

<<交互式计算机图形学>>

图书基本信息

书名：<<交互式计算机图形学>>

13位ISBN编号：9787040137248

10位ISBN编号：7040137240

出版时间：2003-10

出版时间：北京蓝色畅想图书发行有限公司（原高等教育出版社）

作者：安杰尔

页数：719

字数：900000

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<交互式计算机图形学>>

### 内容概要

本书向读者讲授如何用OpenGL从初级阶段快速制作优质的交互式计算机图形实例，OpenGL是业界广泛采用的三维图形API，本书藉此向读者提供了在计算机图形学领域更广阔空间内发展的坚实基础。书中内容覆盖了计算机图形学基础课程所需的所有论题，如光景交互作用、图形明暗处理、图形建模、曲线与曲面、偏差预防、纹理映射和图形组合，还探讨了计算机硬件方面的问题。

本版为配合计算机图形学领域最新进展，增加了更具广度和深度的内容，包括图像处理、多重透视、并行透视等，并运用选择模式、仿真模拟、面向对象图形学与实景图像、射线跟踪等方法讲解了图形交互的内容。

随书所附光盘包括OpenGL指令集参考、OpenGL手册、书中源码代码和附加的实用案例。

本书适用于高等院校本科高年级和研究生的计算机图形学、交互式计算机系统等课程。

<<交互式计算机图形学>>

作者简介

Edward Angel is currently a professor of Computer Science, Electrical and Computer Engineering and Media Arts at the university of new mexico. He was recently named the Director of the Art tech Center at UNM in the College of Fine Arts. Professor Angel

