

Symbols and Numbers

2D antialiasing, 176
 2D slices of noise functions, 74
 2D texture mapping
 for clouds, 267
 limitations of, 10
 methods, 197–199
 3D Nature's World Construction Set, 580
 3D tables for controlling animation, 244–254
 accessing table entries, 245
 breeze effect using attractor, 247, 248, 252–253, 254
 combinations of functions, 251–254
 common dimensions for, 245
 flow into hole in wall, 253–254, 255
 functional flow field functions, 246–251
 functional flow field tables, 245, 246
 nonuniform spacing between entries, 244–245
 overview, 244–245
 vector field tables, 245
 wind effects, 252–254
 3D textures, converting to image maps, 197–199
 4D cellular texturing, 149
 4D noise functions, 84, 181

abs function (RenderMan), 29–30

abstraction, 2
 a-buffer rendering algorithm, 205
 additive cascades, multiplicative cascades vs., 440
Advanced RenderMan, 102, 129
Adventures of André & Wally B, The, 259, 260
 aerial perspective, 529–530
 aesthetic *n*-space model, 548–549, 559
 air currents. *See* wind
Air Force One, 473, 525
 algorithms
 a-buffer rendering algorithm, 205
 for cumulus clouds, 272, 274
 explicit noise algorithms, 82–83
 genetic algorithms vs. genetic programming, 558–559
 multipass pixel shader implementation of noise function, 423–425
 QAEB, 513–514
 real-time procedural solid texturing, 413–416
 REYES (PRMan), 103–106
 ridge algorithm for bump mapping, 187
 shadow table algorithm, 207–208
 volume-rendering algorithm for gases, 206
 Alias Dynamation, 524
 aliasing. *See also* antialiasing

brick texture example, 56, 57
 cyberspace and, 622
 defined, 52
 fractal octaves and, 585
 image textures vs. procedural textures and, 15
 index aliasing, 158–166
 maximum frequency and, 53, 54
 MojoWorld function fractals and, 602–604
 Nyquist frequency and, 53–54
 planetary rings example, 163–166
 in procedural textures, 55–56
 sampling rate and, 53, 54
 in signal processing, 53–54
 signal processing concepts, 52–56
 smoothstep function for avoiding, 31
 sources of, 158–160
 supersampling and, 54
 temporal, 175
 white noise and, 68
 Alias's Maya animation system. *See* Maya animation system
 ambient function (RenderMan), 21
 amplification
 data amplification algorithms for procedural geometry, 305, 312–314
 database amplification, 2

- amplification (*continued*)
 - by fractals, 436
 - intermediate representation and, 313
 - proceduralism and, 549
- amplitude of fractals, 433
- analytic prefiltering
 - box filter for, 61–62, 64–65
 - brick texture example, 64–66
 - filters with negative lobes, 62
 - integrals for, 3–4, 65
 - overview, 56–57
 - smoothstep function for, 62
 - summed-area table method, 63–64
- Animatek's World Builder, 580
- animating gaseous volumes, 235–243
 - helical path effects, 236–243
 - rolling fog, 238–239
 - smoke rising, 239–243
 - steam rising from teacup, 236–238
- animating hypertextures, 254–256
- animating solid spaces, 227–261
 - animation paths, 228–229
 - approaches, 227–228
 - changing solid space over time, 227, 229–232
 - ease-in and ease-out procedures, 229
 - hypertextures, 254–256
 - marble procedure for examples, 229
 - moving screen space point through solid space, 227, 228, 232–233, 235–243
 - solid textured transparency, 233–235
 - three-dimensional tables for controlling animation, 244–254
- animating volumetric procedural clouds, 279–283
 - implicit primitive animation, 280–283
 - procedural animation, 279–280
- animation
 - aliasing problems
 - magnified by, 157
 - cellular texturing for, 149
 - clip animation, 391
 - ease-in and ease-out procedures, 229
 - flame shader, noise-based procedural, 124–129
 - paths for, 228–229
 - of real-time effects, 291
 - textural limb animation, 387, 391–395
 - texture for facial movement, 395–409
 - textures for, 196–197
 - time-animated fractal noise, 181
- anisotropic1 shader, 114–115
- anisotropic2 shader, 115–116
- anisotropic shading models
 - development of, 8
 - Heidrich/Banks model, 112–116
- antialiased rendering of procedural textures, 369–376
 - background, 370
 - basic idea, 370–372
 - detailed description, 372–373
 - examples, 374–376
 - high-contrast filter, 371–372, 373–374
 - ifpos* pseudofunction for, 371, 372–373
 - nested conditionals and, 369
 - overview, 369, 376
- antialiasing, 52–67, 157–176, 369–376. *See also* aliasing
- alternatives to low-pass filtering, 66–67
- analytic prefiltering techniques, 56–57, 61–66
- blending between texture versions for, 66
- blur effects for, 175–176
- blurring color map for, 175
- box filter for, 61–62, 64–65
- brick texture example, 64–66
- bump mapping and, 59
- clamping method, 56, 59–61
- edges that result from conditionals, 369–376
- efficiency issues, 157
- emergency alternatives, 175–176
- exaggerating spot size for, 175
- filters with negative lobes for, 62
- fractal octave reduction for, 585
- high-contrast filter for, 371–372, 373–374
- if* functions vs. step functions for, 28
- image textures vs. procedural textures and, 15
- importance of, 157
- index antialiasing, 160–166
- integrals for, 63–64, 65
- low-pass filtering, 54
- methods for procedural textures, 56–67
- modeling input distribution, 161–162
- object space vs. screen space shading and, 106
- optimization and verification, 173–175
- PhotoRealistic RenderMan scheme, 55

