

Index

Numbers

3D Computer Graphics, 366

A

accumulation buffer, 460, 462, 482–494
 clearing, 32, 463
 depth-of-field effect, use for, 489–493
 examples of use, 482
 full range for best results, use, 758
 motion blur, use for, 489
 sample program with depth-of-field effect, 490
 sample program with full-scene antialiasing, 484
 scene antialiasing, use for, 483

AGL, 14

airbrushing, 621

Akeley, Kurt, 482

aliasing, *see* antialiasing

alpha, 231
 destination alpha, 260
 material properties, 213
 multisampling coverage, 259
 texture image data type, 417

alpha blending, *see* blending

alpha test, 468
 querying current values, 469
 rendering pipeline stage, 14, 669

ambient
 contribution to lighting equation, 223
 global light, 208, 222
 light, 187, 188, 196
 material properties, 189, 213

animation, 20–23, 758

antialiasing, 247–260
 accumulation buffer used for, 483–488
 characters (by masking), 605
 characters (by texturing), 616

color-index mode, 252
 coverage values, 247
 enabling for points or lines, 249
 enabling for polygons, 260
 lines, 247, 249–255
 lines (by texturing), 616
 points, 249–255, 608
 polygons, 260
 RGBA mode, 249
 sample program in color-index mode, 252
 sample program in RGBA mode, 250
 scene, with the accumulation buffer, 483

Apple Interface to OpenGL, *see* AGL

ARB imaging subset, 342–364

architectural applications
 orthographic parallel projection, use of, 136

Architecture Review Board
 extensions, approved, 597

arcs, 516

array elements, *see* vertex arrays

aspect ratio
 perspective projection, 135
 viewport transformation, 139

atmospheric effects, *see* fog

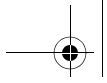
attenuation of light, 197–198

attribute groups, 90–93
 client, 91
 list of, 674–710
 multitexturing, with, 442
 performance tips, 760
 server, 91
 stack depth, obtaining, 91
 stacks, 91

auxiliary buffers, 461, 464

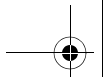
B

back-facing polygons, 56
 culling, 57

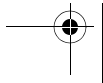
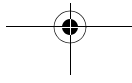


- back-facing polygons (*Continued*)
 - material property, specifying, 212
 - two-sided lighting, 209
 - background, 30–32
 - color, 30
 - drawing a fixed, 466, 617
 - background processing, 743
 - backward compatibility
 - tessellation, 514
 - basis functions, 527, 528
 - Bernstein
 - basis, 527
 - polynomial, 531
 - Bézier
 - basis, 527, 528
 - curve, 531
 - sample program using mesh for surface, 538
 - sample program which draws curve, 529
 - sample program which draws surface, 536
 - BGR and BGRA pixel formats, 315
 - billboarding, 239, 469
 - bitmaps, 302–308
 - display lists cache bitmap data, 283
 - distorting, 603
 - drawing, 306
 - feedback mode, 585
 - fonts, used for, 304, 310
 - imaging pipeline operations, 323
 - ordering of data in, 305
 - origin of, 307
 - sample program, 304
 - sample program that creates a font, 311
 - size of, 305
 - bitplanes, 170, 458
 - displayable colors, number of, 172
 - blending, 231–244, 479
 - antialiasing polygons, 260
 - coverage calculations for antialiasing, 247
 - destination alpha, 260
 - differences among releases, 230
 - enabling, 235
 - enabling for antialiasing, 249
 - equation, 235
 - factors (source and destination), 232
 - images, 608
 - ordering polygons before drawing, 243
 - rendering pipeline stage, 14, 669
 - sample program for three-dimensional, 244
 - sample program with blended polygons, 241
 - texture function, 420
 - three dimensions, in, 243
 - uses of, 238
 - buffer object, 817
 - buffer objects, 82
 - buffer, *see* framebuffer
- C**
- C programming language, 8
 - CAD/CAM, *see* computer-aided design
 - camera analogy, 106–107
 - environment mapping, 435
 - viewport transformations, 138
 - capping, *see* computational solid geometry
 - Chapter, 47
 - characters
 - antialiasing, 616
 - circles, 516
 - clearing the framebuffer, 30–32, 462–463
 - affected by scissoring, dithering, and masking, 463, 670
 - performance tips, 761
 - client-server, *see* networked operation
 - clip coordinates, 108, 150
 - feedback mode, 585
 - clipping, 138
 - interference regions found using clipping planes, 612
 - overview, 105
 - primitives in rendering pipeline, 12
 - viewing volume, 134
 - clipping planes
 - additional clipping planes, 108, 149–152
 - depth-buffer resolution, effect on, 758
 - far, 134–137, 142

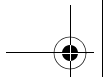




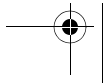
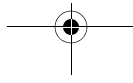
- near, 134–137, 142
- querying number of additional, 150
- sample program with additional clipping planes, 151
- color
 - alpha values, 231
 - background, 30
 - cube showing blended RGB values, 169
 - current raster color, 308
 - human perception, 167
 - RGBA values for, 33, 168
 - specifying, 33
 - specifying for tessellation, 504
 - specifying in color-index mode, 178
 - specifying in RGBA mode, 177
- color buffer, 168, 170, 459, 460, 461
 - clearing, 32
 - masking, 465
- color map, 168, 173
 - loading for antialiasing, 252
 - loading for smooth shading, 181
 - loading, using GLUT, 741
 - size of, 174
- color matrix, 357–358
 - example, 357
 - post transform scale and bias, 358
 - sample program, 358
- color sum mode, 450
- color tables, 344–346
 - proxies, 349
 - replacing part of, 348
 - sample program, 346
 - specifying, 344
- color-index mode, 173–175
 - changing between RGBA mode and, 176
 - choosing between RGBA mode and, 175
 - coverage calculations for antialiasing, 247
 - dithering, 480
 - layering with writemasks, 466
 - lighting, 226–228
 - lighting calculations in, 227
 - texturing limitations, 371, 378
 - vertex arrays, specifying values with, 69
- combiner functions, 444–449
- command syntax, 7–9
- compositing images, 239
- compositing transformations, 152–159
- computational solid geometry
 - capping, 475
 - difference of several contours, 509
 - interference regions, 611
 - intersection of two contours, 509
 - union of several contours, 509
- Computer Graphics: Principles and Practice*, xxii, 167, 171, 751
- computer-aided design
 - orthographic parallel projection, use of, 136
- concave polygons
 - GLU tessellation, 498
 - stencil buffer, drawing with the, 610
- cones, 516, 742
 - improving rendering of, 619
- constant attenuation, 198
- contours, 430
- control points, 526, 530, 534, 543
- convex polygons, 39
- convolutions, 349–357
 - 1d filters, 355
 - 2d filters, 349
 - border modes, 356
 - post convolution scale and bias, 357
 - sample program, 351
 - separable filters, 353
- Conway, John, 619
- coordinate systems
 - grand, fixed, 119, 128, 153
 - local, 119, 128, 153, 157
 - simple 2D, 36
- coordinates
 - see* clip coordinates, depth coordinates, eye coordinates, homogeneous coordinates, normalized device coordinates, object coordinates, q texture coordinates, texture coordinates, w coordinates, or window coordinates
- coverage, pixel, 247



- Coxeter, H. S. M., 751
 cross product, 131, 747
 CSG, *see* computational solid geometry
 cube maps, 436
 culling, 56–57
 enabling, 57
 rendering pipeline stage, 12, 668
 selection mode, 566
 curves and curved surfaces, 41
 see also evaluators or NURBS
Curves and Surfaces for Computer-Aided Geometric Design, 527
 cylinders, 516
- D**
- data types
 RGBA color conversion, 177
 special OpenGL, 8
 texture data, 377
 warning about data type conversions, 759
 decals, 469, 609
 polygon offset used for, 274
 texture function, 419
 depth buffer, 186, 460, 461
 see also hidden-surface removal
 background, using masking for a common, 466
 blending, use for three-dimensional, 243
 clearing, 32, 186, 463
 decals, for, 609
 Dirichlet domains, for, 618
 drawing static backgrounds, 617
 masking, 465
 near frustum plane effect on resolution, 758
 pixel data, 321, 332
 depth coordinates, 109, 141
 perspective division, 141
 picking use, 577
 polygon offset, 274–276
 rendering pipeline stage for depth-range operations, 12, 668
 sample program with picking, 577
 selection hit records, 566
 depth test, 475
 see also depth buffer
 rendering pipeline stage, 14, 669
 depth-cuing, *see* fog
 depth-of-field effect, 489–493
 sample program, 490
 destination factor, *see* blending
 diffuse
 contribution to lighting equation, 223
 light, 188, 196
 material properties, 189, 213
 directional light source, 197
 Dirichlet domains, 618
 disks, 516
 display lists, 29, 279
 changing mode settings, 299
 compiling, 287
 creating, 285
 deleting, 292
 disadvantages, 284, 290
 error handling, 286, 758
 executing, 285, 289–290
 executing multiple, 292
 font creation, 293, 309
 hierarchical, 290
 immediate mode, mixing with, 290
 indices for, obtaining, 286
 naming, 286
 nesting, 290
 nesting limit, querying, 290
 networked operation, 289
 performance tips, 760
 querying use of an index, 291
 rendering pipeline stage, 11
 sample program creating a font, 294
 sample program for creating, 279, 285
 sharing among rendering contexts, 717, 732
 state variables saved and restored, 298
 tessellation, use with, 513
 uses for, 283, 299
 vertex-array data, 289
 what can be stored in, 289



