Index

2D geometry, 37-46, 77, 79-82
    (equation), 81
3D geometry, 4
3dfx company, 10
3ds max (Discreet company), 32, Plate 3, Plate 4

+ operator, 71
, operator, 72
() operator, 40
/ operator, 71, 72
. operator, 46, 72, 113, 129
* operator, 71, 72
- operator, 71
?: operator, 72, 119
() operator, 72
[] operator, 72
% operator, 72
<< operator, 72
>> operator, 72
< operator, 72
<= operator, 72
>= operator, 72
== operator, 72
+ operator, 72
| operator, 72
|| operator, 72
&& operator, 72
= operator, 72
+= operator, 72
-= operator, 72
*= operator, 72
/= operator, 72
1 unary operator, 72
++ unary operator, 72
- unary operator, 72
+ unary operator, 72
* unary operator, 72
& unary operator, 72

A

abs function, 76, 77, 277
absolute value function, 76
absorption
    fog, mathematics of, 236
    absorption
        in nonphotorealistic rendering, handling issues of, 246
    as shadow mapping issue, 257
abstract execution model (Cg), 3
acceleration
    graphics, pre-GPU history, 10
accessing
    decal textures, 69
    matrix rows, 115
    structure members, 46, 129
addition (+) operator, 71
affine modeling transform, 178
Airey, John, 232
Akenine-Möller, Tomas, 35
Alias|Wavefront company, 32, Plate 1
aliasing
    in nonphotorealistic rendering, handling issues of, 246
    as shadow mapping issue, 257
alignment issues in compositing, 259
alpha
    blending, compositing use, 258
    in RGBA, transparency encoding with, 46
testing, 16
alphanumeric characters
   not supported by current GPUs, 73
ambient lighting term, 104, 116, Plate 7
   (equation), 105
amplitude controls
   adding to pulsating objects program, 147
Angel, Edward, 98
animation, 167
   calculating displacement, 146-149
   (chapter), 143-168
   controls, 147
   rate, 4
   (term description), 143
anisotropic lighting model, 102, 142, Plate 29
annotations, 31, 297
API (application programming interface)
   See Direct3D programming interface; DirectX
   programming interface; OpenGL 3D programming interface
appearance(s)
   CgFX file format use, 30-34
   of objects, 2
application(s)
   Cg runtime relationship to, 29, 30
   CgFX relationship with, 33
   data, needed for graphics pipeline, 112
   user interface hooks, CgFX toolkit manipulation, 31
ARB_fragment_program OpenGL extension, 55, 272
ARB (OpenGL Architecture Review Board), 26
ARB_vertex_program OpenGL extension, 11, 47, 269
arbfp1 fragment profile, 55, 74, 272
arbvpl vertex profile, 47, 69, 75, 269
architecture
   See ARB (OpenGL Architecture Review Board);
   CineFX architecture (NVIDIA company);
   PixelFlow architecture
arithmetic operators, 72
array reference operator ([]), 72
arrays, 129-130
   Cg support for, 6
   packed
   matrix data type use, 42
   using, (advanced), 41
   passing, 130
assembly language
   code example, 6
   efficiency issues, 278
GPU, Cg relationship to, 5
graphics hardware, Cg as high-level language replacement, 1
assignment operators, 72
associativity of math operators, 72
ATI company, 10, 11
atmospheric effects
   See fog attenuation
   3D attenuation texture, 255
   C5E10_spotAttenLighting internal function, 139
   C5E6_attenuation internal function, 134
   C5E7_attenuate internal function, 135
   distance, 133
   (equation), 133
   not in Basic lighting model, 110
authors
   bibliographic references, See under specific author names
B
back-projection artifacts, 255
Banks, David, 142
Basic lighting model
   (equation), 103
   extending, 132-140
   fragment program, 120
   per-vertex, 103-120
   concepts, 103-110
   implementation, 110-120
bit masks
   not supported by current GPUs, 73
blending/blend
   alpha, compositing use, 258
   during graphics hardware pipeline raster operations stage, 16
blending/blend (continued)
  key-frame model use, 156, 157
  linear, expressing fog as, 237
  reflections, with decal textures, 175
  shape, Plate 15
Blinn, Jim, 99, 196, 232
Blinn lighting model, 102
boldface convention
  in source code examples, 37
bone set
  (term description), 163
bones
  vertex skinning representation of, 163
brick
  floor, bump mapping, 213-218
  wall
    bump mapping, 199-211
    simple bump mapping, 204
Brinkmann, Ron, 265
bulging objects
  periodic changes, 144-149
bump mapping, 232, Plate 13, Plate 24
  arbitrary geometries, 211
  brick floor, 213-218
  (chapter), 199-232
  nonuniform stretching issues, 228
  normal maps as a type of, 201
  object space, 212
  specular, 208
  (term description), 199
  texture space, 213
  textured polygonal meshes, 224-229
  a torus, 219-224

C
C language, 34
  as Cg development inspiration, 22
Cg differences
  arithmetic operations on vector types, 71
  call-by-result parameter passing, 81
  constructors, 46
  half data type support, 73
  matrix data types, 42
  no Cg Standard Library routines for I/O,
    string manipulation, or memory allocation, 75
  object-oriented programming not supported, 5
  program termination, 51
  Standard Library automatic inclusion, 40
  structure semantics, 39
  swizzling, 114
  vector data types, 40
  Cg language relationship to, 2
  Cg relationship to, 4
  C2E1v_green vertex program, 38
    rendering a triangle with, 58
  C2E2f_passthrough fragment program, 53
    as fragment program of Basic lighting model, 120
    particle systems use, 151
    pulsating object animation use, 144
    rendering a triangle with, 58
  C3E1v_anyColor vertex program, 62
  C3E2v_varying vertex program, 65
    rendering a textured 2D triangle with, 70
  C3E3f_texture fragment program, 67
    rendering a textured 2D triangle with, 70
  C3E4v_twist vertex program, 79
    execution results, 80
  C3E5v_twoTextures vertex program, 83
  C3E6f_twoTextures fragment program, 85
  C3E7f_twoTextures fragment program, 86
  C4E1v_transform vertex program, 97
  C5E1v_basicLight vertex program, 111
    application data needed, 112
    body, 113-120
    lighting results, 120
  C5E2v_fragmentLighting vertex program, 124
  C5E3f_basicLight fragment program, 125
  C5E4v_twoLights vertex program, 131
  C5E5_computeLighting internal function, 132
  C5E6_attenuation internal function, 134
  C5E7_attenuateLighting internal function, 135
  C5E8_spotlight internal function, 136
  C5E9_dualConeSpotlight internal function, 139
C5E10_spotAttenLighting internal function, 139
C6E1v_bulge vertex program, 145
C6E2v_particle vertex program, 152
C6E3v_keyFrame vertex program, 159
C6E4v_litKeyFrame vertex program, 161
C6E5v_skin4m vertex program, 166
C7E1v_reflection vertex program, 177
C7E2f_reflection fragment program, 180
C7E3v_refraction vertex program, 187
C7E4f_refraction fragment program, 188
C7E5v Dispersion vertex program, 193
C7E6f Dispersion fragment program, 194
C8E1v_bumpWall vertex program, 205
C8E2f_bumpSurf fragment program, 206
C8E3v_specWall vertex program, 208
C8E4f_specSurf fragment program, 209
C8E5v_bumpAny vertex program, 216
C8E6v_torus vertex program, 223
C9E1f_fog fragment program, 240
C9E2v_fog vertex program, 240
C9E3v_toonShading vertex program, 247
C9E4f_toonShading fragment program, 248
C9E5v_projTex vertex program, 254
C9E6f_projTex fragment program, 254
C9E7v_screenAlign vertex program, 260
C9E8f_tint fragment program, 260
C9E9f_a_over_b fragment program, 261
C9E10f_a_in_b fragment program, 262
C9E11f_a_out_b fragment program, 262
C9E12f_a_dissolve_b fragment program, 262
C++ language, 35
Cg differences
  arithmetic operations on vector types, 71
  call-by-result parameter passing, 81
  half data type support, 73
  matrix data types, 42
  no Cg Standard Library routines for I/O,
    string manipulation, or memory allocation, 75
  object-oriented programming not supported, 5
  program termination, 51
Standard Library automatic inclusion, 40
  structure semantics, 39
  swizzling, 114
  vector data types, 40
  Cg language relationship to, 2
  Cg relationship to, 4
Cabral, Brian, 232
CAD (computer-aided design) programs
  nonphotorealistic rendering in, 242
calculating
  in Basic lighting model, in object space, 110
  clip space position, 113
  diffuse lighting term, 107
  displacement, for animation, 146-149
  incident vector, in reflective environment mapping, 179
  per-fragment lighting, in object space, 123
  reflected ray, 173
  reflected vector, in Fresnel effect, 192
  reflection coefficient, in Fresnel effect, 192
  reflection vector, in reflective environment mapping, 179
  refracted vectors, in Fresnel effect, 192
  specular lighting term, 109
  spotlight effects, vectors for, 135
  call-by-reference parameter passing, 81
  call-by-result parameter passing, 81
  call-by-value-result parameter passing, 81
  capability errors, 49
  capacity errors, 50
  casting matrices, 177
Cebenoyan, Cem, 265
Central Processing Unit
  See CPU (Central Processing Unit)
Cg language
  advantages of, 5
  API independence, 23
  CgFX file format, 30-34
  compiler, 28-30
  data-flow model, 2
  environment, 26-34
  execution environment characteristics, 7
  getting started, (appendix), 281
  GPU assembly language relationship to, 5
  historical development, 21-26
Cg language (continued)