- planetary rings example, 163–166
- in QAEB tracing, 516
- reference image for, 174
- by renderers, 55–56
- rendering procedural textures, 369–376
- rendering texture as 2D image, 176
- sampling and bumping, 170–173
- sampling rate determination, 57–59
- smoothstep function for, 62
- spot size calculation, 166–170
- stochastic sampling, 55, 67
- sum table for, 162–163
- summed-area table
 - method, 63–64
- supersampling, 10–13, 54, 66–67, 157
- temporal, 175
- test scene design, 174
- by texture instead of renderer, 170–173
- two-dimensional, 176
- API routines
 - interface between shaders and applications, 118–121
 - OpenGL, 118
 - in Stanford shading language, 118–120
- Apollo 13*, 473
- applytexture program, 111–112
- architectexture, 359–362
- area-weighted mesh atlas, 417–418
- art, computer's role in, 482
- artifacts
 - frequency multiplier and lattice artifacts, 88–89
 - grid-oriented artifacts, reducing, 348
 - from noise algorithms, 88–89, 180–181, 348
 - rotation matrices for hiding, 180, 181
 - seam artifacts, avoiding, 419–421
 - terrain creases as, 498
- ArtMatic, 557
- assembly language, for shading, high-level language vs., 100–101
- atlas
 - area-weighted mesh atlas, 417–418
 - based on clusters of proximate triangles, 418, 419
 - defined, 416
 - length-weighted atlas, 418
 - for real-time procedural solid texturing, 416–419
 - to support MIP mapping, 418–419
 - uniform meshed atlas, 416–417
- atmosphere, fractal properties of, 570
- atmospheric dispersion effects for gaseous phenomena, 205
- atmospheric models, 529–544. *See also* color perspective; GADDs (geometric atmospheric density distribution models)
- aerial perspective, 529–530
- Beer's law and homogeneous fog, 531–532
- curved, 569–570
- dispersion effects for gaseous phenomena, 205
- elements involved in, 529, 530
- exponential mist, 532, 534
- extinction, 531–532
- global context for landscape, 530
- integration schemes for, 529, 534–544
- minimal Rayleigh scattering approximation, 536, 539
- numerical quadrature with bounded error for radial GADDs, 542–544
- optical depth, 531
- optical paths, 531, 534
- physical models vs., 530
- radially symmetric planetary atmosphere, 534–536
- scattering models for, 529, 530, 536, 539
- terminology, 531
- trapezoidal quadrature for radial GADDs, 539–542
- atmospheric perspective, 529
- attractors
 - combinations of functions, 251–254
 - creating repulsors from, 246
 - extensions of spherical attractors, 249
 - overview, 247
 - spherical, 247–249
- Avatars, 391
- axioms in L-system, 308
- banded color, sine function for, 230
- band-limited fractals, 434
- bandwidth
 - memory bandwidth and performance tuning, 110
 - in signal processing, 52
- basic_gas function
 - overview, 214–215
 - patchy fog using, 216
 - power function effects on, 215–216
 - sine function effects on, 215, 216
 - steam rising from teacup using, 216–219, 220