## 书籍目录

Preface Chapter 1 Graphics Systems and Models 1.1 Applications of Computer Graphics 1.1.1 Display of Information 1.1.2 Design 1.1.3 Simulation and Animation 1.1.4 User Interfaces 1.2 A Graphics System 1.2.1 Pixels and the Frame Buffer 1.2.2 Output Devices 1.2.3 Input Devices 1.3 Images: Physical and Synthetic 1.3.1 Objects and Viewers 1.3.2 Light and Images 1.3.3 Ray Tracing 1.4 The Human Visual System 1.5 The Pinhole Camera 1.6 The Synthetic-Camera Model 1.7 The Programmer's Interface 1.7.1 Application Programmer's Interfaces 1.7.2 The Pen-Plotter Model 1.7.3 Three-Dimensional APIs 1.7.4 A Sequence of Images 1.7.5 The Modeling-Rendering Paradigm 1.8 Graphics Architectures 1.8.1 Display Processors 1.8.2 Pipeline Architectures 1.8.3 Transformations 1.8.4 Clipping 1.8.5 Projection 1.8.6 Rasterization 1.8.7 Performance Characteristics 1.9 Summary and Notes 1.10 Suggested Readings Exercises Chapter 2 Graphics Programming 2.1 The Sierpinski Gasket 2.2 Programming Two-Dimensional Applications 2.2.1 Coordinate Systems 2.3 The OpenGL API 2.3.1 Graphics Functions 2.3.2 The Graphics Pipeline and State Machines 2.3.3 The OpenGL Interface 2.4 Primitives and Attributes 2.4.1 Polygon Basics 2.4.2 Polygon Types in OpenGL 2.4.3 Drawing a Sphere 2.4.4 Text 2.4.5 Curved Objects 2.4.6 Attributes 2.5 Color 2.5.1 RGB Color 2.5.2 Indexed Color 2.5.3 Setting of Color Attributes 2.6 Viewing 2.6.1 Two-Dimensional Viewing 2.6.2 The Orthographic View 2.6.3 Matrix Modes 2.7 Control Functions 2.7.1 Interaction with the Window System 2.7.2 Aspect Ratio and Viewports 2.7.3 The main, display, and myinit Functions 2.7.4 Program Structure 2.8 The Gasket Program 2.9 Polygons and Recursion 2.10 The Three-Dimensional Gasket 2.10.1 Use of Three-Dimensional Points 2.10.2 Use of Polygons in Three Dimensions 2.10.3 Hidden-Surface Removal 2.11 Summary and Notes 2.12 Suggested Readings Exercises Chapter 3 Input and Interaction 3.1 Interaction 3.2 Input Devices 3.2.1 Physical Input Devices 3.2.2 Logical Devices 3.2.3 Measure and Trigger 3.2.4 Input Modes 3.3 Clients and Servers 3.4 Display Lists 3.4.1 Definition and Execution of Display Lists 3.4.2 Text and Display Lists 3.4.3 Fonts in GLUT 3.5 Programming Event-Driven Input 3.5.1 Using the Pointing Device 3.5.2 Window Events 3.5.3 Keyboard Events 3.5.4 The Display and Idle Callbacks 3.5.5 Window Management 3.6 Menus 3.7 Picking 3.7.1 Picking and Selection Mode 3.8 A Simple Paint Program 3.9 Animating Interactive Programs 3.9.1 The Rotating Square 3.9.2 Double Buffering 3.9.3 Other Buffering Problems 3.10 Design of Interactive Programs 3.10.1 Toolkits, Widgets, and the Frame Buffer 3.11 Logic Operations 3.11.1 Drawing Erasable Lines 3.11.2 XOR and Color 3.11.3 Cursors and Overlay Planes 3.12 Summary and Notes 3.13 Suggested Readings Exercises Chapter 4 Geometric Objects and Transformations 4.1 Scalars, Points, and Vectors 4.1.1 The Geometric View 4.1.2 Coordinate-Free Geometry 4.1.3 The Mathematical View: Vector and Affine Spaces 4.1.4 The Computer-Science View 4.1.5 Geometric ADTs 4.1.6 Lines 4.1.7 Affine Sums 4.1.8 Convexity 4.1.9 Dot and Cross Products 4.1.10 Planes 4.2 Three-Dimensional Primitives 4.3 Coordinate Systems and Frames 4.3.1 Representations and N-tuples 4.3.2 Changes of Coordinate Systems 4.3.3 Example of Change of Representation 4.3.4 Homogeneous Coordinates 4.3.5 Example of Change in Frames 4.3.6 Working with Representations 4.3.7 Frames and ADTs 4.3.8 Frames in OpenGL 4.4 Modeling a Colored Cube 4.4.1 Modeling of a Cube 4.4.2 Inward- and Outward-Pointing Faces 4.4.3 Data Structures for Object Representation 4.4.4 The Color Cube 4.4.5 Bilinear Interpolation 4.4.6 Vertex Arrays 4.5 Affine Transformations 4.6 Translation, Rotation, and Scaling 4.6.1 Translation 4.6.2 Rotation 4.6.3 Scaling 4.7 Transformations in Homogeneous Coordinates 4.7.1 Translation 4.7.2 Scaling 4.7.3 Rotation 4.7.4 Shear 4.8 Concatenation of Transformations 4.8.1 Rotation About a Fixed Point 4.8.2 General Rotation 4.8.3 The Instance Transformation 4.8.4 Rotation About an Arbitrary Axis 4.9 OpenGL Transformation Matrices 4.9.1 The Current Transformation Matrix 4.9.2 Rotation, Translation, and Scaling 4.9.3 Rotation About a Fixed Point in OpenGL 4.9.4 Order of Transformations 4.9.5 Spinning of the Cube 4.9.6 Loading, Pushing, and Popping Matrices 4.10 Interfaces to Three-Dimensional Applications 4.10.1 Using Areas of the Screen 4.10.2 A Virtual Trackball 4.10.3 Smooth Rotations 4.10.4 Incremental Rotation 4.11 Quaternions 4.11.1 Complex Numbers and Quaternions 4.11.2