- distorted images, 603
 - texture images, 423
- dithering, 172–173, 480, 758
 - and clearing, 463
 - rendering pipeline stage, 14, 669
- dot product
 - lighting calculations, use in, 223
 - texture combiner function, 446
- double-buffering, 22–23
 - automatic glFlush(), 35
 - changing between single-buffering and, 176
 - object selection using the back buffer, 602
 - querying its presence, 461
 - sample program, 24
- drawing
 - clearing the window, 30
 - forcing completion of, 34
 - icosahedron, 95
 - points, 43
 - polygons, 43, 56
 - preparing for, 30
 - rectangles, 40
 - spheres, cylinders, and disks, 515–524
- drawing pixel data, *see* pixel data
- Duff, Tom, 239
- E**
- edge flags, 62–63
 - tessellated polygons generate, 502
 - vertex arrays, specifying values with, 69
- emission, 188, 214, 221
- enabling
 - alpha test, 468
 - antialiasing of points or lines, 249
 - antialiasing polygons, 260
 - blending, 235
 - color material properties mode, 217
 - color sum mode, 450
 - culling, 57
 - depth test, 476
 - dithering, 173, 480
 - evaluators, 531, 535
 - fog, 261
 - lighting, 211
 - line stippling, 52
 - logical operations, 481
 - multisampling, 256
 - normal vectors for evaluated surfaces,
 - automatic generation of, 535, 543
 - polygon offset, 274
 - polygon stippling, 58
 - rescaling normals, 65, 192
 - stencil test, 470
 - texture coordinate generation, 433
 - texturing, 372, 375
 - unit length normal vectors ensured, 65, 192
- endianness, 327
- environment mapping, 434, 436
- errata, xxiv
- error handling, 593–595
 - error string description, 595
 - recommended usage, 758
- evaluators, 528–542
 - basis functions, 527, 531
 - evenly spaced values, 533, 537
 - one-dimensional, 528
 - rendering pipeline stage, 11
 - sample program using mesh for 2D Bézier surface, 538
 - sample program which draws 1D Bézier curve, 529
 - sample program which draws 2D Bézier surface, 536
 - sample program which generates texture coordinates, 540
 - tessellation usage, 760
 - texture coordinates, generating, 540
 - two-dimensional, 536
- event management, using GLUT, 19
- example programs, *see* programs
- extensions
 - Architecture Review Board approved, 597
 - Microsoft Windows and
 - wglGetProcAddress(), 599
 - vendor-specific, 597
- eye coordinates, 108, 150
 - texture coordinate generation, 429, 434



F

- fade effect, 600
- Farin, Gerald E., 527
- feedback, 583–590
 - array contents, 589
 - pass-through markers, 586
 - querying current rendering mode, 564
 - returned data, 585
 - sample program, 587
 - steps to perform, 584
 - tessellation, obtaining vertex data after, 513
- Feiner, Steven K., xxii, 751
- field of view, 112
 - calculate, using trigonometry to, 143
- filtering, 406–408
 - mipmapped textures, 396–406, 408
 - texture border colors, 424
- flat shading, 179
- flight simulation
 - fog, use of, 261
- flushing, 34, 758
- fog, 261–271
 - blending factors, 264
 - color-index mode, 266
 - density, 265
 - enabling, 261
 - equations, 264
 - fog coordinates, 268
 - hints, 261
 - RGBA mode, 265
 - sample program in color-index mode, 266
 - sample program in RGBA mode, 262
 - sample program with fog coordinates, 269
- Foley, James D., xxii, 167, 171, 751
- fonts, 309–312
 - antialiased characters (by masking), 605
 - antialiased characters (by texturing), 616
 - bitmapped, 310
 - creating with display lists, 293
 - drawing, 308
 - drawing as bitmaps, 304
 - multi-byte, 310
 - same program, 311
 - sample program using multiple display lists, 294
 - X fonts, using, 718
- Foran, Jim, 452
- foreshortening, perspective, 133
- fragments, 458
 - alpha test, 468
 - blending, 231
 - depth test, 475
 - rendering pipeline operations, 13, 669
 - scissor test, 468
 - tests, 467–481
 - texture functions, 418
- framebuffer, 170, 459
 - capacity per pixel, 460
 - clearing, 462–463
 - copying pixel data within, 313, 321, 322
 - enabling for reading, 464
 - enabling for writing, 464
 - minimum configuration with the X Window System, 460
 - querying color resolution, 170
 - reading pixel data from, 313, 315
 - writing pixel data to, 313, 319
- front-facing polygons, 56
 - specifying material property for, 212
 - two-sided lighting, 209
- frustum, 133
- ftp (file-transfer protocol) site
 - GLX specification, 716
- Fundamentals of Computer Aided Geometric Design*, 527

G

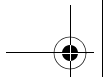
- Game of Life, 619
- gamma correction, 171
- Gardner, Martin, 619
- geometric primitives, 37–48, 666–668
 - performance when specifying, 761
 - rendering pipeline stage, 12
- geosciences
 - use of texturing in applications, 429

- giraffe, 174
- GL_VERTEX_PROGRAM_POINT_SIZE, 661
- GL_VERTEX_PROGRAM_TWO_SIDE, 661
- glAccum(), 482
- glActiveTextureARB*(), 440
- glAlphaFunc(), 469
- glAreTexturesResident(), 414
- glArrayElement(), 72
legal between glBegin() and glEnd(), 47
- Glassner, Andrew S., xxiii
- glAttachShader(), 634
- glBegin(), 42, 43, 502
restrictions, 46
- glBeginQuery(), 478
- glBindBuffer(), 84
- glBindTexture(), 375, 411
multitexturing, 440
- glBitmap(), 303, 307
feedback mode, 585
fonts, used for, 310
imaging pipeline operations, 323
pixel-storage modes effect, 326
- glBlendColor*(), 235
- glBlendEquation(), 235
- glBlendEquationSeparate(), 235
- glBlendFunc(), 233
- glBlendFuncSeparate(), 233
- glBufferData(), 84
- glBufferSubData(), 88
- glCallList(), 282, 285, 289
legal between glBegin() and glEnd(), 47
- glCallLists(), 292
fonts, use for, 309
legal between glBegin() and glEnd(), 47
sample program, 311
- glClear(), 31, 463, 670
depth buffer, clearing the, 186
- glClearAccum(), 32, 463
- glClearColor(), 31, 463
- glClearDepth(), 31, 463
- glClearIndex(), 32, 179, 463
fog, use with, 266
- glClearStencil(), 32, 463
- glClientActiveTextureARB*(), 443
- glClipPlane(), 150
- glColor*(), 33, 177
legal between glBegin() and glEnd(), 46
- glColorMask(), 463, 465
- glColorMaterial(), 217
performance tips, 760
- glColorPointer(), 69
- glColorSubTable(), 348
- glColorTable(), 344
- glColorTableParameter(), 346
- glCompileShader(), 633
- glCompressedTexImage1D(), 393
- glCompressedTexImage2D(), 393
- glCompressedTexImage3D(), 393
- glCompressedTexSubImage1D(), 394
- glCompressedTexSubImage2D(), 394
- glCompressedTexSubImage3D(), 394
- glConvolutionFilter1D(), 355
- glConvolutionFilter2D(), 350
- glConvolutionParameter*(), 356
- glCopyColorSubTable(), 348
- glCopyColorTable(), 347
- glCopyConvolutionFilter1D(), 355
- glCopyConvolutionFilter2D(), 353
- glCopyPixels(), 313, 321
alternative uses, 620
dithering, turn off, 758
feedback mode, 585
glReadBuffer() effect, 465
imaging pipeline operations, 322
pixel-transfer modes effect, 330
- glCopyTexImage1D(), 386
glReadBuffer() effect, 465
pixel-transfer modes effect, 330
- glCopyTexImage2D(), 379
glReadBuffer() effect, 465
pixel-transfer modes effect, 330
- glCopyTexSubImage1D(), 387
glReadBuffer() effect, 465
pixel-transfer modes effect, 330