- basis functions of fractals
 - defined, 582
 - manipulating for variety, 450
 - MojoWorld, 450, 582, 583, 590–597
 - overview, 432–433
 - for random fractals, 583
 - visual effects for homogeneous fBm terrain models, 497–498
- basketball, texture baking example for, 198–199
- Beer's law, atmospheric models and, 531–532
- bevel for bump mapping. *See* ridge for bump mapping
- bias function
 - gain function and, 39
 - for gamma correction, 37–38
- bias_s function
 - defined, 339
 - for tuning fuzzy regions, 339, 340
- bicubic patch modeling with L-system, 311
- bilinear filtering, seam artifacts and, 420–421
- billboards
 - for cloud modeling, 268
 - for gases, 209
- billowing clouds, 523–524
- binding
 - early vs. late, performance and, 117–118
 - of shaders, program responsibility for, 120
 - Stanford shading language early-binding model, 118
- black-body radiators, 461
- black-box textures, future of, 200–201
- blending capability of GPUs, 290
- blinking, 405. *See also* texture for facial movement
- blobby molecules, 3
- Blue Moon Rendering Tools (BRMT), 102
- blurring
 - blur effects for
 - antialiasing, 175–176
 - color map for antialiasing, 175
- bombing, random placement patterns based on, 91–94
- bounding volumes for procedural geometry, 330–332
- box filter for antialiasing, 61–62, 64–65
- boxstep function (RenderMan)
 - analytic prefiltering using, 61–62
 - brick texture antialiasing example, 64–65
- brick textures
 - aliasing in, 56, 57
 - antialiased example, 64–66
 - bump-mapped brick, 41–46
 - bump-mapped mortar, 44–45
 - determining whether point is in brick or mortar, 41
 - images rendered using, 42, 46, 66
 - offsetting alternate rows, 41
 - perturbed, 89–90
 - procedural texture generator, 39–41
 - pulses generating brick shape, 41
 - scoord and tcoord texture coordinates, 40
- BRMT (Blue Moon Rendering Tools), 102
- Bryce software, 580
- bump mapping. *See also* displacement mapping
 - adding to basic texture, 183
 - antialiasing and, 59, 170–173
 - bevel for three-dimensional appearance, 183, 184
 - brick mortar example, 44–45
 - brick texture example, 41–46
 - cellular textures, 140, 141, 142
 - correct vs. incorrect, 171–173
 - geometry of, 42
 - impressive textures' use of, 183
 - methods, 183–187
 - overview, 9, 41–43
 - ridge algorithm for, 183–187
 - rotation matrices for hiding artifacts, 180
 - texture spaces and, 45–46
 - for water ripples, 461, 463
- "Bursting," 17
- bushes, L-system for modeling, 318–320
- C language, translating
 - RenderMan shading language into, 19
- calc_noise function, 212–213
- calculatenormal function, described, 44
- calc_vortex procedure, 250–251
- camera space (RenderMan)
 - current space and, 24
 - solid textures and, 24–25
- Carefree Gum, "Bursting," 17
- Carmack, John, 130
- cascades, multiplicative vs. additive, 440
- ceil function (RenderMan), 33
- ceilf function (ANSI C implementation), macro alternative to, 33
- cellular texturing, 135–155
 - 2D and 4D variants, 149
 - basis functions, 136–140
 - density of feature points in, 144–145, 146–147, 149
 - distance metric for, 147–149