introduction
bibliographic references, 34
(chapter), 1-36
keywords, (appendix), 301
language characteristics, 5
Microsoft and NVIDIA collaboration in developing, 22
overview, 1-8
performance, rationale, 4
platform independence, 23
program, compiling and loading into GPU, 52
programming interface independence, 23
programs, downloading and configuring, 51-53
RenderMan vs.
meaning of uniform, 64
use of varying, 64
runtime, 28-30
(appendix), 283
applications relationship to, 30
loading and compiling Cg programs with, 47
relationship to general purpose languages, 4
simple programs, (chapter), 37-60
simplest programs, bibliographic references, 60
Standard Library, See Standard Library (Cg)
Standard Library functions, (appendix), 303-312
vertex and fragment programming, 21

cgc command-line compiler
debug and test use, 55
CgD3D library, 29
CgFX file format, 30-34, Plate 31
(appendix), 293-300
encapsulating multiple Cg implementations with, 48
CgGL library, 29
chroma key values, 263
chromatic dispersion, 188-194
(term description), 189
understanding, 190
chrome-like reflective objects, 175
CineFX architecture (NVIDIA company), 11, 73, 74, 147
clip space
2D view, 59
calculating, 113
as coordinate system, 95
coordinates, converting eye space coordinates into, projection transform use, 95
(term description), 58
transformations to, optimizations for, 97
clipping, 14, 255
color(s)
2D triangle, a simple program for rendering, 37-46
fog, (term description), 235
gradients, rendering, 66
particle, computing, 153
represented by continuous values, 73
COLOR semantic, 42, 66
fragment program meaning, 54
comma (,) operator, 72
comments
in Cg, notation for, 38
compiler/compiling/compilation, 47-53
Cg compiler characteristics, 28-30
Cg program, 52
dynamic, Cg support for, 29
errors, classes of, 48
static, Cg support for, 29
complexity
bump mapping advantages, 200
compositing, 258-263, 265
Cg as language for, 2
computer graphics
See graphics
computing/computations
math expressions in, 70-87
models, data-flow of Cg, 2
nonlinear, 82
particle
color, 153
positions, 152
size, 153
of reflection vectors, 172
vectorized, efficiency advantages of, 150
concepts
bibliographic references, See under specific concept names
conditional expression(s), 72, 119
cones

**C5E9_dualConeSpotlight** internal function, 139
effect of adding, 137
specification of, for spotlight effects, 136
configuring Cg programs, 51-53
consistency in coordinate systems, 214, 217
const type qualifier, 64
**constantColor** uniform parameter, 62
constants
specification of, with **const** type qualifier, 64
constructor
(term description), 46
content authoring
bump mapping advantages, 199
context errors, 49
continuous data types
representation, floating-point use, 73-74
control flow
See flow control
control(s)
animation, to pulsating objects program, 147
maps, encoding reflectivity as textures with, 181
conventional errors, 48
conventions
boldface, in source code examples, 37
naming
for, 38
of entry functions, 51
in out parameter names, 84
Cook, Robert, 35, 142, 264
as shade trees inventor, 23
coordinate systems, 89-97
clip space, 95
consistency in, 214, 217
eye space, converting into clip space, projection
transform use, 95
normalized device coordinates, 96-97
surface-local, for torus, 220
transformations and, for vertex processing, 90
window, 97
world space, 92
coordinates
homogeneous, 91
texture, 250
texture, sending, while sampling a texture, 69
textures, calculating, for projective texturing, 251
copy-in-copy-out parameter passing, 81
copy-out parameter passing
See copy-by-result parameter passing
cos function, 76
cosine function, 76
CPU (Central Processing Unit)
GPU relationship to, 3
programmable graphics pipeline relationship to, 17
cross function, 76, 217
cross product, 76
cube map(s)
normalization
constructing, 206
simple bump mapping use, 205
normalizing vectors with, 207
texture(s), 170-171
generating cube maps, 171
images, 170, Plate 8
lookup function, description and profile sup-
port, 77
culling
as back-projection artifacts solution, 255
of geometric primitives, in graphics hardware
pipeline, 14
(term description), 14
curly brackets (), 40
curved surfaces
representing with homogeneous positions, 91

D
data
application, needed for graphics pipeline, 112
data (continued)
geometric, transformation into coordinate systems, 89-97
data types
See Also CgFX file format
const, 64
continuous, representation, 73-74
double, representation, 73-74
first-class, arrays as, 130
fixed, characteristics and support for, 74
fixed-point, range-limited data type use, 74
float, representation, 73-74
floating-point
fixed-point vs., 74
representation, 73-74
half, representation, 73-74
matrix, 42
fixed<n>x<m>, 42
float<n>x<m>, 42
packed array use, 42
numeric, profile-dependent, 73-75
performance impact, 275
pointer
not supported by existing GPUs, 73
reasons for Cg lack of support, 6
profile support, 75
return, output structures use, 45
sampler, 68
structure member declaration of, 40
uniform type qualifier, 63
vector, 40
float2, 41
float3, 41
float4, 41
GPU hardware support for, 6
packed array use, (advanced), 41
data-flow model
of Cg, 2
for fragment processing, 20
for vertex processing, 18-19
Davis, Tom, 35, 141
Dawn character, Plate 15
DCC (digital content creation) applications, 32
ddx function, 76
ddy function, 76
depth
buffering, (term description), 96
comparison, in shadow mapping, 257
cueing, (term description), 263
of field, Plate 21
range transform, 97
testing, (term description), 16
determinant function, 76
determinant function, 76
developer.nvidia.com Web site, 35, 232, 265, 279
development
of Cg language, 21-26
cross-platform
Cg use with, 1, 23
CgFX provisions for providing, 48
debugging, cgc command-line compiler use, 55
development strategies, 55
GPU
forces driving, 8
development (continued)
GPU (continued)
history, 9-12
history
Direct3D, 27
OpenGL, 26-28
RenderMan Interface Standard, 23
strategies, for complex programs, 112
toolkit, for CgFX files, 30-34
Dietrich, Sim, 232
diffuse
lighting term, 116, 132, Plate 7
in Basic lighting model equation, 105
computing, 132
(equation), 107
shading, toon shading use, 243
vertex color (Direct3D), COLOR semantic as, 42
digital content creation (DCC) applications, 32
Direct3D programming interface
Basic lighting model simplifications relative to,
110
as Cg development inspiration, 22
development history, 27
as Microsoft 3D programming interface, 10
OpenGL relationship history, 27
OpenGL vs., clip space rules, 95
raster operations pipeline, 16
rendering, a triangle, 57
directions/directionality
decal textures independent of, 228
directional lights, 140
representing with homogeneous positions, 91
vectors, matrix representation, 217
DirectX 8 programming interface
fragment profiles
\texttt{ps\_1\_1}, 55, 270
\texttt{ps\_1\_2}, 55, 270
\texttt{ps\_1\_3}, 55, 270
vertex profiles, \texttt{vs\_1\_1}, 47, 267-268
DirectX 9 programming interface
fragment profiles
\texttt{ps\_2\_0}, 55, 271
\texttt{ps\_2\_x}, 55, 271
vertex profiles
\texttt{vs\_2\_0}, 47, 269
\texttt{vs\_2\_x}, 47, 269
DirectX Graphics
HLSL relationship to, 22
DirectX programming interfaces
as Cg language target, 2
as Microsoft multimedia programming interface, 10
Version 8, 11
Version 9, 11
Discreet company, 32, Plate 3, Plate 4
discrete values
not supported by current GPUs, 73
displacement calculations, 146-149, 148
dissolve compositing operation, 261
distance
attenuation, 133
fog
computing, 241
(term description), 235
distance function, 134
division (/) operator, 71
dot function, 76, 78, 117
dot (.) operator, 72
accessing structure members with, 129
swizzling use, 113
(term description), 46
dot product, 76, 107
double data type, 73-74
double vision effects, 83-87, Plate 6
See Also effects
downloading Cg programs, 51-53
Duff, Tom, 265
dynamic compilation, 29, 51