Quaternions and Rotation 4.12 Summary and Notes 4.13 Suggested Readings ExercisesChapter 5 Viewing 5.1 Classical and Computer Viewing 5.1.1 Classical Viewing 5.1.2 Orthographic Projections 5.1.3 Axonometric Projections 5.1.4 Oblique Projections 5.1.5 Perspective Viewing 5.2 Viewing with a Computer 5.3 Positioning of the Camera 5.3.1 Positioning of the Camera Frame 5.3.2 Two Viewing APIs 5.3.3 The Look-At Function 5.3.4 Other Viewing APIs 5.4 Simple Projections 5.4.1 Perspective Projections 5.4.2 Orthogonal Projections 5.5 Projections in OpenGL 5.5.1 Perspective in OpenGL 5.5.2 Parallel Viewing in OpenGL 5.6 Hidden-Surface Removal 5.6.1 Culling 5.7 Walking Through a Scene 5.8 Parallel-Projection Matrices 5.8.1 Projection Normalization 5.8.2 Orthogonal-Projection Matrices 5.8.3 Oblique Projections 5.9 Perspective-Projection Matrices 5.9.1 Perspective Normalization 5.9.2 OpenGL Perspective Transformations 5.10 Projections and Shadows 5.11 Summary and Notes 5.12 Suggested Readings ExercisesChapter 6 Shading 6.1 Light and Matter 6.2 Light Sources 6.2.1 Color Sources 6.2.2 Ambient Light 6.2.3 Point Sources 6.2.4 Spotlights 6.2.5 Distant Light Sources 6.3 The Phong Reflection Model 6.3.1 Ambient Reflection 6.3.2 Diffuse Reflection 6.3.3 Specular Reflection 6.4 Computation of Vectors 6.4.1 Normal Vectors 6.4.2 Angle of Reflection 6.4.3 Use of the Halfway Vector 6.4.4 Transmitted Light 6.5 Polygonal Shading 6.5.1 Flat Shading 6.5.2 Interpolative and Gouraud Shading 6.5.3 Phong Shading 6.6 Approximation of a Sphere by Recursive Subdivision 6.7 Light Sources in OpenGL 6.8 Specification of Materials in OpenGL 6.9 Shading of the Sphere Model 6.10 Global Rendering 6.11 Summary and Notes 6.12 Suggested Readings ExercisesChapter 7 Discrete Techniques 7.1 Buffers 7.2 Digital Images 7.3 Writes into Buffers 7.3.1 Writing Modes 7.3.2 Writes with XOR 7.4 Bit and Pixel Operations in OpenGL 7.4.1 OpenGL Buffers and the Pixel Pipeline 7.4.2 Bitmaps 7.4.3 Raster Fonts 7.4.4 Pixels and Images 7.4.5 Lookup Tables 7.4.6 Buffers for Picking 7.5 Mapping Methods 7.6 Texture Mapping 7.6.1 Two-Dimensional Texture Mapping 7.6.2 Texture Mapping in OpenGL 7.6.3 Texture Objects 7.6.4 Multitexturing 7.6.5 Texture Generation 7.7 Environmental Maps 7.8 Bump Maps 7.9 Compositing Techniques 7.9.1 Opacity and Blending 7.9.2 Image Compositing 7.9.3 Blending and Compositing in OpenGL 7.9.4 Antialiasing 7.9.5 Back-to-Front and Front-to-Back Rendering 7.9.6 Depth Cueing and Fog 7.10 Multirendering and the Accumulation Buffer 7.10.1 Scene Antialiasing 7.10.2 Bump Mapping and Embossing 7.10.3 Image Processing 7.10.4 Imaging Extensions 7.10.5 Other Multipass Methods 7.11 Sampling and Aliasing 7.11.1 Sampling Theory 7.11.2 Reconstruction 7.11.3 Quantization 7.12 Summary and Notes 7.13 Suggested Readings ExercisesChapter 8 Implementation of a Renderer 8.1 Basic Implementation Strategies 8.2 Four Major Tasks 8.2.1 Modeling 8.2.2 Geometric Processing 8.2.3 Rasterization 8.2.4 Display 8.3 Implementation of Transformations 8.4 Line-Segment Clipping 8.4.1 Cohen-Sutherland Clipping 8.4.2 Liang-Barsky Clipping 8.5 Polygon Clipping 8.6 Clipping of Other Primitives 8.6.1 Bounding Boxes and Volumes 8.6.2 Curves, Surfaces, and Text 8.6.3 Clipping in the Frame Buffer 8.7 Clipping in Three Dimensions 8.8 Hidden-Surface Removal 8.8.1 Object-Space and Image-Space Approaches 8.8.2 Sorting and Hidden-Surface Removal 8.8.3 Back-Face Removal 8.8.4 The z-Buffer Algorithm 8.8.5 Depth Sort and the Painter's Algorithm 8.8.6 The Scan-Line Algorithm 8.9 Scan Conversion 8.10 Bresenham's Algorithm 8.11 Scan Conversion of Polygons 8.11.1 Inside-Outside Testing 8.11.2 OpenGL and Concave Polygons 8.11.3 Scan Conversion with the z-Buffer 8.11.4 Fill and Sort 8.11.5 Flood Fill 8.11.6 Scan-Line Algorithms 8.11.7 Singularities 8.12 Antialiasing 8.13 Display Considerations 8.13.1 Color Systems 8.13.2 The Color Matrix 8.13.3 Gamma Correction 8.13.4 Dithering and Halftoning 8.14 Summary and Notes 8.15 References ExercisesChapter 9 Hierarchical and Object-Oriented Modeling 9.1 Symbols and Instances 9.2 Hierarchical Models 9.3 A Robot Arm 9.4 Trees and Traversal 9.4.1 A Stack-Based Traversal 9.5 Use of Tree Data Structures 9.6 Animation 9.7 Graphical Objects 9.7.1 Methods, Attributes, and Messages 9.7.2 A Cube Object 9.7.3 Implementing the Cube Object 9.7.4 Objects and Hierarchy 9.7.5 Geometric Objects 9.8 Scene Graphs 9.9 A Simple Scene Graph API 9.9.1 The Node Class 9.9.2 Geometry Nodes 9.9.3 Camera Class 9.9.4 Lights and Materials 9.9.5 Transformations 9.9.6 The Robot Figure 9.9.7 Implementing the Viewer 9.9.8 Implementing a Node 9.10 Other Tree Structures 9.10.1 CSG Trees 9.10.2 Shade Trees 9.10.3 BSP Trees 9.10.4 Quadtrees and Octrees