- extensions and alternatives, 147–149
- feature points defined, 136
- fractal combinations, 138–140, 141
- implementation strategy, 140–149
- isotropic property of, 142, 143, 145–147
- linear combinations of functions F , 137–138, 139
- “Manhattan” distance metric, 148
- mapping function values onto colors and bumps, 137
- modifying the algorithm, 136, 140–141
- neighbor testing, 144–145
- noise compared to, 135–136
- nonlinear combinations of polynomial products, 140
- partitioning space into cellular regions, 136–137, 142–144
- Poisson distribution of feature points, 145–147
- population table for, 145–147
- properties of functions F , 137
- random number generator for, 143
- sample code, 149–155
- speed vs. isotropy in, 146–147
- Cg language (NVIDIA), 111, 292–296
- checkerboard basis function (MojoWorld), 596–597
- checkerboard pattern generation, 39
- cirrus cloud models, 275–276, 277, 282, 283, 453–458
- Cirrus procedure, 275–276
- clamp function (RenderMan), 28–29
- clamping
 - closeness to texture pattern and, 60–61
 - described, 56
 - fade-out for sampling rate changes, 60
 - limitations of, 61
 - spectral synthesis
 - antialiasing using, 59–61, 86–87
 - for turbulence function, 86–87
- clip animation, 391
- clipping planes in QAEB tracing, 514–515
- cloth, hypertexture example, 357, 359, 360, 361
- cloudplane shader, 50
- clouds. *See also* fog; gases; volumetric cloud modeling and rendering
 - altitude and types of, 266
 - animating volumetric procedural clouds, 279–283
 - atmospheric dispersion effects, 267
 - basics, 263, 266–267
 - billowing, 523–524
 - challenge of, 448
 - cirrus cloud models, 275–276, 277, 282, 283, 453–458
 - cloud creatures, 273, 278
 - commercial packages for rendering, 284, 285
 - Coriolis effect, 458–460
 - cumulus cloud models, 272–274, 282
 - difficulties modeling, 263, 267
 - distorted noise function for, 450–453
 - ellipsoid surfaces for modeling, 195, 267
 - example photographs, 264–265
 - fBm-valued distortion for, 454–456
 - forces shaping, 263, 266
 - fractal solid textures for, 448–460
 - functional composition for modeling, 453–454
 - hardware acceleration for, 298–301
 - illumination models, 266–267
 - for interactive applications, 267–268
 - interactive models, 283–284, 285
 - jet stream warping, 279–280
 - MojoWorld for, 611
 - ontogenetic modeling of hurricane, 456–458
 - previous approaches to modeling, 267–268
 - psychedelic, 525, 526
 - puffy, 448–449
 - QAEB-traced hyper-textures for, 520–525, 526
 - real-time, 298–301
 - spectral synthesis for simulating, 50–51
 - stratus cloud models, 276, 277–278, 282, 283
 - surface-based modeling approaches, 267–268
 - on Venus, 458–460
 - visual characteristics, 266
 - volume perturbation for modeling, 299–301
 - volumetric modeling, 268–272
 - volumetric rendering, 272–279
 - weather forecasting and, 266
- color. *See also* RGB color
 - black-body radiators, 461
 - blurring color map for antialiasing, 175
 - color mappings, 181–182
 - color splines, 182, 189–192

- color (*continued*)
 - color table equalization, 189, 190–192
 - fBm coloring, 477–478
 - for fire, 461
 - GIT texturing system, 478–480
 - index aliasing, 158–163
 - layering texture patterns, 25–26
 - L-system specification, 311–312
 - for marble using NVIDIA's Cg language, 293–294
 - “multicolor” texture, 482–485
 - normalizing noise levels for, 189–192
 - plastic shader output, 21
 - random coloring methods, 477–485
 - sine function for banding, 230
 - solid color texture, 196
 - spline function for mapping, 36
 - usage in this book, 182
- color perspective. *See also* atmospheric models
 - in *Carolina*, 534
 - defined, 529
 - in *Himalayas*, 537
 - in planetary atmosphere, 538
 - raw form, 538
 - Rayleigh scattering and, 536
- color splines
 - color table equalization, 189, 190–192
 - normalizing noise levels, 189–192
 - overview, 182
- color table equalization
 - histogram of noise values, 190, 191
 - mapping percentage levels to noise values, 190–192
 - need for, 189, 192
 - normalization plot, 190, 191
- complexity
 - fractal vs. nonfractal, 431
 - fractals and visual complexity, 429, 506, 567–571
 - hyperspace in MojoWorld and, 588–589
 - realism and, 434–435
 - as work, 434
- composition. *See* functional composition
- computational precision issues for GPUs, 288
- compute_color function for raymarcher renderer, 350, 351
- computer graphics research community, 576, 578–579
- “Computer Rendering of Stochastic Models,” 576
- computer's role in art, 482
- compute_shading function for raymarcher renderer, 350
- conditional functions
 - abs (RenderMan), 30
 - antialiased rendering of edges resulting from, 369–376
 - clamp (RenderMan), 28–29
 - max (RenderMan), 28–29
 - min (RenderMan), 28–29
 - smoothstep (RenderMan), 30–31
 - step (RenderMan), 27–28
- conferencing, electronic, 391
- conservation of angular momentum, spiral vortex functions based on, 251
- constant density medium for modeling gases, 204
- convective cloud modeling, 268
- convolution kernel, 595
- convolution noise
 - lattice, 78–80
 - sparse, 80–82
- coordinate systems in RenderMan, 24. *See also* texture spaces
- Coriolis effect, 458–460
- cosf function (ANSI C implementation), RenderMan cos function vs., 31
- cosine function
 - graph of, 32
 - overview, 31
 - for rotation on helical path, 228
- CPUs
 - GPUs vs., 100, 102, 109
 - migrating procedural techniques to GPUs from, 287–289
 - real-time procedural solid texturing on, 421
 - virtualization supported by, 109
- crop circles, PGI for modeling, 326
- crossover scales
 - of fractals, 434
 - in MojoWorld, 606–607
- cubic Catmull-Rom spline interpolation for value noise, 71–72
- cumulus cloud models, 272–274, 282
- cumulus procedure, 272, 274
- current space, described, 24
- cyberspace
 - aliasing and, 622
 - as context for information content, 621, 622, 623
 - defined, 617
 - features of “natural” cyberspace, 619–620
 - fractal geometry of, 620–622
 - MojoWorld photorealistic renderer and, 617–618