E
Ebert, David S., 264
dynamic compilation, 29, 51
edge detection
nonphotorealistic rendering issue, 246
effects
See Also double vision effects; fog; Fresnel, effect;
glowing effects; spotlight effects;
texture(s)/texturing, double vision effects
effects (continued)
  atmospheric, See fog
  combining bump mapping with other, 229-231
  representation of, 30-34

efficiency
  bump mapping advantages, 200
  of Cg Standard Library, 78
  in displacement calculations, for pulsating objects
  program, 148
  as homogeneous position advantage, 91
  importance for Cg programming, 7
  of sin function, in CineFX architecture, 147
  vectorized computation advantages, 150
else keyword, 119
emissive lighting term, 103, 115
entry functions
  declaration, 54
  multiple, 50
  (term description), 43
environment
  of Cg language, 26-34
  execution, characteristics of, 7
  mapping, 169-174, 172
    assumptions, 174
    bibliographic references, 196
    concept description, 171
    reflective, 175-182
    refractive, 182-188
    techniques, (chapter), 169
    (term description), 170
error(s)
  capability, 49
  capacity, 50
  compilation, classes of, 48
  context, 49
  conventional, 48
  debugging, cgc command-line compiler use, 55
  locating, in IDE, 56
  prevention of, 50
  profile-dependent, 49-50
Evans & Sutherland, 10
Everitt, Cass, 265
example(s)
  naming conventions for, 38

source code, boldface convention in, 37
execution
  environment, characteristics of, 7
  isolation, of Cg programs, advantages of, 7
exp2 function, 239
exponential function(s), 76, 159
expressions
  math, 70-87
  textures, and parameters, (chapter), 61-99
extensibility in OpenGL, 27
eye space, 93
  coordinates, converting into clip space, projection
    transform use, 95
  positions, converting to world space positions,
    view transform use, 93
  positions, converting to world space positions,
    view transform use, 93

F
Farin, Gerald, 88
FFD (free-form deformation), 168
file(s)
  See Also CgFX file format
  Cg, CgFX file format and toolkit, 30-34
  I/O, reasons for Cg lack of support, 6
first-class types
  arrays as, 130
fixed data type
  characteristics and support for, 74
  performance improvements in fog programs, 239
fixed<n>x<m> matrix data type, 42
fixed-function lighting model, 141
  See Also Basic lighting model for a simplified ver-
    sion
  concepts, 103-110
  implementation, 110-120
  per-vertex version, 103-120
  (term description), 102
fixed-point data types
  earlier profile use, 74
  floating-point vs., 74
  range-limited data type use, 74
float data type, 73-74
float<nxxm> matrix data type, 42
float2 vector data type, 41
float3 vector data type, 41
float4 vector data type, 41
floating-point data types, 73-74, 74
floor function, 76
flow control, 6, 130-132
fog, 233-241, 264
See Also effects
applying over multiple units of distance, 237
attributes, 235
color, (term description), 235
density, (term description), 235
distance
computing, 241
(term description), 235
equations, understanding, 238
expressing as a linear blend, 237
impact on light, 235
uniform, 234-235
creating, 239
(equation), 238
Foran, Jim, 196, 264
format of files
See CgFX file format
Fournier, Alain, 265
fp20 fragment profile, 55, 69, 74, 75, 271
fp30 fragment profile, 55, 69, 74, 75, 272
fragment(s)
as Cg data-flow model component, 3
per-fragment lighting, 121
profiles, 54-56
arbfp1, 55, 272
fp20, 55, 271
fp30, 55, 272
ps_1_1, 55, 270
ps_1_2, 55, 270
ps_1_3, 55, 270
ps_2_0, 55, 271
ps_2_x, 55, 271
texture-sampling, 69
programmable fragment processor
capabilities, 19-20
flow chart, 10, 20
operations, 52
programs
of Basic lighting model, 120
for chromatic dispersion, 194
downloading and configuring, 51-53
for per-fragment lighting, 124-126
for reflective environment mapping, 180
refraction environment mapping, 187
for simple bump mapping, 205
simple program for rendering, 53-56
for specular bump mapping, 209
(term description), 15
transformation of, 5
vertices and graphics pipeline relationship to, 8-21
frame buffer
CPU relationship to, 10
pixel relationship to, 15
frequency controls
adding to pulsating objects program, 147
Fresnel
See Also effects
effect, 188-194, 197, Plate 11, Plate 28
equations approximation, (equation), 189
lighting model, 102
frustum
view
defined by projection transform, 95
(term description), 14
function call operator (()), 72
function(s), 43-44
body, 45-46, 54
continuous, step function vs., 244
debugging, (appendix), 312
declaration of, 43, 126-127
definition, 62
definition, 62
definition, 62
entry, 43, 50, 54
exponential, quadratic interpolation use, 159
gleometric, (appendix), 308
internal, 44
interpolation, 160
lighting, 126, 128
mathematical, (appendix), 304
overloading, 77
function(s) (continued)
  partial derivatives, (appendix), 312
  spline, 159
Standard Library
  (appendix), 303-312
  description and profile support, 76
  step
    continuous function vs., 244
    encoding in a 1D texture, 245
    quadratic interpolation use, 159
    (term description), 243
  texture map, (appendix), 309
  time-based, 146-149
further reading
  2×2 matrix manipulation, 88
  animation, 167
  atmospheric effects, 264
  bump mapping, 232
  Cg language
    introduction, 34
    simplest programs, 60
  compositing, 265
  environment mapping, 196
  fog, 264
  lighting, 141
  matrix mathematics, for vertex transformations, 98
  nonphotorealistic rendering, 264
  performance, 279
  profile-independent function overloading, 78
  profiles, 279
  projective transformations, for vertex transformations, 98
  shadow mapping, 264
  vertex transformations, 98