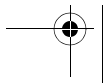
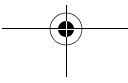
- glCopyTexSubImage2D(), 384
 - glReadBuffer() effect, 465
 - pixel-transfer modes effect, 330
- glCopyTexSubImage3D(), 390
 - pixel-transfer modes effect, 330
- glCreateProgram(), 634
- glCreateShader(), 632
- glCullFace(), 57
- glDeleteBuffers(), 90
- glDeleteLists(), 292, 310
- glDeleteProgram(), 637
- glDeleteQueries(), 479
- glDeleteShader(), 637
- glDeleteTextures(), 413
- glDepthFunc(), 476
- glDepthMask(), 465
 - blending opaque and translucent objects, 244
- glDepthRange(), 141
 - gluUnProject(), relationship to, 160
 - glWindowPos*() effect, 306
- glDetachShader(), 634
- glDisable(), 10, 49
- glDisableClientState(), 68
- glDisableVertexAttribArray(), 660
- glDrawArrays(), 77
- glDrawBuffer(), 320, 321, 464
- glDrawBuffers(), 464
- glDrawElements(), 73
- glDrawPixels(), 313, 319, 472, 617
 - alternative uses, 620
 - feedback mode, 585
 - pixel-storage modes effect, 326
 - pixel-transfer modes effect, 330
- glDrawRangeElements(), 76
 - version, 29
- glEdgeFlag*(), 63
 - legal between glBegin() and glEnd(), 47
- glEdgeFlagPointer(), 69
- glEnable(), 49, 193
 - see also enabling
- glEnableClientState(), 47, 67
- glEnableVertexAttribArray(), 660
- glEnd(), 42, 43, 502
 - restrictions, 46
- glEndList(), 281, 285, 287
- glEndQuery(), 478
- glEvalCoord*(), 533, 535
 - legal between glBegin() and glEnd(), 47
 - used instead of glVertex*(), 528, 531
- glEvalMesh*(), 534, 537
- glEvalPoint*()
 - legal between glBegin() and glEnd(), 47
- glFeedbackBuffer(), 584
 - glRenderMode(), use with, 564
- glFinish(), 35
- glFlush(), 35, 758
- glFog*(), 265
- glFogCoord*(), 269, 272
- glFrontFace(), 57
- glFrustum(), 113, 134, 670
- glGenBuffers(), 83
- glGenLists(), 281, 286
 - fonts, use for, 309
- glGenQueries(), 477
- glGenTextures(), 375, 410
- glGetAttachedShaders(), 672
- glGetAttribLocation(), 658
- glGetBooleanv(), 10, 49, 674
 - double-buffering support, querying, 461
 - stereo support, querying, 461
- glGetBufferParameteriv(), 672
- glGetBufferPointerv(), 672
- glGetBufferSubData(), 672
- glGetClipPlane(), 672
- glGetColorTable(), 672
 - pixel-storage modes effect, 326
- glGetColorTableParameter*(), 672
- glGetCompressedTexImage(), 672
- glGetConvolutionFilter(), 672
 - pixel-storage modes effect, 326
- glGetConvolutionParameter*(), 672
- glGetDoublev(), 10, 49, 674

- glGetError(), 10, 594, 672
- glGetFloatv(), 10, 49, 674
 - line width attributes, obtaining, 51
- glGetHistogram(), 360, 672
 - pixel-storage modes effect, 326
- glGetHistogramParameter*(), 672
- glGetIntegerv(), 10, 49, 674
 - alpha test information, obtaining, 469
 - attribute stack depth, obtaining, 91
 - clipping planes, obtaining number of additional, 150
 - color resolution, obtaining, 170
 - display list nesting limit, obtaining, 290
 - matrix stack depth, obtaining, 148
 - maximum texture size, obtaining, 380
 - name stack depth, obtaining, 565
 - pixel map information, obtaining, 333
 - rendering mode, obtaining current, 564
 - stencil-related values, obtaining, 471
 - vertex array range values, obtaining, 76
- glGetLight*(), 10, 673
- glGetMap*(), 673
- glGetMaterial*(), 673
- glGetMinmax(), 363, 673
 - pixel-storage modes effect, 326
- glGetMinmaxParameter*(), 673
- glGetPixelMap(), 673
- glGetPointerv(), 10, 49, 674
- glGetPolygonStipple(), 10, 673
- glGetProgramInfoLog(), 635
- glGetProgramiv(), 673
- glGetProgramLogInfo(), 673
- glGetQueryiv(), 673
- glGetQueryObjectiv(), 478, 673
- glGetQueryObjectuiv(), 478, 673
- glGetSeparableFilter(), 673
 - pixel-storage modes effect, 326
- glGetShaderInfoLog(), 633, 673
- glGetShaderiv(), 673
- glGetShaderSource(), 673
- glGetString, 598
- glGetString(), 595, 673
- glGetTexEnv*(), 673
- glGetTexGen*(), 673
- glGetTexImage(), 673
 - pixel-storage modes effect, 326
 - pixel-transfer modes effect, 330
- glGetTexLevelParameter*(), 381, 673
- glGetTexParameter*(), 673
 - texture residency, obtaining, 414
- glGetUniform*(), 674
- glGetUniformLocation(), 646
- glGetVertexAttrib*(), 674
- glGetVertexAttribPointerv(), 674
- glHint(), 248
 - fog use, 261
 - texture use, 375
- glHistogram(), 359
- glIndex*(), 178
 - fog, use with, 266
 - legal between glBegin() and glEnd(), 46
- glIndexMask(), 463, 465
- glIndexPointer(), 69
- glInitNames(), 563, 564, 565
- glInterleavedArrays(), 79
- glIsBuffer(), 83, 674
- glIsEnabled(), 10, 49, 674
- glIsList(), 291, 674
- glIsProgram(), 638, 674
- glIsQuery(), 477, 674
- glIsShader(), 638, 674
- glIsTexture(), 410, 674
- glLight*(), 193, 194, 195, 200
- glLightModel*(), 208
- glLineStipple(), 52
- glLineWidth(), 51
- glLinkProgram(), 635
- glListBase(), 292
 - fonts, use for, 309
 - sample program, 311
- glLoadIdentity(), 113, 115, 125, 670
 - performance tips, 760
 - viewing transformations, use before, 111

- glLoadMatrix*(), 114, 115, 116, 670
- glLoadName(), 564, 566
- glLoadTransposeMatrix*(), 114, 116
- glLogicOp(), 236, 481
- glMap*(), 530, 532, 535
- glMapBuffer(), 89
- glMapGrid*(), 533, 537
- glMaterial*(), 194, 212
 - legal between glBegin() and glEnd(), 46
 - performance tips, 760
- glMatrixMode(), 113, 115
 - use with matrix stacks, 146
- glMinmax(), 362
- glMultiDrawArrays(), 78
 - version, 29
- glMultiDrawElements(), 75
 - version, 29
- glMultiTexCoord*ARB(), 442
- glMultMatrix*(), 114, 115, 670
 - performance tips, 760
- glMultTransposeMatrix*(), 114, 116
- glNewList(), 281, 285, 287
- glNormal*(), 64
 - legal between glBegin() and glEnd(), 46
- glNormalPointer(), 69
- glOrtho(), 137, 670
 - picking matrix use, 571
- glPassThrough(), 584, 586
- glPixelMap*(), 333
- glPixelStore*(), 326, 390
 - cannot be stored in display lists, 289
 - polygon stippling, 58
 - texture image data, effect on, 377, 379, 382, 384, 386, 389, 394
- glPixelTransfer*(), 330, 617
 - texture image data, effect on, 377, 379, 382, 384, 386, 389, 394
- glPixelZoom(), 334, 603
- glPointParameter*(), 272
- glPointSize(), 50, 661
- glPolygonMode(), 56
 - antialiasing, effect on, 260
 - polygon offset, use with, 274
- glPolygonOffset(), 274
- glPolygonStipple(), 58
 - pixel-storage modes effect, 326
- glPopAttrib(), 10, 91, 298, 442, 675
- glPopClientAttrib(), 10, 93, 442, 675
- glPopMatrix(), 146, 157, 203, 298
 - restore orientation of coordinate systems, 159
 - selection, use with, 563
- glPopName(), 564, 565
- glPrioritizeTextures(), 415
- glPushAttrib(), 10, 91, 298, 442, 675
- glPushClientAttrib(), 10, 93, 442, 675
- glPushMatrix(), 146, 157, 203, 298
 - save orientation of coordinate systems, 159
 - selection, use with, 563
- glPushName(), 563, 564, 565
- glRasterPos*(), 303, 305
 - images, for positioning, 313
 - multitexturing, with, 442
 - selection hit, 566
- glReadBuffer(), 320, 321, 465
- glReadPixels(), 313, 315
 - glReadBuffer() effect, 465
 - pixel-storage modes effect, 326
 - pixel-transfer modes effect, 330
- glRect*(), 40
- glRenderMode(), 563, 564, 566, 584
- glResetHistogram(), 362
- glResetMinmax(), 364
- glRotate*(), 122, 153, 156, 670
 - performance tips, 760
- glSampleCoverage(), 259
- glScale*(), 111, 122, 156, 670
 - performance tips, 760
- glScissor(), 468
- glSecondaryColor*(), 450
- glSelectBuffer(), 563, 564
 - display lists, cannot be stored in, 289
- glSeparableFilter2D(), 354