- social aspects of, 621–622
 - vision of, 618–620
- cyclic scene graphs, 320, 321
- cyclone procedure, 456–458
- dance
 - bias and gain controls for, 397
 - emotional gestures in, 391
- Dark Tree, 447
- data amplification algorithms
 - for procedural geometry, 312–314
 - described, 305
 - intermediate representation and, 313
 - in L-system, 312–314
 - for particle systems, 313–314
- data types, GPU support for, 108–109
- database amplification, 2
- “Death Star” surface, 148
- debugging, image textures vs. procedural textures and, 14
- declarative components of procedural techniques, 12
- deep shadow map for gases, 208
- degrees of freedom in n -spaces, 548
- DEM (digital elevation map)
 - format, 494
- density
 - constant, for modeling gases, 204
 - of feature points in cellular texturing, 144–145, 146–147, 149
 - function for hypertexture explosion example, 355
 - height attenuation for rising steam, 218–219
 - height attenuation for smoke dispersion, 220
 - implicit density functions
 - for cloud modeling, 270–271
 - power function and gas density, 215–216
 - raymarcher renderer user-defined functions, 350–352
 - scaling vector returned by functional flow field functions, 246–247
 - smoke density function, 367–368
 - spherical attenuation to confine steam within cup radius, 216–218
 - turbulence-based, for cloud modeling, 271
- density_function for raymarcher renderer, 350
- depth buffer renderers, implicit vs. explicit procedures in, 13
- depth-of-field effects, object space vs. screen space shading and, 105–106
- derivative discontinuities in turbulence function, 86
- diffuse function (RenderMan), 21
- diffuse model, 7–8
- digital elevation map (DEM)
 - format, 494
- digital terrain elevation data (DTED) format, 494
- dilation symmetry
 - in fractal geometry, 621
 - fractals as, 431, 573–574
- dimension, 588
- Direct3D, OpenGL vs., 102
- displacement mapping. *See also* height fields
 - in MojoWorld, 610–611
 - object space vs. screen space shading and, 104, 105–106
 - overview, 9
 - scanline rendering vs. ray tracing and, 511
 - for water ripples, 461–463
- distorted fractal functions in MojoWorld, 606
- distorted noise (DistNoise)
 - function, 450–453
 - for altering fractal basis function, 450
 - C++ version, 450
 - RenderMan version, 452–453
 - terse version, 452
 - undistorted noise function vs., 451–452
 - VecNoise function in, 450–451
- domain distortion
 - for cirrus cloud modeling, 453–454
 - DistNoise function based on, 450–453
 - in MojoWorld, 604–606
- domain function dimensions in MojoWorld, 587–588
- dot product capability of GPUs, 290
- Dr. Mutatis program
 - demise of, 550
 - images generated by, 558–563
- DTED (digital terrain elevation data) format, 494
- Du and Dv functions (RenderMan), 58–59
- du and dv variables (RenderMan), 58
- dynamic bounding volumes, 331
- earth textures. *See also* MojoWorld; procedural fractal terrains
 - fractal solid textures for earth textures, 466–477
 - Gaea (Earth-like planet), 467–472