2D
  simple program for rendering a triangle, 37-46
  texture lookup function, description and profile support, 77
  twisting, 79-82
  data, transformation into coordinate systems, 89-97
  functions, (appendix), 308
  parameters, needed for graphics pipeline, 112
  primitives
    assembly and rasterization, 14
    in graphics hardware pipeline, 13
  types of, 14
  self-shadowing, 230-231
  Glassner, Andrew, 99, 142
  gloss maps, 230
  glowing effects, 104
  See Also effects
  Gooch, Amy, 264
  Gooch, Bruce, 264
  Gooch shading, Plate 30
  goto, 132
  Gouraud shading, 121
GPU (Graphics Processing Unit)
  assembly language, Cg relationship to, 5
  Cg programs executed by, 3
  compiling and loading a Cg program into, 52
  CPU relationship to, 3
  development history, 9-12
  evolution and characteristics, 8
  execution environment, limitations of, 7
  first generation, capabilities and limitations, 10
  forces driving development of, 8
  fourth generation
    capabilities and limitations, 11
    required for per-fragment lighting, 123
  NVIDIA company, features and performance, 12
  per-fragment, implementation of, 123
  second generation, capabilities and limitations, 10
  third generation, capabilities and limitations, 11
gradient(s)
  inverse
    defining a surface-local coordinate system with, 221
gradient(s) (continued)
  inverse (continued)
    normal of a surface, (equation), 221
    in object-space partial derivatives equation, 226
rendering, 66
graphics
  acceleration, pre-GPU history, 10
  computer, Cg relationship to, 1
  hardware
    Cg as language for programming, 2
    pipeline, 13-17
  pipeline, 8-21
    programmable, 17-20
    semantics relationship to, 44
Graphics Processing Unit
  See GPU (Graphics Processing Unit)
Greene, Ned, 196
Gritz, Larry, 264
Heidrich, Wolfgang, 142
height fields, Plate 12
  bump map relationship, 203
  generating normal maps from, 202
  (term description), 202
hidden surface removal, 15
high-level language graphics programming, 1
history
  development
    Cg language, 21-26
    Direct3D, 27
    GPU, 9-12
    OpenGL, 26-28
  of RenderMan Interface Standard, 23
  key-frame use, 155
  OpenGL and Direct3D relationship, 27
HLSL (High-Level Shading Language), 2, 22
homogeneous
  coordinates, 91
  texture, 250
position
  converting between nonhomogeneous and, 91
  (term description), 91
hotspot specification, 136

I
  identifiers, 39
  if keyword, 119
  image rendering, 8-21
  in parameter qualifier, 81
  incident vector calculation, 179
  #include preprocessor statement, 40
  inlining of functions, 126
  inner product
    See vector dot product
  inout parameter qualifier, 81
  input semantics, 44-45, 45
  input/output routines, 75
  intensity
    light, modeling changes in, 133
    variation, 136
  interactivity performance requirements, 4
interface
See programming interface; RenderMan Interface
Standard (Pixar company); user interface

internal
functions, 44
function overloading in, 78
refraction, impact on refract function, 187
interpolation, 158-159
functions, 160
key frame, 155-162
with lighting, 161
linear, 158
quadratic, 158
by rasterization hardware, rendering gradients, 66
smooth color, 121

introduction
(chapter), 1-36
inverse
gradients
defining a surface-local coordinate system with, 221
normal of a surface, (equation), 221
in object-space partial derivatives equation, 226
kinematics solvers, vertex skinning use, 165
IRIS GL (SGI company), 22
Iritor Online (Wootsoft Entertainment), Plate 19
isnan function, 76

K
Java language, 2

Kernighan, Brian, 34
key frame(s)
animation, bibliographic references, 167
basic interpolation, 159-161
history of, 155
interpolation, 155-162
with lighting, 161
vertex skinning compared with, 164
keywords, 40

Kilgard, Mark, 232
kinematics, 167
inverse kinematics solvers, vertex skinning use, 165
vector kinematic equation, particle systems use, 149
Klassen, Victor, 264
Korobkin, Carl, 264

L
Lander, Jeff, 167
languages
See Also assembly language; C language; C++
language; Cg language; HLSL (High Level
Shading Language) language; PixelFlow
Shading Language (UNC university); Real-
Time Shading Language (Stanford University); shading languages
general purpose, Cg relationship to, 4
shading, noninteractive, 23-25
Lastra, Anselmo, 35
Lawson, Jim, 35
Legakis, Justin, 264
length function, 80
Lengyel, Eric, 142
lerp function, 76
libraries
Cg runtime, 29
CgD3D, 29
CgGL, 29
lighting, 141
See Also ambient lighting term; bump mapping;
chromatic dispersion; diffuse lighting term;
emissive lighting term; environment, mapping;
fog; Fresnel effect; reflectance/reflection;
refractive/refraction; specular lighting
term
(chapter), 101-142
directional, 140
functions, 128
creating, 126
intensity, modeling changes in, 133
key-frame interpolation with, 161
lighting (continued)
models, 101-103
  anisotropic, 102
  Basic, 103-120, 103-110, 110-120, 132-140
  Blinn, 102
  fixed-function, (term description), 102
  Fresnel, 102
  Phong, 102
  (term description), 102
parameters needed for graphics pipeline, 112
per-fragment, 121, 122
  advantages of, 123
  fragment program for, 124-126
  implementation of, 123
  vertex program for, 123-124
per-pixel, 122
per-vertex, Basic lighting model, 103-120
  results, 120
  with vertex skinning, 164
linear interpolation, 85, 158
  (equation), 158
lines
  as geometric primitive, 13
Linux operating system, 2, 10, 27
loading Cg programs, 52
log2 function, 76
logarithm function, 76
logical operators, 72
loop restrictions, 130

M
Macintosh operating systems (Apple Computer company), 2, 10, 27
magnitude controls, 148
main function, 43, 50
mapping
  See bump mapping; environment, mapping;
textures, mapping
Mark, Bill, 25, 35
Marselas, Herbert, 168
material parameters for graphics pipeline, 112
mathematics
  expressions, 70-87
  of fog, 236-238
  functions, (appendix), 304
  operations, on vectors, 71
  operators, associativity, precedence, and usage, 72
  of a torus, 219-222
matrix/matrices, 42
  casting, in reflective environment mapping, 177
  Cg support for, 6
  constructors used for, 46
  data types, packed array use, 42
  functions, description and profile support, 76
  manipulation, bibliographic references, 88
  modelview, 94
  palette blending, See vertex/vertices, skinning
  projection, 95
  effect of, 96
  representing
    modeling transform with, 92
    view transform with, 93
  rotation, 215
  rows, accessing with [] array operator, 115
max function, 76, 117, 231
Maya (Alias|Wavefront company), 32, Plate 1, Plate 2
member(s)
  names, semantic names distinct from, 43
  operator, See dot (.) operator
  structure, 40
    accessing with dot operator, 46
memory allocation, 6, 75
Mesa open source OpenGL implementation, 27
meshes, 224-229
micron, 11
Microsoft company, 11, 22, 26, 35
mirror direction in specular lighting, 107
mist
  See fog
model(s)
  data-flow
    of Cg, 2
    for fragment processing, 20
    for vertex processing, 18
  lighting, 101-103
  anisotropic, 102
model(s) (continued)
lighting (continued)
  Basic, extending, 132-140
  Basic, per-vertex, 103-120
  Basic, per-vertex concepts, 103-110
  Basic, per-vertex implementation, 110-120
  Blinn, 102
  fixed-function, (term description), 102
  Fresnel, 102
  Phong, 102
  (term description), 102
space, See object(s), space
modeling transform, 92, 93
See Also transformations
modelview matrix, 94
Molnar, Steven, 35
Moore's Law, 8
motion
capture, 164
of objects, 2
in time, 143
msdn.microsoft.com Web site, 35
msdn.microsoft.com/library Web site, 279
mul function, 76
multipass
effects, 31
rendering, 31
multiplication (*) operator, 71
Musgrave, Ken, 264