- glShadeModel(), 179
- glShaderSource(), 633
- glStencilFunc(), 470
- glStencilFuncSeparate(), 470
- glStencilMask(), 465
- glStencilMaskSeparate(), 465
- glStencilOp(), 470
- glStencilOpSeparate(), 470
- glTexCoord*(), 375, 421
 - legal between glBegin() and glEnd(), 47
 - texture unit 0, for, 442
- glTexCoordPointer(), 69
- glTexEnv*(), 375, 416, 444
 - level of detail bias, 400
 - multitexturing, 440
- glTexGen*(), 429
 - cube maps, 438
 - environment mapping, 435
 - multitexturing, 440, 443
- glTexImage1D(), 385
 - pixel-storage modes effect, 326
 - pixel-transfer modes effect, 330
- glTexImage2D(), 375, 376
 - cube map textures, 436
 - pixel-storage modes effect, 326
 - pixel-transfer modes effect, 330
 - specifying mipmaps, 397
- glTexImage3D(), 388
 - pixel-storage modes effect, 326
 - pixel-transfer modes effect, 330
- glTexParameter*(), 375, 427
 - automatic mipmap regeneration, 406
 - mipmap level of detail, controlling, 403
 - mipmap levels, controlling base and maximum, 402
 - multitexturing, 440
 - specifying filtering methods, 407
- glTexSubImage1D(), 386
 - pixel-storage modes effect, 326
 - pixel-transfer modes effect, 330
- glTexSubImage2D(), 382
 - pixel-storage modes effect, 326
 - pixel-transfer modes effect, 330
- glTexSubImage3D(), 389
 - pixel-storage modes effect, 326
 - pixel-transfer modes effect, 330
- glTranslate*(), 121, 153, 156, 670
 - performance tips, 760
- GLU, 3, 14, 498
 - drawing spheres, cylinders, and disks, 515–524
 - error string description, 595
 - obsolete routines
 - gluBeginPolygon(), 514
 - gluEndPolygon(), 514
 - gluNextContour(), 514
 - quadrics, 515–524
 - tessellation, 39, 498–515
 - version numbers, obtaining, 596
- gluBeginCurve(), 543, 553
- gluBeginSurface(), 543, 551
- gluBeginTrim(), 557
- gluBuild1DMipmapLevels(), 405
- gluBuild1DMipmaps(), 405
- gluBuild2DMipmapLevels(), 405
- gluBuild2DMipmaps(), 405
- gluBuild3DMipmapLevels(), 405
- gluBuild3DMipmaps(), 405
- gluCheckExtension(), 598
- gluCylinder(), 516, 519
- gluDeleteNurbsRenderer(), 547
- gluDeleteQuadric(), 516, 517
- gluDeleteTess(), 513, 514
- gluDisk(), 516, 520
- gluEndCurve(), 543, 553
- gluEndSurface(), 543, 551
- gluEndTrim(), 557
- gluErrorString(), 517, 551, 595
 - polygon tessellation, 502
- gluGetNurbsProperty(), 550, 674
- gluGetString, 598
- gluGetString(), 597, 674
- gluGetTessProperty(), 509, 674
- gluLoadSamplingMatrices(), 550
- gluLookAt(), 109, 111, 129, 153



- gluNewNurbsRenderer(), 543, 547
- gluNewQuadric(), 516, 517
- gluNewTess(), 500, 514
- glUniform*(), 646
- glUniformMatrix*(), 646
- glUnmapBuffer(), 89
- gluNurbsCallback(), 543, 551, 554
- gluNurbsCallbackData(), 555
- gluNurbsCurve(), 543, 553
- gluNurbsProperty(), 543, 548
 - returning tessellated data, 553
- gluNurbsSurface(), 543, 552
- gluOrtho2D(), 138, 759
 - resized windows, use with, 36
- gluPartialDisk(), 516, 520
- gluPerspective(), 113, 136, 153
 - picking matrix use, 571
- gluPickMatrix(), 571
- gluProject(), 163
- gluPwlCurve(), 557
- gluQuadricCallback(), 516, 517
- gluQuadricDrawStyle(), 516, 517
- gluQuadricNormals(), 516, 518
- gluQuadricOrientation(), 516, 518
- gluQuadricTexture(), 516, 518
- gluScaleImage(), 378
- glUseProgram(), 635
- gluSphere(), 516, 519
- GLUT, 15, 737–743
 - basic functions, 16–20
 - event management, 19
 - glutCreateWindow(), 17, 739
 - glutDisplayFunc(), 17, 739
 - glutIdleFunc(), 20, 743
 - glutInit(), 17, 738
 - glutInitDisplayMode(), 17, 738
 - glutInitWindowPosition(), 17, 739
 - glutInitWindowSize(), 17, 739
 - glutKeyboardFunc(), 19, 740
 - glutMainLoop(), 18, 743
 - glutMotionFunc(), 19, 740
 - glutMouseFunc(), 19, 740
 - glutPostRedisplay(), 18, 282, 741
 - glutReshapeFunc(), 19, 740
 - simple example, 36
 - glutSetColor(), 17, 179, 227, 741
 - smooth shading, use for, 181
 - glutSolidCone(), 742
 - glutSolidCube(), 20, 742
 - glutSolidDodecahedron(), 742
 - glutSolidIcosahedron(), 742
 - glutSolidOctahedron(), 742
 - glutSolidSphere(), 20, 741
 - glutSolidTeapot(), 742
 - glutSolidTetrahedron(), 742
 - glutSolidTorus(), 742
 - glutSwapBuffers(), 23
 - glutWireCone(), 742
 - glutWireCube(), 20, 742
 - glutWireDodecahedron(), 742
 - glutWireIcosahedron(), 742
 - glutWireOctahedron(), 742
 - glutWireSphere(), 20, 153, 741
 - glutWireTeapot(), 742
 - glutWireTetrahedron(), 742
 - glutWireTorus(), 742
 - multisampling, 256
 - sample program introducing GLUT, 18
 - window management, 17, 36
- gluTessBeginContour(), 511
- gluTessBeginPolygon(), 510
- gluTessCallback(), 501, 511, 514
- gluTessEndContour(), 511
- gluTessEndPolygon(), 510
- gluTessNormal(), 509, 510, 513
- gluTessProperty(), 506, 511
- gluTessVertex(), 511, 514
- gluUnProject(), 160, 163
- gluUnProject4(), 163
- glValidateProgram(), 638
- glVertex*(), 41
 - legal between glBegin() and glEnd(), 46
 - using glEvalCoord*() instead, 528
- glVertexAttrib*(), 658
- glVertexAttrib4N*(), 658
- glVertexAttribPointer(), 659