- earth textures (*continued*)
 - sedimentary rock strata, 466–467, 468
 - Selene (Moon), 473–477
- ease-in and ease-out procedures, 229
- edge events, 370. *See also*
 - antialiased rendering of procedural textures
- egg shape
 - for creating hypertextures, 354
 - explosion inside, 355, 356, 357
- electronic conferencing, 391
- emittance, volume rendering
 - with surface textures and, 196
- emotive gesturing. *See* textural limb animation
- environment function (RenderMan), 22
- environment mapping, 9
- E-on Software's Vue d'Esprit, 580
- Euclidean geometry, fractal geometry vs., 620–621
- exact self-similarity, statistical self-similarity vs., 435, 575
- explicit procedures
 - defined, 12
 - implicit procedures vs., 12–14
- explosions
 - hypertexture example, 354–356
 - inside an egg, 355, 356, 357
 - pyroclastic flow, 523–524
- expression trees, 553–556
- extinction in atmospheric models, 531–532
- F (turtle graphics symbol), 308
- fabs function with turbulence functions, 369
- faceforward function (RenderMan), 21
- Facial Action Coding System (FACS), 396
- facial expressions. *See* texture for facial movement
- fade in, solid color texture for, 196
- fast Fourier transform (FFT), spectral synthesis and, 49
- fBm. *See* fractional Brownian motion (fBm)
- feature points in cellular texturing
 - defined, 136
 - density of, 144–145, 146–147, 149
 - isotropic distribution of, 142, 143, 145–147
 - precomputed, 149
 - testing regions of space for, 142–145
- feature spaces for textures
 - overview, 25
 - star texture, 46
- FFT (fast Fourier transform), spectral synthesis and, 49
- filtering capability of GPUs, 290
- fire. *See also* volumetric procedural modeling and texturing
 - colors for, 461
 - fBm-valued distortion for, 460–461, 462
 - fractal solid textures for, 460–461
 - noise-based procedural flame shader, 124–129
 - particle system for wall of fire, 257, 258
 - pyroclastic flow, 523–524
 - QAEB-rendered fireballs, 525
- flagstone texture, 140, 142
- flame. *See* fire
- flame procedure, 460–461
- flexibility
 - GPU issues, 288–289
 - procedural techniques and, 2
- floor function (RenderMan), 32–33
- floorf function (ANSI C implementation), macro alternative to, 33
- flow field functions. *See* functional flow field functions
- flow noise, 384–387
 - example flow textures, 387, 388–389
 - need for, 384–385
 - pseudoadvection in, 386–387
 - rotation gradients for, 385–386
- fog. *See also* gases; volumetric procedural modeling and texturing
 - animating solid textured transparency, 233–235
 - Beer's law and homogeneous fog, 531–532
 - exponential mist, 532, 534
 - hardware acceleration for space-filling fog, 297–298
 - patchy fog modeling, 216
 - rolling fog animation, 238–240
- fog procedure, 233–234
 - described, 233
 - helical path for, 233
 - values for, 235
- Fourier synthesis
 - basis function for, 497
 - for cloud modeling, 267
 - for controlling transparency, 204
 - point evaluation vs., 490
- Fourier transform
 - domain change from, 49
 - inverse, 49

