. 656
- 11.11.4. Patching Together Bezier Patches p. 658
- 11.11.5. B-Spline Patches p. 659
- 11.11.6. NURBS Surfaces p. 660
- 12 Color Theory p. 672
- 12.1. Introduction p. 672
- 12.2. Describing Colors p. 673
- 12.2.1. Dominant Wavelength p. 675
- 12.2.2. Color Matching p. 676
- 12.3. The International Commission on Illumination Standard p. 678
- 12.3.1. Constructing the CIE Chart p. 680
- 12.3.2. Using the CIE Chromaticity Diagram p. 681
- 12.3.3. Color Gamuts p. 682

- 12.4. Color Spaces p. 683
- 12.4.1. The RGB and CMY Color Spaces p. 683
- 12.4.2. Additive and Subtractive Color Systems p. 685
- 12.4.3. The HLS Color Model p. 686
- 12.5. Color Quantization p. 688
- 12.5.1. Uniform Quantization p. 690
- 12.5.2. The Popularity Algorithm p. 692
- 12.5.3. The Median-cut Algorithm p. 692
- 12.5.4. Octree Quantization p. 693
- 13 Hidden Surface Removal p. 699
- 13.1 Introduction p. 700
- 13.1.2. Object Precision versus Image Precision Approaches p. 701
- 13.1.3. Description of the Polygon Mesh Data p. 702
- 13.2. The Depth Buffer Algorithm Revisited p. 704
- 13.3. List Priority HSR Methods p. 705
- 13.3.1. The Heedless Painter's Algorithm p. 706
- 13.3.2. HSR Using Binary Space Partition Trees p. 707
- 13.3.3. The Depth Sort Algorithm p. 708
- 13.4. A Scan-Line HSR Method p. 713
- 13.5. Area Subdivision Approaches p. 716
- 13.5.1. Quadrant Subdivision p. 716
- 13.5.2. Other Definitions of a Simple Region p. 719
- 13.6. On Hidden Line Removal Methods p. 721
- 13.6.1. The Geometric Testing in edgeTest() p. 724
- 13.7. HSR Methods for Curved Surfaces p. 725
- 14 Ray Tracing p. 732
- 14.1. Introduction p. 733
- 14.2. Setting Up the Geometry of Ray Tracing p. 734
- 14.3. Overview of the Ray-Tracing Process p. 736
- 14.4. Intersection of a Ray with an Object p. 737
- 14.4.1. Intersection of a Ray with a Generic Plane p. 738
- 14.4.2. Intersection with a Generic Sphere p. 739
- 14.4.3. Intersection of the Ray with Transformed Objects p. 740
- 14.5. Organizing a Ray Tracer Application p. 741
- 14.5.1. A Routine to Compute Ray-Sphere Intersections p. 747
- 14.5.2. A Complete Ray Tracer for Emissive-Sphere Scenes p. 749
- 14.6. Intersecting Rays with Other Primitives p. 750
- 14.6.1. Intersecting with a Square p. 750
- 14.6.2. Intersecting with a Tapered Cylinder p. 750
- 14.6.3. Intersecting with a Cube (or any Convex Polyhedron) p. 754
- 14.6.4. Adding More Primitives p. 759
- 14.7. Drawing Shaded Pictures of Scenes p. 760
- 14.7.1. Finding the Normal at the Hit Spot p. 761
- 14.7.2. Coloring Objects According to their Surface Materials p. 762

- 14.7.3. Physically-based Shading Models: Cook-Torrance Shading p. 764
- 14.8. Adding Surface Texture p. 770
 - 14.8.1. Solid Texture p. 770
 - 14.8.2. Pasting Images onto Surfaces p. 778
 - 14.8.3. Antialiasing Ray Tracings p. 780
- 14.9. Using Extents p. 781
 - 14.9.1. Box and Sphere Extents p. 783
 - 14.9.2. Using Projection Extents p. 788
- 14.10. Adding Shadows for Greater Realism p. 790
- 14.11. Reflections and Transparency p. 793
 - 14.11.1. The Refraction of Light p. 795
 - 14.11.2. Dealing with Refraction in shade() p. 799
- 14.12. Compound Objects: Boolean Operations on Objects p. 802
 - 14.12.1. Ray Tracing CSG Objects p. 803
 - 14.12.2. Data Structure for Boolean Objects p. 805
 - 14.12.3. Intersecting Rays with Boolean Objects p. 806
 - 14.12.4. Building and Using Extents for CSG Objects p. 811
- Appendixes
 - 1 Graphics Tools - Obtaining OpenGL p. 818
 - A1.1 Obtaining and Installing OpenGL p. 818
 - 2 Some Mathematics for Computer Graphics p. 820
 - A2.1 Some Key Definitions for Matrices and their Operations p. 820
 - A2.2 Some Properties of Vectors and their Operations p. 826
 - A2.3. The Arithmetic of Complex Numbers p. 828
 - A2.4. Spherical Coordinates and Direction Cosines p. 830
 - 3 Some Useful Classes and Utility Routines p. 833
 - Classes for 2D Graphics p. 833
 - RGBPixmap Class p. 837
 - The Scene and Supporting Classes p. 840
 - Noise Class p. 862
 - Some Classes that are Useful for Ray Tracing p. 863
 - 4 An Introduction to PostScript p. 866
 - A4.1. About the PostScript Language p. 867
 - A4.2. Graphics Operators in PostScript p. 872
 - A4.3. Drawing Text in PostScript p. 880
 - A4.4. Defining New Variables and Procedures p. 881
 - A4.5. Decisions and Iterations p. 886
 - A4.6. Printing Values p. 893
 - A4.7. Drawing Gray-scale Image p. 894
 - 5 An Introduction to SDL p. 897
 - A5.1 Syntax of SDL p. 898
 - A5.2. Macros in SDL p. 901
 - A5.3. Extending SDL p. 902
- References p. 904

- Index p. 909