glVertexPointer(), 47, 69
 glViewport(), 114, 139
 using with resized windows, 36
 glWindowPos*(), 306
 multitexturing, with, 442
 selection hit, 566
 GLX, 14, 716
 ftp site for GLX specification, 716
 glXChooseFBConfig(), 716, 719
 glXChooseVisual(), 716, 721, 762
 glXCopyContext(), 717, 720
 glXCreateContext(), 718, 721
 glXCreateGLXPixmap(), 716, 721
 glXCreateNewContext(), 717, 720
 glXCreatePbuffer(), 716, 720
 glXCreatePixmap(), 716, 720
 glXCreateWindow(), 716, 720
 glXDestroyContext(), 717, 720
 glXDestroyGLXPixmap(), 721
 glXDestroyPbuffer(), 719, 721
 glXDestroyPixmap(), 719, 721
 glXDestroyWindow(), 719, 721
 glXGetClientString(), 716, 719
 glXGetConfig(), 460, 716, 721
 glXGetCurrentContext(), 717, 720
 glXGetCurrentDisplay(), 717, 720
 glXGetCurrentDrawable(), 717, 720
 glXGetCurrentReadDrawable(), 720
 glXGetFBConfigAttrib(), 716, 719
 glXGetFBConfigs(), 719
 glXGetSelectedEvent(), 718, 720
 glXGetVisualFromFBConfig(), 716, 719
 glXIsDirect(), 717, 720
 glXMakeContextCurrent(), 717, 720
 glXMakeCurrent(), 718, 721
 glXQueryContext(), 717, 720
 glXQueryExtension(), 716, 719
 glXQueryExtensionsString(), 716, 719
 glXQueryServerString(), 716, 719
 glXQueryVersion(), 716, 719
 glXSelectEvent(), 718, 720
 glXSwapBuffers(), 23, 718, 721
 glXUseXFont(), 718, 721
 glXWaitGL(), 718, 721
 performance tips, 762
 glXWaitX(), 718, 721
 performance tips, 762

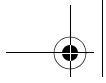
glXGetCurrentReadDrawable(), 717, 718
 glXQueryExtensionString(), 597
 Gouraud shading, *see* smooth shading

H

Haerberli, Paul, 452, 482
 haze, *see* fog
 header file, 15
 hidden-line removal, 614
 polygon offset used for, 274
 hidden-surface removal, 185–187, 475
 hierarchical models, 145, 290
 picking, 575–577
 highlights, *see* specular
 hints, 248
 fog, 261
 perspective correction, 248, 375
 histogram, 359–362
 resetting, 360, 362
 retrieving, 360
 sample program, 360
 hits (selection), *see* selection (hit records)
 holes in polygons, 39, 611
 homogeneous coordinates, 38, 752
 Hoschek, Josef, 527
 Hughes, John F., xxii, 751

I

IBM OS/2 Presentation Manager to OpenGL
 Interface, *see* PGL
 icosahedron, drawing, 95
 identity matrix, 111, 115, 125, 760
 illumination, *see* lighting
 images, 302, 312–321
 see also pixel data
 blending, 608
 compositing, 232
 distorted, 603
 imaging pipeline, 321–337

images (*Continued*)

- interpolating between, 608
- magnifying or reducing, 334
- nonrectangular, 239
- projecting, 616
- sample code which draws an image, 320
- sample program which draws, copies, and zooms an image, 335
- scaling and rotating, 616
- sources of, 312
- superimposing, 609
- transposing, 621
- warping, 616
- imaging pipeline, *see* images (imaging pipeline)
- imaging subset, 342–364
 - extension string, 598
 - texture images, effect on, 378, 385
- immediate mode, 29, 278
 - display lists, mixing with, 289
- infinite light source, 197
- input events
 - handling, using GLUT, 19
- intensity
 - texture image data type, 417
- Interactive Inspection of Solids: Cross-sections and Interferences*, 611
- interference regions, 611
- interleaved arrays, 78
- interpolating
 - color values and texture coordinates, 248, 420
 - texture combiner function, 448
- invariance
 - of an OpenGL implementation, 758, 763

J

- jaggies, 247
- jittering, 483, 488, 494
 - accFrustum() routine, 484
 - accPerspective() routine, 484
 - sample code to jitter projection transformations, 484

- sample program with orthographic projection, 488

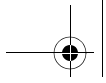
K

- Kilgard, Mark, xxiii, 15, 716, 737
- Korobkin, Carl, 452

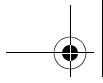
L

- Lasser, Dieter, 527
- layers, drawing, 604
- Life, Game of, 619
- light sources, 194–207
 - ambient light, 188, 196
 - contribution to lighting equation, 222
 - diffuse light, 188, 196
 - directional, 197
 - display lists cache values, 283
 - infinite light source, 197
 - local light source, 197
 - maximum number of sources, 193
 - moving along with the viewpoint, 205
 - moving light sources, 201–206
 - multiple light sources, 200
 - performance tips, 193
 - positional, 197
 - rendering pipeline stage, 12, 667
 - RGBA values, 189
 - sample program that moves the light source, 204
 - specifying a light source, 193
 - specular light, 188
 - spotlights, 199–200
 - stationary, 202
- lighting
 - see also* light sources, material properties
 - ambient light, 187
 - approximation of the real world, 187
 - attenuation, 197–198
 - calculations in color-index mode, 227
 - color-index mode, 226–228
 - default values, using, 194

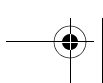


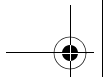


- display lists cache values, 283
- enabling, 193, 194
- enabling and disabling, 211
- equation that calculates lighting, 221
- global ambient light, 208, 222
- lighting model, 207–210
- lighting model, specifying a, 193
- rendering pipeline stage, 12, 667
- sample program introducing lighting, 190
- specular color separated, 210, 225, 450
- steps to perform, 190
- two-sided materials, 209
- viewer, local or infinite, 209
- line segment, 38
- linear attenuation, 198
- lines, 38
 - antialiasing, 249–255, 616
 - connected closed loop, specifying, 43, 45
 - connected strip, specifying, 43, 45
 - feedback mode, 585
 - querying line width, 51
 - sample program with wide, stippled lines, 54
 - specifying, 43, 45
 - stippling, 52
 - tessellated polygons decomposed into, 502
 - width, 51
- local light source, 197
- logical operations
 - rendering pipeline stage, 14, 669
 - transposing images, using for, 621
- lookup table, *see* color map
- luminance, 315, 341
 - pixel data formats for, 317, 325
 - texture image data type, 417
- M**
 - magnifying images, 334
 - masking, 465
 - antialiasing characters, 607
 - layers, drawing, 604
 - rendering pipeline stage, 14, 669
 - material properties, 194, 211–220
 - ambient, 189, 213
 - changing a single parameter with `glColorMaterial()`, 217
 - changing material properties, 215
 - diffuse, 189, 213
 - display lists cache values, 283
 - emission, 188, 214, 221
 - enabling color material properties mode, 217
 - performance when changing, 760
 - rendering pipeline stage, 12, 667
 - RGBA values, 190
 - sample program which changes material properties, 215
 - sample program which uses `glColorMaterial()`, 218
 - shininess, 214
 - specular, 189, 214
 - two-sided lighting, 209
 - matrix
 - see also* matrix stack
 - choosing which matrix is current, 115
 - column-major ordering, 116
 - current, 111
 - danger of extensive changes, 758
 - display lists cache matrix operations, 283
 - identity, 111, 115, 125, 760
 - loading, 115
 - loading transposed, 116
 - modelview, 108, 115
 - multiplying matrices, 115
 - multiplying transposed matrices, 116
 - NURBS, specifying for sampling, 549
 - orthographic parallel projection, 756
 - perspective projection, 755
 - projection, 113, 115
 - rotation, 754
 - row-major ordering, 116
 - scaling, 754
 - texture, 451
 - transformation pipeline, 106
 - transformations of homogeneous coordinates, 752
 - translation, 754