9.11 Graphics and the Web 9.11.1 Networks and Protocols 9.11.2 Hypermedia and HTML 9.11.3 Databases and VRML 9.11.4 JAVA and Applets 9.12 Summary and Notes 9.13 Suggested Readings Exercises

Chapter 10 Curves and Surfaces 10.1 Representation of Curves and Surfaces 10.1.1 Explicit Representation 10.1.2 Implicit Representations 10.1.3 Parametric Form 10.1.4 Parametric Polynomial Curves 10.1.5 Parametric Polynomial Surfaces 10.2 Design Criteria 10.3 Parametric Cubic Polynomial Curves 10.4 Interpolation 10.4.1 Blending Functions 10.4.2 The Cubic Interpolating Patch 10.5 Hermite Curves and Surfaces 10.5.1 The Hermite Form 10.5.2 Geometric and Parametric Continuity 10.6 Bezier Curves and Surfaces 10.6.1 Bezier Curves 10.6.2 Bezier Surface Patches 10.7 Cubic B-Splines 10.7.1 The Cubic B-Spline Curve 10.7.2 B-Splines and Basis 10.7.3 Spline Surfaces 10.8 General B-Splines 10.8.1 Recursively Defined B-Splines 10.8.2 Uniform Splines 10.8.3 Nonuniform B-Splines 10.8.4 NURBS 10.9 Rendering of Curves and Surfaces 10.9.1 Polynomial Evaluation Methods 10.9.2 Recursive Subdivision of Bezier Polynomials 10.9.3 Rendering of Other Polynomial Curves by Subdivision 10.9.4 Subdivision of Bezier Surfaces 10.10 The Utah Teapot 10.11 Algebraic Surfaces 10.11.1 Quadrics 10.11.2 Rendering of Surfaces by Ray Casting 10.11.3 Subdivision Curves and Surfaces 10.12 Curves and Surfaces in OpenGL 10.12.1 Bezier Curves 10.12.2 Bezier Surfaces 10.12.3 Displaying the Teapot 10.12.4 NURBS Functions 10.12.5 Quadrics 10.13 Summary and Notes 10.14 References and Notes Exercises

Chapter 11 Procedural Methods 11.1 Reasons for Using Procedural Models 11.2 Physically Based Models and Particle Systems 11.3 Newtonian Particles 11.3.1 Independent Particles 11.3.2 Spring Forces 11.3.3 Attractive and Repulsive Forces 11.4 Solving Particle Systems 11.5 Constraints 11.5.1 Collisions 11.5.2 Particles Inside a Sphere 11.5.3 Soft Constraints 11.6 Language-Based Models 11.7 Recursive Methods and Fractals 11.7.1 Rulers and Length 11.7.2 Fractal Dimension 11.7.3 Midpoint Division and Brownian Motion 11.7.4 Fractal Mountains 11.8 The Mandelbrot Set 11.9 Summary and Notes 11.10 Suggested Readings Exercises

Chapter 12 Visualization 12.1 Data + Geometry 12.2 Height Fields and Contours 12.2.1 Meshes 12.2.2 Contour Plots 12.2.3 Marching Squares 12.3 Visualizing Surfaces and Scalar Fields 12.3.1 Volumetric Data Sets 12.3.2 Visualization of Implicit Functions 12.4 Isosurfaces and Marching Cubes 12.5 Mesh Simplification 12.6 Direct Volume Rendering 12.6.1 Assignment of Color and Opacity 12.6.2 Splatting 12.6.3 Volume Ray Tracing 12.6.4 Texture Mapping of Volumes 12.7 Vector-Field Visualization 12.7.1 Hedgehogs 12.7.2 Glyphs 12.7.3 Color 12.7.4 Particle Traces and Streamlines 12.8 Tensor Visualization 12.9 Summary and Notes 12.10 Suggested Readings Exercises