N
names/naming
  context differentiation of namespaces, 40
  conventions
    of entry functions, 51
    for examples, 38
    in out parameter names, 84
    member, semantic names distinct from, 43
NaN function, 76
negation (-) operator, 71
Neider, Jackie, 35, 141
Newell, Martin, 196
nis-lab.is.s.u-tokyo.ac.jp/~nis Web site, 264
Nishita, Tomoyuki, 264
nonlinear computations, 82
nonphotorealistic rendering (NPR), 241-248, 264,
  Plate 14
nonpremultiplied operator, 261
normal map, Plate 25
  brick wall, 200
  generating from height fields, 202
  (term description), 201
textures, storing bump maps as, 201
normalization
cube map
  constructing, 206
  simple bump mapping use, 205
  normal vector, (term description), 13
  normalized device coordinates, 96-97
  of vectors, 179
normalize function, 116
  normalization cube map vs., 206
NPR (nonphotorealistic rendering), 241-248, 264,
  Plate 14
NV_fragment_program OpenGL extension, 55, 272
NV_vertex_program OpenGL extension
  vertex profile for, 47
NVIDIA company
  Cg language
    browser interface, Plate 29
development, 25
  CineFX architecture
    fixed continuous data type support, 74
    as fourth-generation GPU, 11
    half-precision data type support for fragment
    programs, 73
sin function efficiency, 147
collaboration with Microsoft, in Cg and HLSL
  language development, 22
dancing ogre, Plate 17
developer Web site, see developer.nvidia.com
  Web site
fragment profiles
  fp20, 55, 271
  fp30, 55, 272
GeForce 2, as second-generation GPU, 10
NVIDIA company (continued)

GeForce 256, as second-generation GPU, 10
GeForce3, as third-generation GPU, 11
GeForce4 Ti, as third-generation GPU, 11
GeForce FX, as fourth-generation GPU, 11
GeForce FX GPU, number of transistors in, 9 as GPU term inventor, 9
GPUs, features and performance, (table), 12
NVMeshMender software, negative triangle handling with, 228
OpenGL driver, Linux support by, 27
time machine demo, Plate 16
TNT2, as first-generation GPU, 10
vertex profiles
  vp20, 47, 268
  vp30, 47, 269
Web site bibliographic references, 35
NVMeshMender software (NVIDIA company), 228

nvparse textural representation output, 279

0

object(s)
  appearance of, Cg as language for manipulating, 2
  motion of, Cg as language for manipulating, 2
  positioning within world space, modeling transform use, 92
  pulsating, 144-149
  shape of, Cg as language for manipulating, 2
  space, 91
  Basic lighting model calculations done in, 110
  bump mapping, 212
  orthogonality of texture space caveats, 227
  partial derivatives, for texture-space triangle, (equation), 226
  per-fragment lighting calculations done in, 123
  world space as absolute reference for, 92
  object-oriented programming, 5
  occlusion in projective texturing, 253
  offline rendering
  See RenderMan Interface Standard (Pixar company)

Olano, Marc, 25, 35
  opacity encoding, 46

OpenGL 3D programming interface, 35
  ARB_fragment_program extension
    arbfp1, 55, 272
  ARB_vertex_program extension
    arbvp1 vertex profile, 47, 269
  third-generation GPU support, 11
  Basic lighting model simplifications relative to, 110
  as Cg development inspiration, 22
  as Cg language target, 2
  development history, 26-28
  Direct3D relationship history, 27
  Direct3D vs., clip space rules, 95
  fixed-function lighting model, bibliographic references, 141
  fourth-generation GPU support, 11
  fragment profiles, 55
  NV_fragment_program extension
    fp30, 55, 272
  NV_register_combiners2 extension
    fp20, 55, 271
  NV_texture_shader extension
    fp20, 55, 271
  NV_vertex_program2 extension
    vp30 vertex profile, 269
  as open 3D programming standard, 10
  raster operations pipeline, 16
  rendering, a triangle, 57
  vertex profiles, 47

OpenGL Shader system (Silicon Graphics), 25
  operators/operations, 70-72, 72
  programmable fragment processor, 52
  programmable vertex processor, 51
  raster, 15, 16
  vectors, 71
  optimization(s)
    of displacement calculations, for pulsating objects program, 148
    for transformations to clip space, 97
    orthogonality
      of texture space and object space, caveats, 227
      oss.sgi.com/projects/ogl-sample/registry Web site, 279
out parameters
call-by-result parameter passing with, 81
output structures vs., 84
returning multiple results with, 127
outlining of silhouette, 245
output semantics
and input semantics, flow for C2Elv_green, 45
input vs., 44-45
structures, 38, 53
declaration of, 39
fragment vs. vertex programs, 53
(term description), 39, 45
over compositing operator, 261
overloading
functions, 77
in length and sincos routines, 80
profile-independent function, bibliographic references, 78
overview of Cg language, 1-8