- matrix stack, 145–149
 - choosing which matrix stack is current, 146
 - current matrix stack, 670
 - modelview, 148
 - popping, 146
 - projection, 148
 - pushing, 146
 - querying stack depth, 148
 - texture, 451
 - Megahed, Abe, 611
 - Microsoft
 - callback functions on Windows, 503
 - Microsoft Win32, *see* Win32
 - Microsoft Windows 95/98/NT, xxiv, 14, 731
 - Microsoft Windows to OpenGL interface, *see* WGL
 - minmax, 362–364
 - reseting, 363, 364
 - retrieving results, 363
 - sample program, 363
 - mipmapping, 396–406
 - automated generation, 404
 - base and maximum levels, 402
 - level of detail control, 401
 - minification filters, 408
 - texture objects for mipmaps, 413
 - mirroring objects, *see* scaling
 - modeling transformations, 111, 117, 120–125
 - camera analogy, 106
 - connection to viewing transformations, 111
 - example, 123
 - rotation, 122
 - rotation matrix, 754
 - sample program, 125
 - scaling, 122
 - scaling matrix, 754
 - translation, 121
 - translation matrix, 754
 - models
 - rendering wireframe and solid, 20, 741
 - modelview matrix, 108, 115
 - arbitrary clipping planes, effect on, 150
 - stack, 148
 - mosaicing, 401
 - motion blur, 489
 - stippling, with, 601
 - motion, *see* animation
 - movie clips, 620
 - multiple layers
 - displaying with overlap, 604
 - multisampling, 255–259
 - fading point primitives, 272
 - sample program, 256
 - multitexture
 - extension string, 598
 - multitexturing, 438–443
- N**
- name stack, 563–567
 - creating, 564
 - initializing, 564
 - loading, 564
 - multiple names, 575–577
 - popping, 564
 - pushing, 564
 - querying maximum depth, 565
 - networked operation, 34–35
 - attribute groups, saving and restoring, 91
 - display lists, 289
 - versions, 596
 - Non-Uniform Rational B-Splines, *see* NURBS
 - nonplanar polygons, 40
 - normal vectors, 63–65, 192
 - calculating, 746
 - calculating for analytic surfaces, 747
 - calculating for polygonal data, 749
 - calculating length, 65
 - cross product, calculating normalized, 97
 - enabling automatic unit length division, 65, 192
 - inverse matrix generated, 670
 - matrix transformations, 108





normalized, 65
 NURBS, generating for, 552
 quadrics, generated for, 518
 rendering pipeline stage, 12, 667
 specifying, 64
 tessellation, specifying for, 504
 transformations, 753
 uniform rescaling, 65
 unit length optimizes performance, 761
 vertex arrays, specifying values with, 69

normal, *see* normal vectors

normalized device coordinates, 108

NURB Curves and Surfaces, 527

NURBS, 542–560
 creating a NURBS curve or surface, 551–553
 creating a NURBS object, 547
 culling, 548
 deleting a NURBS object, 547
 display list use, 282
 error handling, 550
 method of display (lines or filled polygons), 548
 normal vectors, generating, 552
 properties, controlling NURBS, 547
 querying property value, 550
 references, 527
 sample program which draws a lit NURBS surface, 544
 sample program with a trimmed surface, 559
 sampling precision, 548
 source for matrices, 549
 steps to use, 543
 texture coordinate generation, 552
 trimming, 557–560

NURBS Book, The, 527

NURBS for Curve and Surface Design, 527

NURBS tessellator
 sample code, 555, 556

O

object coordinates, 108
 texture coordinate generation, 429

objects, *see* models

occlusion query, 476

opacity, 232

OpenGL Extension to the X Window System, *see* GLX

OpenGL Programming for the X Window System, xxiii

OpenGL Programming for the X Window System, 15, 17, 716, 737

OpenGL Reference Manual, xxiii

OpenGL Reference Manual, 665, 672, 716

OpenGL Utility Library, *see* GLU

OpenGL Utility Toolkit, *see* GLUT

orthographic parallel projection, 113, 136–137
 jittering, 487
 matrix, 756
 specifying with integer coordinates, 759

outer product, 353

outlined polygons, 56, 63
 polygon offset solution, 274

overlapping objects, 611

P

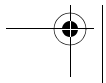
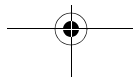
packed pixel data, 317–318

painting, 232, 238, 621

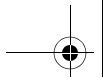
partial disks, 516

pass-through markers, 586

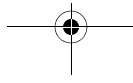
performance tips
 clearing the window, 32
 display lists, 282
 flat shading, 761
 flushing the pipeline, 34
 fog, 261
 GLX tips, 762
 hints, 248
 light source attenuation, effect of, 198
 light sources, effect of additional, 193
 list of general tips, 760
 material properties, changing, 760
 NURBS and display lists, 282
 pixel data alignment, 328

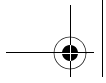


- performance tips (*Continued*)
 - pixel data, drawing, 341
 - polygon restrictions, 39
 - polygon subdivision, 94
 - pushing and popping attribute groups, 760
 - rasterization and fragment operations for pixel data, 761
 - removing hidden surfaces, 187
 - specifying geometric primitives, 761
 - tessellation and display lists, 282
 - tessellation, use of, 513
 - texture images, internal format of, 377
 - texture objects, 409, 760
 - texture subimages, 760
 - two-sided lighting, 210
 - unit-length normal vectors, 761
 - vector and scalar forms of commands, 761
 - vertex arrays, 761
- perspective projection, 133–136
 - correction hint, 248, 375
 - depth coordinates, effect on, 141
 - jittering, 484
 - matrix, 755
 - perspective division, 108
- PGL, 14, 727
 - `pglChooseConfig()`, 727, 729
 - `pglCopyContext()`, 728, 730
 - `pglCreateContext()`, 728, 730
 - `pglDestroyContext()`, 728, 730
 - `pglGetCurrentContext()`, 728, 730
 - `pglGetCurrentWindow()`, 728, 730
 - `pglGrabFrontBitmap()`, 728, 730
 - `pglIsIndirect()`, 728, 730
 - `pglMakeCurrent()`, 728, 730
 - `pglQueryCapability()`, 727, 729
 - `pglQueryConfigs()`, 728, 729
 - `pglQueryVersion()`, 727, 729
 - `pglReleaseFrontBitmap()`, 728, 730
 - `pglSelectColorIndexPalette()`, 729, 730
 - `pglSwapBuffers()`, 729, 730
 - `pglUseFont()`, 729, 730
 - `pglWaitGL()`, 728, 730
 - `pglWaitPM()`, 728, 730
- picking, 570–580
 - back buffer for, using the, 602
 - depth coordinates, 577
 - hierarchical models, 575–577
 - projection matrix, special, 571
 - sample program, 572
 - sample program with depth coordinates, 577
 - strategies, 581
 - sweep selection, 582
- Piegl, Les, 527
- pipeline
 - geometric processing, 667–668
 - imaging, 321–337
 - rendering, 10–14
 - vertex transformation, 106
- pixel
 - coverage, 247
- pixel data, 302, 312–321
 - see also* images
 - BGR and BGRA formats, 315
 - byte alignment, 328
 - byte swapping, 327
 - copying within the framebuffer, 13, 313, 321, 322, 669
 - depth buffer pixel data, 321, 332
 - drawing or reading a subrectangle of, 328
 - drawing process in detail, 338–339
 - endianness, 327
 - feedback mode, 585
 - formats for reading or drawing, 315
 - formats for storing in memory, 317, 325
 - mapping, 13, 333–334, 668
 - packed, 317–318
 - packing into processor memory, 13, 324–327, 668
 - performance tips, 341
 - pipeline operations, 12, 321–337, 668
 - pixel zoom, 334
 - querying pixel mapping information, 333
 - reading from the framebuffer, 313, 315
 - reading process in detail, 340–341
 - sample code which draws an image, 320
 - sample program which draws, copies, and zooms pixel data, 335
 - stencil buffer pixel data, 317, 332
 - storage modes, 325, 390–392
 - transfer modes, 13, 330, 417, 668



- unpacking from processor memory, 12, 324–327, 668
 - writing to the framebuffer, 313, 319
- point light source, *see* positional light source
- point parameters, 271–??
 - sample program, 273
- points, 38
 - antialiasing, 249–255, 608
 - drawing, 43
 - feedback mode, 585
 - point parameters, 271–??
 - round, 249–255, 608
 - size, 50
 - specifying, 43, 45
- polygon offset, 274–276
 - depth slope of a polygon, 275
 - enabling, 274
 - hidden-line removal, 614
 - shadowing use, 454
- polygonal approximations to surfaces, 93
- polygons, 39
 - boundary edges, 62–63
 - concave, drawing filled, 498, 610
 - convex, 39
 - culling the faces, 56
 - drawing, 43
 - drawing as points, lines, or filled, 56
 - feedback mode, 585
 - front and back faces, 56
 - holes in, 39
 - non-convex, 39, 62
 - nonplanar, 40
 - polygon mode, 12, 56, 668, 761
 - reversing the faces, 56
 - sample program with stippled polygons, 60
 - self-intersecting, 505
 - simple, 39
 - specifying, 43, 46
 - stippling, 58
 - tessellation, specifying for, 510
 - Voronoi, 618
- positional light source, 197
- primitives
 - geometric, 37–48
 - raster, 302
- priority of texture objects, 415
- Procedural Elements for Computer Graphics*, 516
- programs
 - aaindex.c, 252
 - aargb.c, 250
 - accanti.c, 488
 - accpersp.c, 484
 - alpha3D.c, 244
 - alpha.c, 241
 - bezcurve.c, 529
 - bezmesh.c, 538
 - bezsurf.c, 536
 - checker.c, 373
 - clip.c, 151
 - colormat.c, 218
 - colormatrix.c, 358
 - colortable.c, 346
 - combiner.c, 448
 - convolution.c, 351
 - cube.c, 110
 - cubemap.c, 438
 - dof.c, 490
 - double.c, 24
 - drawf.c, 304
 - feedback.c, 587
 - fog.c, 262
 - fogcoord.c, 269
 - fogindex.c, 266
 - font.c, 311
 - hello.c, 18
 - histogram.c, 360
 - image.c, 335
 - light.c, 190
 - lines.c, 54
 - list.c, 285
 - material.c, 215
 - minmax.c, 363
 - mipmap.c, 398
 - model.c, 125
 - movelight.c, 204
 - multisamp.c, 256