Chapter 13 Advanced Rendering 13.1 Going Beyond Pipeline Rendering 13.2 Ray Tracing 13.3 Building a Simple Ray Tracer 13.3.1 Recursive Ray Tracing 13.3.2 Calculating Intersections 13.3.3 Ray-Tracing Variations 13.4 The Rendering Equation 13.5 Radiosity 13.5.1 The Radiosity Equation 13.5.2 Solving the Radiosity Equation 13.5.3 Computing Form Factors 13.5.4 Carrying Out Radiosity 13.6 Renderman 13.7 Large-Scale Rendering 13.7.1 Sort-Middle Rendering 13.7.2 Sort-Last Rendering 13.7.3 Sort-First Rendering 13.8 Image-Based Rendering 13.8.1 A Simple Example 13.9 Summary and Notes 13.10 Suggested Readings Exercises

Appendix A Sample Programs Appendix B Spaces Appendix C Matrices References Function Index Subject Index

## 章节摘录

插图：Classical graphics techniques arose as a medium to convey information among people. Although spoken and written languages serve a similar purpose, the human visual system is unrivaled both as a processor of data and as a pattern recognizer. More than 4000 years ago, the Babylonians displayed floor plans of buildings on stones. More than 2000 years ago, the Greeks were able to convey their architectural ideas graphically, even though the related mathematics was not developed until the Renaissance. Today, the same type of information is generated by architects, mechanical designers, and draftspeople using computer-based drafting systems. For centuries, cartographers have developed maps to display celestial and geographical information. Such maps were crucial to navigators as these people explored the ends of the earth: maps are no less important today in fields such as geographic information systems. Now, maps can be developed and manipulated in real time over the Internet. Over the past 100 years, workers in the field of statistics have explored techniques for generating plots that aid the viewer in determining the information in a set of data. Now, we have computer plotting packages that provide a variety of plotting techniques and color tools, and that can handle multiple large data sets. Nevertheless, it is still the human's ability to recognize visual patterns that ultimately allows us to interpret the information contained in the data. Medicine poses interesting and important data analysis problems. New imaging technologies—such as computed tomography (CT), magnetic resonance imaging (MRI), ultrasound, and positron emission tomography (PET)—generate three-dimensional data that must be subjected to algorithmic manipulation to provide useful information. Color Plate 20 shows an image of a person's head in which the skin is displayed as transparent and the muscles are displayed as opaque. Although the data were collected by a medical imaging system, computer graphics produced the image that shows the structures.

## <<交互式计算机图形学>>

### 媒体关注与评论

书评本套教学用书的特点： 权威性——教育部高等教育司推荐、教育部高等学校信息科学与技术引进教材专家组遴选。

系统性——覆盖计算机专业主干课程和非计算机专业计算机基础课程； 先进性——著名计算机专家近两年的最新著作，内容体系先进； 经济性——价格与国内自编教材相当，是国内引进教材价格最低的。

This is third edition of this widely adopted textbook. It gets beginners creating exciting interactive graphics applications quickly using OpenGL, the most widely used 3D graphics API in the industry, and provides a solid background for future work in computer graphics. The book covers all topics required for a fundamental course in computer graphics, such as light-material interactions, shading, modeling, curves and surfaces, antialiasing, texture mapping, and compositing, as well as hardware issues. This edition includes an expanded breadth and depth of material to account for new developments in the field, including new material on image processing, texture mapping, multirendering, image-based rendering, and parallel rendering, as well as more material on interaction using selection mode, smooth animations, object-oriented graphics and scene graphics, and ray tracing and radiosity. Included is a CD-ROM with OpenGL resources and book supplements.



编辑推荐

《交互式计算机图形学:自顶向下方法与OpenGL应用(第3版 影印版)》由高等教育出版社出版。本套教学用书的特点:权威性——教育部高等教育司推荐、教育部高等学校信息科学与技术引进教材专家组遴选。

系统性——覆盖计算机专业主干课程和非计算机专业计算机基础课程;先进性——著名计算机专家近两年的最新著作,内容体系先进;经济性——价格与国内自编教材相当,是国内引进教材价格最低的。

This is third edition of this widely adopted textbook. It gets beginners creating exciting interactive graphics applications quickly using OpenGL, the most widely used 3D graphics API in the industry, and provides a solid background for future work in computer graphics. The book covers all topics required for a fundamental course in computer graphics, such as light-material interactions, shading, modeling, curves and surfaces, antialiasing, texture mapping, and compositing, as well as hardware issues. This edition includes an expanded breadth and depth of material to account for new developments in the field, including new material on image processing, texture mapping, multirendering, image-based rendering, and parallel rendering, as well as more material on interaction using selection mode, smooth animations, object-oriented graphics and scene graphics, and ray tracing and radiosity. Included is a CD-ROM with OpenGL resources and book supplements.

<<交互式计算机图形学>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>