P
packed arrays, 41, 42
parallel processing, 4
parameter(s), 61-66
array passing as, 130
chromatic dispersion, 191
Fresnel effect, 191
geometric, 112
lighting, 112
material, 112
out, 84
particle systems, 151
passing
call-by-result, 81
call-by-value-result, 81
call-by-value, 81
copy-in-copy-out, 81
reflective environment mapping, 176
textures, and expressions, (chapter), 61-99
uniform, 61-64, 274
varying, 65-66
parametric equations
for a torus, (equations), 220
partial derivatives
functions, (appendix), 312
object-space, for texture-space triangle, (equation), 226
of parametric torus, (equation), 221
particle(s)
color, computing, 153
positions, computing, 152
size, computing, 153
systems, 149-155, 149
bibliographic references, 167
parameters, 151
with point sprites, 155
(term description), 149
trajectory, (equation), 149
passing
arrays, 130
parameters
call-by-result, 81
call-by-value-result, 81
call-by-value, 81
copy-in-copy-out, 81
Peercy, Mark, 232
per-fragment lighting, 122
advantages of, 123
fragment program for, 124-126
implementation of, 123
vertex program for, 123-124
per-pixel lighting, 122
performance, 273-278
assembly code efficiency issues, 278
bibliography, 279
Cg, 4
CineFX architecture, half data type advantage, 73
data types impact, 275
encoding functions as textures advantages, 276
fog programs, fixed data type advantages, 239
NVIDIA company GPUs, 12
profiles and, (chapter), 267-279
shading issues, 277
Standard Library advantages, 273
performance (continued)
  swizzling advantages, 277
  uniform parameter issues, 274
  variable negation advantages, 277
  vectorization advantages, 275
  vertex programs vs. fragment programs, 274
  in vertex skinning example, impact on matrices, 165
Perlin, Ken, Plate 27
period operator
  See dot (.) operator
periodicity in shape changing, 144-149
Phong
  lighting model, 102
  See Also Basic lighting model for a simplified version
  shading, 122
pipeline
  graphics, 8-21
  application data needed for, 112
  semantics relationship to, 44
  hardware graphics, 13-17
  programmable graphics, 17-20, 17
Pixar company, 22, 23
pixel(s)
  Cg fragments relationship to, 3
  (term description), 15
  PixelFlow architecture, 25, 35
  PixelFlow Shading Language (UNC university), 22
platforms
  multiple
    Cg use with, 1, 23
    CgFX provisions for providing, 48
    development strategies, 55
point(s)
  converting to point sprites, 154
  as geometric primitive, 13
  size
    controlling with PSIZE semantic, 153
    function for, 154
  sprites
    bibliographic references, 168
    manipulating point appearance with, 154
  pointer types
  not supported by existing GPUs, 73
  reasons for Cg lack of support, 6
  Policarpo, Fabio, Plate 23
polutants
  See fog
polygons/polygonal
  as geometric primitive, 13
  meshes
    generalizing single triangle approaches to, 228
    textured, bump mapping of, 224-229
portability, 5
Porter, Thomas, 265
pose(s)
  constructing from matrices, in vertex skinning animation, 163
  default, in vertex skinning animation, 163
position(s)
  homogeneous, (term description), 91
  particle, computing, 152
  vectors, matrix representation, 217
POSITION semantic, 42
  errors of, 49
  initializing varying parameters with, 65
  with input structures, meaning of, 44
  input vs. output meaning of, 44
Poulin, Pierre, 265
pow function, 76, 118
precedence (math operators), 72
precision in Cg Standard Library, 78
primitives (geometric)
  assembly and rasterization, 14
  in graphics hardware pipeline, 13
  types of, 14
PRMan (PhotoRealistic RenderMan) rendering system (Pixar company), 23
procedural generation
  of a wood texture, Plate 20
  of a terrain, Plate 27
  of a torus, 222
processor
  programmable fragment capabilities, 19-20
  flow chart, 10
  operations, 52
programmable processor (continued)
programmable vertex capabilities, 18-19
flow chart, 18
operations, 51
profile(s), 78, 267-273
bibliography, 279
data types supported by, 75
-dependent errors, 49-50
-dependent numeric data types, 73-75
fragment, 54-56, 69
hardware, 7
independence, 48, 78
multiple, 48
performance and, 267-279
vertex, 47, 47-48, 48
program(s)
fragment, downloading and configuring, 51-53
termination, Cg vs. C and C++, 51
vertex, downloading and configuring, 51-53
programmable
fragment processor
capabilities, 19-20
flow chart, 10
operations, 52
graphics hardware, Cg relationship to, 5
graphics pipeline, 17-20, 17
vertex processor
capabilities, 18-19
flow chart, 18
operations, 51
programming
in Cg, simple programs, (chapter), 37-60
interface
See Also Direct3D programming interface
See Also DirectX programming interface
See Also OpenGL 3D programming interface
vertex profile selection relationship to, 47
projective/projection
3D texture lookup function, description and profile support, 77
back-projection artifacts, projective texturing issue, 255
matrix, 95
effect of, 96
texturing, 248-255, 249
calculating projective coordinates for, 251
implementation, 251
transform, 95-96
transformations, homogeneous position advantages, 91
Proudfoot, Kekoa, 25, 35
ps_1_1 fragment profile, 55, 69, 74, 75, 270
ps_1_2 fragment profile, 55, 75, 270
ps_1_3 fragment profile, 55, 75, 270
ps_2_0 fragment profile, 55, 75, 271
ps_2_3 fragment profile, 55, 69, 75, 271
PSIZE semantic, 153
pulsating
alien, 145
objects, 144-149
Q
quadratic interpolation, 158
querying
See Also accessing
textures, 68
R
Radeon 7500 GPU (ATI company), 10
Radeon 8500 GPU (ATI company), 11
Radeon 9700 GPU (ATI company), 11, 123
radians function, 76
Rage (ATI company), 10
raster operations
graphics hardware pipeline stage, 15
OpenGL and Direct3D operations pipeline, 16
rasterization, 14
rational polynomials, 91
real-time
3D applications, performance requirements, 4
shading language, Cg characterized as, 4
Real-Time Shading Language (Stanford University), 22
Reality Lab (RenderMorphics company), 22
reciprocal function, 77
redirection
  fog, mathematics of, 236
Reeves, William, 264
**reflect** function, 6, 76, 173
reflectance/reflect(s)
  ambient lighting term relationship to, 104
  blending, with decal textures, 175
  coefficient, calculating, in Fresnel effect, 192
  control map management of, 181
  diffuse lighting term relationship to, 105
  environment mapping (chapter), 169-197, Plate 9
  reflected, ray, 171, 173
  vectors, 76, 192
    calculating, 179
    computing, 172
    (equation), 173
  shading, bibliographic references, 196
**refract** function, 186
refractive/refraction
  into environment map, 185
  environment mapping, 182-188, Plate 10
  indices of, 184
    ratio of, 185
  internal, impact on **refract** function, 187
  one vs. multiple refractions, 185
  refracted vectors, calculating, in Fresnel effect, 192
    (term description), 183
Rege, Ashu, 265
render(ing)
  ambient lighting term, 105
  animated, time issues, 143
  bead shapes, Plate 2
  Cg as language for, 2
  diffuse lighting term, 106
  efficiency, Cg programming impact on, 7
  emissive lighting term, 104
  gradients, 66
  images, GPU handling of, 8-21
  multipass, specification of, as CgFX capability, 31
  nonphotorealistic, 241-248
  a simple program for fragments, 53-56
  vertices, 37-46
  simple vertex and fragment examples, 56-60
  specular lighting term, 108
  state for an effect, CgFX file description of, 30
  textured 2D triangle, 70
  triangle
    with Direct3D, 57
    with OpenGL, 57
RenderMan (Pixar company), 22, 23, 24, 35, 64, 196
RenderMorphics company, 22
representation
  of complex surfaces, bump mapping use, 199
  of continuous data types, 73-74
  direction vectors, 217
  effects and appearances, CgFX file format use, 30-34
  of modeling transform, matrix use, 92
  of position vectors, 91
  position vectors, 217
  of time, 143
  of view transform, matrix use, 93
retrieving
  See accessing
**return** statement, 46
return types, 45
Reyes algorithm
  hardware implementation issues, 25
RGBA (Red Green Blue Alpha) color values
  in **C2E1v_green** vertex program, 38
    (term description), 46
Ritchie, Dennis, 34
rotating/rotation
  matrices, 215
    (equation), 215
  as modeling transform, 92
  vertices, 81
**round** function, 77
**rsqrt** function, 77
RTSL (Real-Time Shading Language) Stanford University, 25
runtime
  Cg, 28-30
    (appendix), 283-291
runtime (continued)
  Cg (continued)
    applications relationship to, 30
    loading and compiling Cg programs with, 47
    generation of Cg programs, 51