programs (*Continued*)

multitex.c, 441
 mvarray.c, 76
 pickdepth.c, 577
 picksquare.c, 572
 planet.c, 154
 pointp.c, 273
 polys.c, 60
 quadric.c, 521
 robot.c, 157
 select.c, 567
 shadowmap.c, 454–??
 smooth.c, 180
 stencil.c, 472
 stroke.c, 294
 surface.c, 544
 surfpoints.c, 555, 556
 tess.c, 502, 504, 512
 texbind.c, 411
 texgen.c, 430
 texsub.c, 383
 texture3d.c, 388
 texturesurf.c, 540
 torus.c, using a display list, 279
 trim.c, 559
 unproject.c, 161
 varray.c, 70

projecting images, 616

projection matrix, 113, 115

matrix stack, 148
 orthographic parallel projection matrix,
 756
 perspective projection matrix, 755
 shadows created with, 613

projection transformations, 112, 133–138

camera lens analogy, 106
 collapsing geometry to a single plane, 758
 jittering, 484, 487
 orthographic parallel, 113, 136–137, 759
 perspective, 133–136
 picking, 571
 texturing effects, 452
 two-dimensional, 138

proxies

color table, *see* color table proxies, 349

proxy textures, 380

cube maps, 437

Q

q texture coordinates, 451

avoiding negative values, 759

quadratic attenuation, 198

quadrics, 515–524

creating an object, 517
 destroying an object, 517
 drawing as points, lines, and filled
 polygons, 517
 error handling, 517
 normal vectors, generating, 518
 orientation, 518
 quadratic equation, 516
 sample program, 521
 steps to use, 516
 texture coordinates, generating, 518

quadrilateral

specifying, 43
 strip, specifying, 43, 46

R

raster position, 305

after drawing a bitmap, 307
 current, 305, 670
 current raster color, 308
 current, obtaining the, 306
 selection hit, 566
 transformation of, 305

rasterization, 170, 458

exact, two-dimensional, 759
 rendering pipeline stage, 13

readImage(), 347

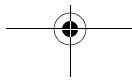
reading pixel data, *see* pixel data

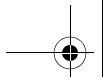
Real Projective Plane, The, 751

rectangles

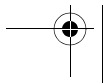
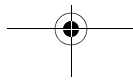
specifying, 40

reducing images, 334





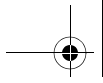
- reflecting objects, *see* scaling
 - reflection, *see* material properties
 - reflective objects, *see* environment mapping
 - refresh, screen, 21
 - removing hidden surfaces, *see* hidden-surface removal
 - repeatability, 764
 - rescaling normals, 65, 192
 - resident textures, 381, 414
 - management strategies, 415
 - querying residence status, 414
 - RGBA mode, 171
 - changing between color-index mode and, 176
 - choosing between color-index mode and, 175
 - coverage calculations for antialiasing, 247
 - data type conversion, 177
 - light source colors, 189
 - lighting calculations in, 221
 - material property values, 190
 - vertex arrays, specifying values with, 69
 - Robins, Nate, xxiv, 125, 131, 138, 160, 207, 216, 409, 420, 421, 428, 452
 - robot arm example, 156–159
 - Rogers, David, 516
 - Rossignac, Jarek, 611
 - rotating images, 616
 - rotation, 122
 - matrix, 754
- S**
- sample programs, *see* programs
 - scaling, 122
 - matrix, 754
 - scaling images, 616
 - Schneider, Bengt-Olaf, 611
 - Scientific American*, 619
 - scissor test, 468
 - and clearing, 463
 - rendering pipeline stage, 14, 669
 - secondary color, 450–451
 - specular, 210, 225
 - Segal, Mark, 452
 - selection, 562–583
 - back buffer for, using the, 602
 - hit records, 566
 - programming tips, 581
 - querying current rendering mode, 564
 - rendering pipeline stage, 668
 - sample program, 567
 - steps to perform, 563
 - sweep selection, 582
 - shading
 - flat, 179
 - performance tips, 761
 - sample program with smooth shading, 180
 - smooth, 179
 - specifying shading model, 179
 - shadows, 221, 494, 613
 - shininess, 214
 - see also* environment mapping
 - silhouette edges, 94
 - smoke, *see* fog
 - smooth shading, 179
 - solar system example, 153–156
 - source factor, *see* blending
 - specular
 - contribution to lighting equation, 224
 - light, 188
 - material properties, 189, 214
 - secondary specular color, 210, 225, 450
 - sphere map, 434
 - spheres, 516, 741
 - split-screen
 - multiple viewports, 139
 - spotlights, *see* light sources
 - state machine, 9–10
 - state variables, 48
 - attribute groups, 90–93
 - display list execution, effect of, 297
 - enable and disable states, 49
 - list of, 674–710
 - performance of storing and restoring, 760
 - querying, 49



stencil buffer, 460, 461
 clearing, 32, 463
 concave polygons, for drawing, 610
 decals, for, 609
 Dirichlet domains, for, 618
 Game of Life, for the, 619
 hidden-line removal, 615
 masking, 465
 pixel data, 317, 332
 stencil test, 470–475
 examples of using, 472
 interference regions found using clipping planes, 612
 querying stencil parameters, 471
 rendering pipeline stage, 14, 669
 sample program, 472
 stereo, 461, 464
 querying its presence, 461
 stippling
 display lists cache stipple patterns, 283
 enabling line stippling, 52
 enabling polygon stippling, 58
 fade effect, use for, 600
 line pattern reset, 53, 585, 589
 lines, 52
 polygons, 58
 sample program with line stipple, 54
 sample program with polygon stippling, 60
 stencil test, use of, 475
 translucency, use to simulate, 600
 stitching, 274
 stretching objects, *see* scaling
 stride
 vertex arrays, 71, 79
 subdivision, 93–101
 generalized, 100
 icosahedron example, 98
 recursive, 100
 subimages, 382–385, 386, 389
 superimposing images, 609
 surface normals, *see* normal vectors
 surfaces, *see* evaluators or NURBS
 swapping buffers, *see* double-buffering
 syntax, *see* command syntax

T

Terminator 2, 434
 tessellation, 40, 498–515
 backward compatibility with obsolete routines, 514
 begin and end callback routines, 502
 callback routines, 500–505
 combine callback routine, 502, 505
 contours, specifying, 510
 converting code to use the GLU 1.2 tessellator, 515
 creating an object, 500
 decomposition into geometric primitives, 502
 deleting objects, 513
 display list use, 282
 edge flag generation, 502
 error handling, 502
 evaluators used to perform, 760
 interior and exterior, determining, 506–509
 intersecting contours combined, 502, 505
 performance tips, 513
 polygons, specifying, 510
 properties, 506–510
 reuse of objects, 500, 514
 reversing winding direction, 510
 sample code, 502, 504, 512
 user-specified data, 505
 vertices, specifying, 503, 511
 winding rules, 506–509
 texels, 14, 367
 text, *see* characters
 texture coordinates, 375, 420–436
 assigning manually, 420
 avoiding negative q values, 759
 clamping, 423–427
 computing manually, 422
 cube maps, 438
 enabling automatic generation of, 433
 environment mapping, automatic generation for, 435
 evaluators, generated by, 540
 generating automatically, 429–436



