

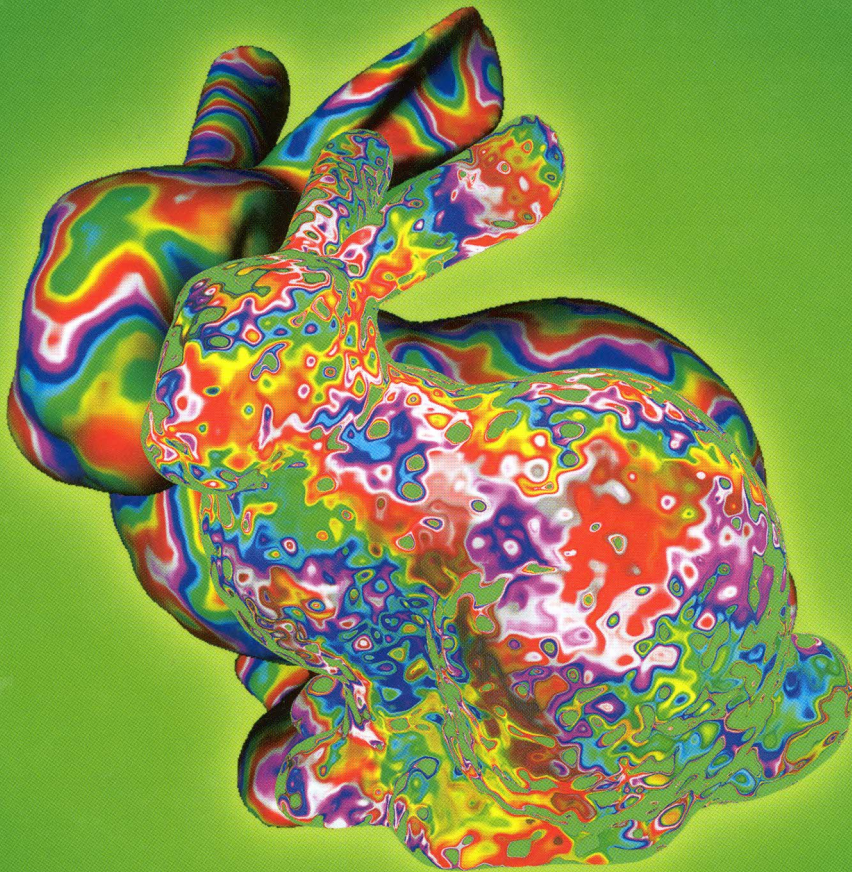
INCLUDES
OPENGL 4.X

 CRC Press
Taylor & Francis Group
AN A K PETERS BOOK

Graphics Shaders

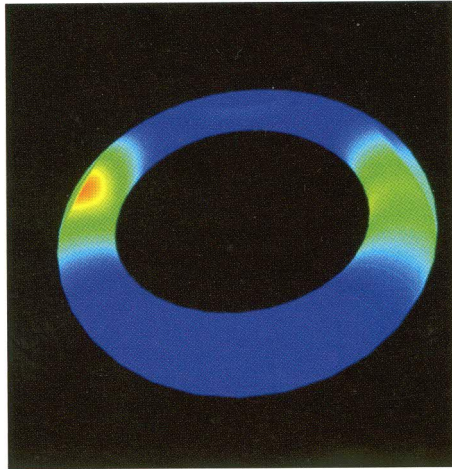
THEORY AND PRACTICE

SECOND EDITION



MIKE BAILEY • STEVE CUNNINGHAM

Contents



Foreword	xix
-----------------	------------

Preface	xxiii
----------------	--------------

1. The Fixed-Function Graphics Pipeline	1
The Traditional View	2
The Vertex Operation	2
The Fragment Processing Part of the Pipeline	6
State in the Graphics Pipeline	7

How the Traditional View Is Implemented	8
Vertex Processing	9
Rendering Processing	10
Homogeneous Coordinates in the Fixed-Function Pipeline	14
Vertex Arrays	17
Conclusions	20
Exercises	21
2. OpenGL Shader Evolution	25
History of Shaders	27
OpenGL Shader History	30
OpenGL 2.0/GLSL 1.10	30
OpenGL 3.x/GLSL 3.30	31
OpenGL 4.0/GLSL 4.00	32
OpenGL 4.x/GLSL 4.x0	33
What's Behind These Developments?	34
OpenGL ES	34
How Can You Respond to These Changes?	35
Our Approach in this Book	36
Variable Name Convention	36
Exercises	37
3. Fundamental Shader Concepts	39
Shaders in the Graphics Pipeline	39
Vertex Shaders	42
Fragment Shaders	47
Tessellation Shaders	50
Geometry Shaders	53