S
S3 company, 10
Salesin, David, 264
sampler1D type qualifier, 68
sampler2D type qualifier, 68
sampler3D type qualifier, 68
samplerCUBE type qualifier, 68
samplerRECT type qualifier, 68, 271, 272
samplers/sampling
  objects, 67-68
  textures, 67-70, 69
  types, 68
saturate function, 210, 277
Savage3D (S3 company), 10
scalar/vector relationships, 71
scaling, 92
scattering
  diffuse light, 106
  fog, mathematics of, 236
  in specular lighting, 107
Schlechtweg, Stefan, 264
scissor testing, 16
Segal, Mark, 264
Seidel, Hans-Peter, 142
semantics, 42-43
  in Cg structures, 39
COLOR semantic, 42, 54, 66
errors, 49
initializing varying parameters with, 65
input, 44, 44-45, 45
not used with internal functions, 44
output, input vs., 44-45
POSITION semantic, 42, 44, 49, 65
PSIZE semantic, 153
of return values, handling of, 46
specification of variable values with, 63
  (term description), 42
TEXCOORD0 semantic, 66
SGI (Silicon Graphics Inc.), 10, 22, 25
shade trees, 24, 35
  concept and history of, 23-24
shaders/shading
  algorithms, Cg translation of, 21
  annotations to, 31
  diffuse, toon shading use, 243
fragment profiles
  ps_1_1, 55, 270
  ps_1_2, 55, 270
  ps_1_3, 55, 270
  ps_2_0, 55, 271
  ps_2_x, 55, 271
Gooch, Plate 30
Gouraud, 121
languages, 22
  Cg relationship to, 2
  hardware-amenable, 25
  noninteractive, 23-25
  RenderMan, 24
in Maya 4.5, Plate 1
multiple instancing of, 31
performance issues, 277
Phong, 122
skin, Plate 15
toon, 242-246, 242
  implementing, 243
vertex profiles
  vs_1_1, 47, 267-268
  vs_2_0, 47, 269
  vs_2_x, 47, 269
shadow mapping, 264
shadows/shadowing
  adding to a scene, 255-258
  mapping, 255-258
  stencil-based, 16
shininess exponents, 108
Shreiner, Dave, 35, 141
side effects, 7
silhouette outlining, 245
Silicon Graphics
  See SGI (Silicon Graphics Inc.)
simulation
  Cg as language for, 2
  computation required for, 9
sin function, 147
sincos function, 80
size
  point, controlling with PSIZE semantic, 153
  reasons for Cg programming restrictions, 7
sizeof unary operator, 72
skin representation, 164
skinning (vertex), 162-166, Plate 15
  key frames compared with, 164
smearing, 71, 114
smoke
  See fog
smooth color interpolation, 121
smoothstep function, 137, 138
Snell’s Law, 183, 184
Softimage company, 32, Plate 5
space
  See clip space; eye space; object(s), space;
texture(s)/texturing, space; world space
specification
  of constants, with const type qualifier, 64
  of variables
    with uniform type qualifier, 61-64
    without uniform type qualifier, 63
specular
  bump mapping, 208
  highlights, in nonphotorealistic rendering, 245
  lighting term, 117, 132, Plate 7
    in Basic lighting model equation, 107
    computing for per-fragment lighting, 132
    (equation), 109
spline functions
  quadratic interpolation use, 159
spotlight effects, 134-139
See Also effects
  C5E10_spotAttenLighting internal function, 139
  C5E8_spotlight internal function, 136
  C5E9_dualConeSpotlight internal function, 139
  not in Basic lighting model, 110
  specifying a cut-off angle, 135
  vectors for calculating, 135
sprites
  bibliographic references, 168
  manipulating point appearance with, 154
Standard Library (Cg), 75-79
  automatic inclusion of, 40
  functions, 76
    (appendix), 303-312
  graphics operation support in, 6
  performance advantages, 273
standards (OpenGL), 10
Stanford University, 22, 25
state
  pixel, components of, 15
  render, for an effect, CgFX file description of, 30
static compilation
  Cg support for, 29
  (term description), 29
  stencil testing, 16
  stencil-based shadowing, 16
step functions
  continuous function vs., 244
  encoding in a 1D texture, 245
  quadratic interpolation use, 159
  (term description), 243
strategies
  cross-platform development, 55
  development and debugging, for complex pro-
grams, 112
stretching
  nonuniform, bump mapping issues, 228
string manipulation, 75
Strothotte, Thomas, 264
Stroustrup, Bjarne, 35
struct
  keyword, 39, 128
  structure(s), 128-129
structure(s), 128-129
See Also effects
  C5E10_spotAttenLighting internal function, 139
  C5E8_spotlight internal function, 136
  C5E9_dualConeSpotlight internal function, 139
  not in Basic lighting model, 110
  specifying a cut-off angle, 135
  vectors for calculating, 135
 bibliographic references, 168
  manipulating point appearance with, 154
  automatic inclusion of, 40
  functions, 76
    (appendix), 303-312
  graphics operation support in, 6
  performance advantages, 273
  standards (OpenGL), 10
  Stanford University, 22, 25
state
  pixel, components of, 15
  render, for an effect, CgFX file description of, 30
  stencil testing, 16
  stencil-based shadowing, 16
step functions
  continuous function vs., 244
  encoding in a 1D texture, 245
  quadratic interpolation use, 159
  (term description), 243
strategies
  cross-platform development, 55
  development and debugging, for complex pro-
grams, 112
stretching
  nonuniform, bump mapping issues, 228
string manipulation, 75
Strothotte, Thomas, 264
Stroustrup, Bjarne, 35
struct
  keyword, 39, 128
  structure(s), 128-129
structure(s), 128-129
See Also effects
  C5E10_spotAttenLighting internal function, 139
  C5E8_spotlight internal function, 136
  C5E9_dualConeSpotlight internal function, 139
  not in Basic lighting model, 110
structure(s) (continued)
output (continued)
  out parameters vs., 84
    (term description), 39, 45
subtraction (-) operator, 71
surface(s)
  complex, 199
  curved, 91
  hidden, removal, 15
  -local coordinate systems, torus, 220
  tangent, binormal, and normal on, (equation), 222
switch
  as unsupported reserved keyword, 132
swizzling, 113-115
  compositing use, 261
  smearing in, 114
  (term description), 19
T
tangent-space bump mapping
  See texture(s)/texturing, space, bump mapping
termination of program, 51
tessellation
  effects on lighting quality, 121
  vertex programs, importance of, 82
tex2D function, 69, 77
tex3Dproj function, 77
TEXCOORD0 semantic, 66
texcube function, 69, 77, 181
texels
  See Also texture(s)/texturing
  height field relationship to, 202
  storing data in, 201
texgen, 252
textures/texturing
  1D, encoding a step function in, 245
  accesses, vertex, errors, 49
  bump mapping, textured polygonal meshes, 224-229
  bump mapping use for representation of complex surfaces, 199
  coordinates
calculating, for projective texturing, 251
  sending, while sampling a texture, 69
cube map, 170-171
  images, 170
  lookup function, description and profile support, 77
  lookup function, for chromatic dispersion, 195
deal
  accessing, 69
  blending reflections with, 175
direction independence, 228
  lookups, 181
  (term description), 68
  uniform fog creation use, 239
double vision effects, 83-87
  results, 84
dynamic diffusion, Plate 22
encoding
  fog factor in, 241
  reflectivity in, 181
encoding functions as, performance advantages, 276
expressions, and parameters, (chapter), 61-99
filtering specification, 31
homogeneous coordinates, 250
mapping, Plate 23
  functions, (appendix), 309
  matrix, (term description), 252
  normal map, storing bump maps as, 201
  programmable fragment processor support, 19
  projective, 248-255, 249
  calculating projective coordinates for, 251
  implementation, 251
querying, 68
sampling, 67-70
  lookup use, 69
  sending coordinates while, 69
space
  bump mapping, 213
  negative-area triangles caveats, 227
  orthogonality of object space caveats, 227
  per-vertex, 218
  zero-area triangles caveats, 227
  type qualifiers for, 68

Index
time
- based functions, creating, 146-149
changing rate of transition over, quadratic interpolation use for, 158
global, 152
movement in, 143
representation of, 143
TNT2 (NVIDIA company), 10
toon shading, 242-246
Torrance, Kenneth, 142
torus
mathematics of, 219-222
parametric, partial derivatives of, (equation), 221
procedural generation of, from a 2D grid, 222
rendered, 224
tessellated, bump mapping of a, 219-224
tradeoffs
See Also performance
profile selection, 48
vertex programs vs. fragment programs, 182, 274
transform
See transformation(s)
transformation(s), 98
(chapter), 89-99
to clip space, 97
comparing, 250
depth range, 97
fragment, 5
of geometric data, 89-97
modeling, 92, 93
projection, 95-96
projective, 91
texturing, 252
rotation, 81
of vectors into world space, 177
vertex, 5, 13
view, 93, 94
viewport, 97
translation, 92
transparency, 46
trees
See shade trees
triangles
negative-area, caveats, 227
rendering
with Direct3D, 57
with OpenGL, 57
a simple program for rendering, 37-46
single, in 3D wireframe models, bump mapping, 224
single-textured, plane equation coefficients, (equation), 226
zero-area, caveats, 227
twisting 2D geometry, 79-82
(type) unary operator, 72
types
See data types
Tzvetkov, Svetoslav, 25, 35

U
UNC (University of North Carolina, Chapel Hill), 25
uniform
fog, 234-235
creating, 239
(equation), 238
parameters, 61-64
performance issues, 274
uniform type qualifier, 63
University of North Carolina, Chapel Hill, 22
UNIX operating system, 10, 27
updating vectors, 115
Upstill, Steve, 35
user interface hooks for applications, 31