- multitexturing, special situations with, 443
- NURBS, generating for, 552
- q coordinate, 451
- quadrics, generated for, 518
- reference planes, specifying, 429
- rendering pipeline stage, 12, 667
- repeating, 423–427
- sample program with texture coordinate generation, 430
- tessellation, specifying for, 504
- vertex arrays, specifying values with, 69
- wrapping modes, 423–427
- texture functions, 416–420
 - add, 420
 - blend, 420
 - blending color, 420
 - decal, 375, 419
 - fragment operations, 418
 - level of detail bias, 416
 - modulate, 419
 - pixel-transfer modes effect, 417
 - replace, 419
 - texture internal format, interaction with, 418
- texture images
 - alpha data, 417
 - borders, 395, 424
 - components, 376
 - data types, 377
 - distorting, 423
 - framebuffer as a source of, 379, 384, 386, 390
 - imaging pipeline operations, 324
 - intensity data, 417
 - internal format, 376
 - luminance data, 417
 - mipmaps, 396–406
 - multitexturing, 440
 - one-dimensional, 385–387
 - performance affected by internal format, 377
 - performance of texture subimages, 760
 - power of 2 size restriction, 378
 - proxy textures, 380
 - querying maximum size, 380
- residence status, 414
- resident textures, 381, 414
- resident textures, management strategies
 - of, 415
- sample program with mipmaps, 398
- sample program with subimages, 383
- specifying, 375–395
- subimages, 382–385, 386, 389
- three-dimensional, 387–392
- working set of textures, 381, 409, 414
- texture mapping
 - sample program using 3D textures, 388
- texture mapping, *see* texturing
- texture matrix, 451
 - rendering pipeline stage, 667
- texture objects, 375, 409–414
 - binding, 410
 - creating, 410
 - data which can be stored in, 410
 - deleting, 413
 - fragmentation of texture memory, 416
 - least-recently used (LRU) strategy, 416
 - mipmaps, 413
 - naming, 410
 - performance tips, 409, 760
 - priority, 415
 - rendering pipeline, 13, 669
 - sample program, 373
 - sample program with multiple texture objects, 411
 - sharing among rendering contexts, 717, 732
 - steps to perform, 409
 - using, 410
- texturing
 - see also* texture coordinates, texture functions, texture images, texture matrix, and texture objects
 - antialiasing characters, 616
 - antialiasing lines, 616
 - blending, 239
 - border colors, treatment of, 424
 - color-index mode limitations, 371, 378
 - combiner functions, 444–449
 - compressed textures, 392
 - creating contours, 430

texturing (*Continued*)

- cube maps, 436
- decals with alpha testing, 469
- differences among releases, 369
- enabling, 372, 375
- filtering, 406–408
- image transformations, 616
- mipmapping, 396–406, 408
- mosaic texture, 401
- multitexturing, 438–443
- perspective correction hint, 375
- popping visual artifact, 401
- rendering pipeline stage, 13, 669
- sample code with a depth texture, 454–??
- sample code with combiner functions, 448
- sample code with multitexturing, 441
- sample program, 373
- sample program with cube maps, 438
- sample program with evaluated, Bézier surface, 540
- sample program with mipmapping, 398
- sample program with texture coordinate generation, 430
- sample uses for, 616
- simulating shadows or spotlights, 451
- specular color separated, 210, 225, 450
- sphere map, 434
- steps to perform, 371

3D Computer Graphics: A User's Guide for Artists and Designers, xxiii

3D models, rendering, 20, 741

Tiller, Wayne, 527

tips, programming, 757

see also performance tips

error handling, 758

selection and picking, 581

transformations, 142

transformations

see also modeling transformations,
 projection transformations, viewing
 transformations, and viewport
 transformations

- combining multiple, 152–159
- display lists cache transformations, 283
- general-purpose commands, 114
- matrices, 753–756

- modeling, 117, 120–125
- ordering correctly, 117–120
- overview, 104
- performance tips, 760
- projection, 112, 133–138
- reversing the geometric processing
 pipeline, 160
- sample program, 110
- sample program combining modeling
 transformations, 154, 157
- sample program for modeling
 transformations, 125
- sample program showing reversal of
 transformation pipeline, 161
- troubleshooting, 142–144
- units, 136
- viewing, 117, 126–131
- viewport, 114, 138–140

translation, 121

- matrix, 754

translucent objects, 232, 600

- stencil test, creating with the, 475

transparent objects, 232

- creating with the alpha test, 469

transposing images, 621

triangle

- fan, specifying, 43
- specifying, 43, 45
- strip, specifying, 43, 45
- tessellated polygons decomposed into, 502

trimming

- curves and curved surfaces, 557–560
- sample program, 559

tutorials

- on-line, xxiv

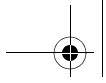
two-sided lighting, 209

U

up-vector, 111

Utility Library, OpenGL, *see* GLU

Utility Toolkit, OpenGL, *see* GLUT

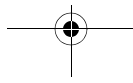
**V**

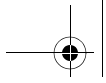
van Dam, Andries, xxii, 167, 171, 751
 van Widenfelt, Rolf, 452
 vendor-specific extensions, 597
 versions, 595–597
 GLU, 596
 vertex, 37
 see also vertex arrays
 evaluators, generating with, 528
 feedback mode, 585
 per-vertex operations pipeline stage, 12, 667
 specifying, 41
 tessellation, specifying for, 503, 511
 transformation pipeline, 106
 vertex arrays, 65–81
 dereference a list of array elements, 73, 75, 76
 dereference a sequence of array elements, 77, 78
 dereference a single element, 72
 differences among releases, 29
 disabling, 68
 display list use, 289
 enabling, 67
 interleaved arrays, 78
 interleaved arrays, specifying, 79
 multitexturing texture coordinates, 443
 performance tips, 761
 querying, 674
 querying range values, 76
 reuse of vertices, 74
 sample program, 70, 76
 specifying data, 68
 steps to use, 66
 stride between data, 71, 79
 vertex shader
 rendering pipeline stage, 667
 video
 fake, 620
 flipping an image with `glPixelZoom()`, 335
 textured images, 382
 viewing
 camera analogy, 106–107

viewing transformations, 110, 117, 126–131
 connection to modeling transformations, 111
 default position, 111
 different methods, 131
 pilot view, 132
 polar view, 132
 tripod analogy, 106
 up-vector, 111
 viewing volume, 134
 clipping, 138, 149
 jittering, 484, 487
 viewpoint
 lighting, for, 209
 viewport transformations, 109, 114, 138–140
 photograph analogy, 106
 rendering pipeline stage, 12, 668
 visual simulation
 fog, use of, 261
 Voronoi polygons, 618

W

w coordinates, 38, 109, 114
 avoiding negative values, 759
 lighting, use with, 197
 perspective division, 141, 668
 warping images, 616
 Watt, Alan, 366
 web sites, xxiii
 errata list, xxiv
 IBM OS/2 software and documentation, 727
 Microsoft Developer Network, 731
 WGL, 14, 731
 `wglCopyContext()`, 732, 734
 `wglCreateContext()`, 731, 732, 734
 `wglCreateLayerContext()`, 732, 734
 `wglDeleteContext()`, 734
 `wglDescribeLayerPlane()`, 731, 734
 `wglDestroyContext()`, 732
 `wglGetCurrentContext()`, 732, 734
 `wglGetCurrentDC()`, 732, 734
 `wglGetLayerPaletteEntries()`, 733, 735



**WGL (Continued)**

wglGetProcAddress(), 734
 wglMakeCurrent(), 732, 734
 wglRealizeLayerPalette(), 733, 735
 wglSetLayerPaletteEntries(), 735
 wglShareLists(), 732, 734
 wglSwapLayerBuffers(), 733, 735
 wglUseFontBitmaps(), 733, 735
 wglUseFontOutlines(), 733, 735
 wglGetProcAddress(), 599
 Williams, Lance, 396
 Win32
 ChoosePixelFormat(), 731, 733
 CreateDIBitmap(), 732, 734
 CreateDIBSection(), 732, 734
 DeleteObject(), 732, 734
 DescribePixelFormat(), 731, 734
 GetVersion(), 731, 733
 GetVersionEx(), 731, 733
 SetPixelFormat(), 731, 733
 SwapBuffers(), 733, 735
 winding, 57
 winding rules, 506–509
 computational solid geometry, used for, 507
 reversing winding direction, 510
 window coordinates, 109, 138
 feedback mode, 585
 polygon offset, 275
 raster position, 305

window management

 glViewport() called, when window resized, 139
 using GLUT, 17, 36
 Windows, *see* Microsoft
 working set of textures, 381, 409, 414
 fragmentation of texture memory, 416
 writemask, *see* masking (buffers)
 writing pixel data, *see* pixel data (drawing)
 www.opengl.org, xxiii

X

X Window System, 14, 716
 client-server rendering, 5
 minimum framebuffer configuration, 460
 X Visual, 176, 715

Z

z buffer, *see* depth buffer
 z coordinates, *see* depth coordinates
 zooming images, 334
 filtered, 621