The GLSL Shading Language	54
Passing Data from Your Application into Shaders	59
Defining Attribute Variables in Your Application	59
Defining Uniform Variables in Your Application	62
A Convenient Way to Transition to the Newer Versions of GLSL	64
Exercises	67
4. Using glman	69
Using glman	71
Loading a GLIB File	72
Editing GLIB and Shader Source Files	72
GLIB Scene Creation	72
Window and Viewing	73
Transformations	73
Defining Geometry	73
Specifying Textures	76
Specifying Shaders	77
Miscellaneous	78
Specifying Uniform Variables	79
Examples of GLIB Files	81
More on Textures and Noise	82
Using Textures	82
Using Noise	84
Functions in the glman Interface Window	86
Generating and Displaying a Hardcopy of Your Scene	86
Global Scene Transformation	86
Eye Transformation	87
Object Picking and Transformation	87
Texture Transformation	88
Monitoring the Frame Rate	88
Miscellaneous	89
Exercises	90

Setting Up Lighting for Shading	131
Flat Shading	132
Smooth (Gouraud) Shading	133
Phong Shading	134
Anisotropic Shading	135
Exercises	137
7. Vertex Shaders	139
Vertex Shaders in the Graphics Pipeline	140
Input to Vertex Shaders	140
Output from Vertex Shaders	142
Fixed-Function Processing After the Vertex Shader	145
The Relation of Vertex Shaders to Tessellation Shaders	146
The Relation of Vertex Shaders to Geometry Shaders	146
Replacing Fixed-Function Graphics with Vertex Shaders	146
Standard Vertex Processing	147
Going Beyond the Fixed-Function Pipeline with Vertex Shaders	148
Vertex Modification	148
Issues in Vertex Shaders	151
Creating Normals	152
Summary	153
Exercises	154
8. Fragment Shaders and Surface Appearance	157
Basic Function of a Fragment Shader	158
Inputs to Fragment Shaders	158
Particularly Important “In” Variables for the Fragment Shader	161
Coordinate Systems	162

Fragment Shader Processing	163
Outputs from Fragment Shaders	163
Replacing Fixed-Function Processing with Fragment Shaders	163
Shading	164
Traditional Texture Mapping	165
False Coloring	166
What Follows a Fragment Shader?	168
Additional Shader Effects	169
Discarding Pixels	169
Phong Shading	169
Shading with Analytic Normals	170
Anisotropic Shading	172
Data-Driven Coloring	173
Images Using Other Data	175
Exercises	177
9. Surface Textures in the Fragment Shader	179
Texture Coordinates	180
Traditional Texture Mapping	180
GLSL Texture Mapping	182
The Texture Context	184
Texture Environments in the Fixed-Function World	185
Texture Sampling Parameters	186
Samplers	186
Procedural Textures	187
Bump Mapping	193
Cube Maps	200
Render to Texture	205
Render to Texture for Multipass Rendering in glman	209
Exercises	212

10. Noise	213
Fundamental Noise Concepts	214
Three Types of Noise: Value, Gradient, and Value+Gradient	214
Cubic and Quintic Interpolation	215
Noise Equations	216
Other Noise Concepts	220
Fractional Brownian Motion (FBM, 1/f, Octaves)	220
Noise in Two and Three Dimensions	221
Using Noise with glman	223
Using Noise with the Built-In GLSL Functions	225
Turbulence	225
Some Examples of Noise in Different Environments	228
Marble Shader	230
Cloud Shader	231
Wood Shader	233
Advanced Noise Topics	235
Using Noisegraph	235
Exercises	237
11. Image Manipulation with Shaders	239
Basic Concepts	240
Single-Image Manipulation	241
Luminance	241
CMYK Conversions	243
Hue Shifting	246
Image Filtering	248
Image Blurring	249
Chromakey Images	251
Stereo Anaglyphs	252
3D TV	256
Edge Detection	259
Embossing	260

Toon Shader	262
Artistic Effects	264
Image Flipping, Rotation, and Warping	265
The Image Blending Process	270
Blending an Image with a Constant Base Image	271
Color Negative	272
Brightness	273
Contrast	274
Blending an Image with a Version of Itself	275
Saturation	275
Sharpness	276
Blending Two Different Images	277
Other Combinations	278
Image Transitions	281
Notes	286
Exercises	287

12. Geometry Shader Concepts and Examples **291**

What Does the Geometry Shader Do?	292
New Adjacency Primitives	294
Layouts for Input and Output Variables	295
New OpenGL API Functions	296
New GLSL Variables and Variable Types	299
Communication between a Vertex or Tessellation Shader and a Geometry Shader	299
Normals in Geometry Shaders	301
Examples	301
Bézier Curves	301
Shrinking Triangles	303
Sphere Subdivision	305
3D Object Silhouettes	309
Exercises	312