V
van Widenfelt, Rolf, 264
variables specification, 61-64, 63
varying
magnitude of bulging, 148
parameters, 65-66
vector(s), 40
Cg support for, 6
constructors used for, 46
cross product relationships, (equation), 215
data types, packed array use, 41
vector(s) (continued)
direction, matrix representation, 217
dot product, in diffuse lighting calculation, 107
efficiency advantages of vectorizing computations, 150
half-angle, (term description), 118
incident, calculating, 179
kinematic equation, particle systems use, 149
math operations, 71
normal, (term description), 13
normalization, 179
performance advantages, 275
position, matrix representation, 217
reflection
  calculating, 179, 192
  computing, 172
  (equation), 173
refracted, calculating, 192
represented by continuous values, 73
as return value from texCUBE, texture sampling use, 69
scalar relationships, in vector operations, 71
selective updating, with write masking, 115
transforming into world space, in reflective environment mapping, 177
view, (term description), 118
vertex/vertices
  assembling into geometric primitives, in graphics hardware pipeline, 14
  attributes of, in graphics hardware pipeline, 13
  as Cg data-flow model component, 3
  fragments and graphics pipeline relationship to, 8-21
  positions, transformation, as requirement for all vertex programs, 90
profiles, 47-48
arbvp1, 47, 269
selection criteria, 47, 48
vp20, 47, 268
vp30, 47, 269
vs_1_1, 47, 267-268
vs_2_0, 47, 269
vs_2_x, 47, 269
programmable vertex processor capabilities, 18-19
flow chart, 18
operations, 51
programs
  of Basic lighting model, 103-120
  for bump mapping a torus, 222-224
calculating projective texturing coordinates in, 251
for chromatic dispersion, 192
downloading and configuring, 51-53
fragment program performance tradeoffs, 274
for Fresnel effect, 192
for per-fragment lighting, 123-124
for pulsating objects, 144
for reflective environment mapping, 176
for refractive environment mapping, 186-187
rendering, 37-46
for rendering a brick floor, 215
for simple bump mapping, 204
for specular bump mapping, 208
vertex skinning in, 165
representation, in object space, 91
rotating, 81
shared, blending, in polygonal mesh bump mapping, 228
skinning, 162-166
  bibliographic references, 167
  key frames compared with, 164
  theory, 163
texture accesses, 49
transformation, 13
  (term description), 13
  transformation of, 5
VGA (Video Graphics Array) controller, 9
view
  frustum
    defined by projection transform, 95
    (term description), 14
  ray, (term description), 171
reflected, view ray relationship to, 171
space, See eye space
transform, 93
effect of, 94
vector, (term description), 118
viewport
    projective texturing relationship, 249
    transform, 97
Vince, John, 168
Voodo03 (3dfx company), 10
Voorhies, Doug, 196
vp_2_0 vertex profile, 74
vp20 vertex profile, 47, 75, 268
vp30 vertex profile, 47, 75, 269
vs_1_1 vertex profile, 47, 75, 267, 267-268
vs_2_0 vertex profile, 47, 75, 269
vs_2_x vertex profile, 47, 269

W
Waliszewski, Arkadiusz, Plate 20, Plate 21
Wang, Ulan, 35
water
    interaction, Plate 26
    vapor, See fog
Williams, Lance, 264
window coordinates, 97
Windows operating system, 27
Windows operating systems, 10
Windows platform, 2
wireframe models, 224
Wloka, Matthias, 197
Woo, Andres, 265
Woo, Mason, 35, 141
Woodland, Ryan, 168
Wootsoft Entertainment, Plate 19
world space, 92
    positioning objects within, modeling transform use, 92
    positions, converting eye space positions to, view transform use, 93
    transforming vectors into, in reflective environment mapping, 177
Worley, Steven, 264
write masking, 115
www.cgshaders.org Web site, 60
www.darwin3d.com Web site, 167
www.opengl.org Web site, 35, 168

X
Xbox (Microsoft company), 2, 11, 27
XSI 3.0 (Softimage company), 32, Plate 5

Y
Yeti Studios Limited, Plate 18
YOUR GUIDE TO IT REFERENCE

Articles

Keep your edge with thousands of free articles, in-depth features, interviews, and IT reference recommendations – all written by experts you know and trust.

Online Books

Answers in an instant from *InformIT Online Book’s* 600+ fully searchable online books. For a limited time, you can get your first 14 days **free**.

Catalog

Review online sample chapters, author biographies and customer rankings and choose exactly the right book from a selection of over 5,000 titles.
Wouldn’t it be great
if the world’s leading technical
publishers joined forces to deliver
their best tech books in a common
digital reference platform?

They have. Introducing
**InformIT Online Books**
powered by Safari.

- **Specific answers to specific questions.**
InformIT Online Books’ powerful search engine gives you relevance-ranked results in a matter of seconds.

- **Immediate results.**
With InformIT Online Books, you can select the book you want and view the chapter or section you need immediately.

- **Cut, paste and annotate.**
Paste code to save time and eliminate typographical errors. Make notes on the material you find useful and choose whether or not to share them with your work group.

- **Customized for your enterprise.**
Customize a library for you, your department or your entire organization. You only pay for what you need.

**Get your first 14 days FREE!**
For a limited time, InformIT Online Books is offering its members a 10 book subscription risk-free for 14 days. Visit [http://www.informit.com/onlinebooks](http://www.informit.com/onlinebooks) for details.
If you are interested in writing a book or reviewing manuscripts prior to publication, please write to us at:
Editorial Department
Addison-Wesley Professional
75 Arlington Street, Suite 300
Boston, MA 02116 USA
Email: AWPro@aw.com

You may be eligible to receive:
• Advance notice of forthcoming editions of the book
• Related book recommendations
• Chapter excerpts and supplements of forthcoming titles
• Information about special contests and promotions throughout the year
• Notices and reminders about author appearances, tradeshows, and online chats with special guests

Contact us

If you are interested in writing a book or reviewing manuscripts prior to publication, please write to us at:

Editorial Department
Addison-Wesley Professional
75 Arlington Street, Suite 300
Boston, MA 02116 USA
Email: AWPro@aw.com

Visit us on the Web: http://www.awprofessional.com
Addison-Wesley Warranty on the CD-ROM
Addison-Wesley warrants the enclosed disc to be free of defects in materials and faulty workmanship under normal use for a period of ninety days after purchase. If a defect is discovered in the disc during this warranty period, a replacement disc can be obtained at no charge by sending the defective disc, postage prepaid, with proof of purchase to:

Editorial Department
Addison-Wesley Professional
Pearson Technology Group
75 Arlington Street, Suite 300
Boston, MA 02116
Email: AWPro@awl.com

Addison-Wesley makes no warranty or representation, either expressed or implied, with respect to this software, its quality, performance, merchantability, or fitness for a particular purpose. In no event will Addison-Wesley, its distributors, or dealers be liable for direct, indirect, special, incidental, or consequential damages arising out of the use or inability to use the software. The exclusion of implied warranties is not permitted in some states. Therefore, the above exclusion may not apply to you. This warranty provides you with specific legal rights. There may be other rights that you may have that vary from state to state. The contents of this CD-ROM are intended for personal use only, unless otherwise addressed under a separate license.

NVIDIA Statement on the Software
The source code provided is freely distributable, so long as the NVIDIA header remains unaltered and user modifications are detailed.

NO WARRANTY
THE SOFTWARE AND ANY OTHER MATERIALS PROVIDED BY NVIDIA ON THE ENCLOSED CD-ROM ARE PROVIDED “AS IS.” NVIDIA DISCLAIMS ALL WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.

LIMITATION OF LIABILITY
NVIDIA SHALL NOT BE LIABLE TO ANY USER, DEVELOPER, DEVELOPER’S CUSTOMERS, OR ANY OTHER PERSON OR ENTITY CLAIMING THROUGH OR UNDER DEVELOPER FOR ANY LOSS OF PROFITS, INCOME, SAVINGS, OR ANY OTHER CONSEQUENTIAL, INCIDENTAL, SPECIAL, PUNITIVE, DIRECT OR INDIRECT DAMAGES (WHETHER IN AN ACTION IN CONTRACT, TORT OR BASED ON A WARRANTY), EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THESE LIMITATIONS SHALL APPLY NOTWITHSTANDING ANY FAILURE OF THE ESSENTIAL PURPOSE OF ANY LIMITED REMEDY. IN NO EVENT SHALL NVIDIA’S AGGREGATE LIABILITY TO DEVELOPER OR ANY OTHER PERSON OR ENTITY CLAIMING THROUGH OR UNDER DEVELOPER EXCEED THE AMOUNT OF MONEY ACTUALLY PAID BY DEVELOPER TO NVIDIA FOR THE SOFTWARE OR ANY OTHER MATERIALS.

More information and updates are available at:
http://developer.nvidia.com
http://www.awprofessional.com/

Versions for other operating systems will be available at:
http://developer.nvidia.com/CgTutorial