- signal processing and, 52–53
 - for spectral synthesis, 49–51
- frac macro, 65
- fractal dimension
 - amplitude scaling and, 89
 - defined, 582–583
 - for homogenous fBm terrain models, 495–497
 - overview, 432–433
 - for random fractals, 583–584
 - roughness of surface and, 444, 495–496, 583–584
- fractal geometry of cyberspace, 620–622
- Fractal Geometry of Nature, The*, 575, 576
- fractal noise. *See* noise functions; Perlin's noise function
- Fractal Planetrise*, 576
- fractal planets. *See* MojoWorld
- fractal solid textures, 447–487
 - clouds, 448–460
 - earth, 466–477
 - fire, 460–461, 462
 - iterative design of, 447–448
 - planetary rings, 485–487
 - random coloring methods, 477–485
 - water, 461–465
- fractal sum of pulses. *See* sparse convolution noise
- fractals, 429–445. *See also* MojoWorld; multifractals; procedural fractal terrains
 - amplification by, 436
 - amplitude, 433
 - band-limited, 434
 - basis function, 432–433, 450, 497–498, 582, 583
 - cellular texture combinations, 138–140, 141
 - complexity of, 431, 434–435
 - computer graphics and, 429
 - defined, 431, 571
 - difficulties for understanding, 430
 - as dilation symmetry, 431, 573–574
 - distribution of fractal functions, 190
 - fractal dimension property, 89, 432–433, 444, 495–497
 - fractal forgeries, 567–568
 - fractal increment, 432
 - fractional Brownian motion (fBm) and, 433
 - further information, 429
 - heuristic approach to, 431
 - history of fractal terrains, 575–582
 - lacunarity of, 433, 444, 583, 585–586
 - as language of visual complexity, 429, 506, 567
 - level of detail (LOD) in, 433–434, 437–438, 511
 - literature on, 579
 - local dimensionality in, 432
 - lower crossover scale, 434
 - mathematical history of fractal terrains, 575–576
 - mathematical imaging of fractal terrains, 576
 - misunderstanding of, 430–431
 - for modeling gases, 204
 - monofractal graphics, 445
 - multifractals, 438, 440–442, 569
 - “naturalness” of, 485
 - in nature, 567–568, 572–573
 - nonfractal complexity vs., 431
 - octaves in, 433–434, 444, 583, 584–585
 - ontogenetic modeling and, 442–444
 - poem about, 435
 - as primary building blocks, 430
 - procedural fBm, 436–438, 440
 - procedural fractal terrains, 489–506
 - proceduralism and, 436
 - random, 574–575, 582–586, 611, 613
 - self-similarity in, 496–497, 572–573
 - simplicity of, 435
 - software, history of, 579–580
 - space-filling hypertexture example, 357, 358–359
 - spatial frequency of basis function, 433
 - statistical self-similarity vs. exact self-similarity, 435
 - for synthesizing cloud images, 267
 - turbulence and, 435–436
 - in turbulence function power spectrum, 85
 - upper crossover scale, 434
 - uses for, 434–436, 506, 608–611
 - visual complexity and, 567–571
- fractal sum function
 - power spectrum of, 86, 87
 - turbulence function vs., 86
- fractional Brownian motion (fBm)
 - approximating, 89
 - basis function effects, 497–498
 - for color perturbation, 470
 - described, 599–600

- fractional Brownian motion (*continued*)
 - DistNoise function based on, 450–453
 - fBm-valued distortion for clouds, 454–456
 - fBm-valued distortion for fire, 460–461, 462
 - fractals and, 433
 - homogenous fBm terrain models, 495–498
 - level of detail and, 437–438
 - monofractals using, 599–600
 - multiplicative-cascade multifractal variation, 440–441
 - in PGI, 327–329, 330
 - power spectrum of, 433
 - procedural, 436–438, 440
 - random fBm coloring, 477–478
 - as weighting function in windywave procedure, 465
- fragment processing in GPUs, 287, 289
- frequency of fractal basis function, 433
- function fractals in MojoWorld, 602–604
- functional composition
 - for cirrus cloud modeling, 453–454
 - defined, 453
 - in DistNoise function, 453
 - expression trees and, 553–554
 - for texture pattern generation, 26
- functional flow field functions
 - attractors, 247–249
 - breeze effect using attractor, 247, 248, 252–253, 254
 - combinations of functions, 251–254
- density_scaling parameter, 246–247
- direction parameter, 246, 247
- extensions of spherical attractors, 249
- for flow into hole in wall, 253–254, 255
- information returned by, 246–247
- overview, 246–247
- percent_to_use parameter, 246, 247
- repulsors, 246
- spherical attractor function, 248–249
- spiral vortex functions, 249–251
- velocity parameter, 246, 247
- wind effects using, 252–253
- functional flow field tables
 - accessing table entries, 245
 - combinations of functions, 251–254
 - functions, 246–251
 - overview, 245, 246
- functional programming, 26
- fur, volume perturbation for, 301, 302
- fuzzy bobbies for modeling gases, 204
- fuzzy regions
 - for bias_b function for tuning, 339, 340
 - combining solid textures with, 338
 - defined, 338
 - gain_g function for tuning, 339–340
- GADDs (geometric atmospheric density distribution models). *See also* atmospheric models
 - for exponential mist, 532, 534
- homogenous and isotropic, 531–532
- local vs. global, 350
- numerical quadrature with
 - bounded error for radial GADDs, 542–544
 - overview, 530
 - for radially symmetric planetary atmosphere, 534–536
 - with Rayleigh scattering approximation, 539
 - trapezoidal quadrature for radial GADDs, 539–542
- Gaea (Earth-like planet) model, 467–472
 - climactic zones, 468–469
 - coastline, 469
 - continents and oceans, 467–468
 - deserts, 469, 470
 - fractals in, 467
 - terrain procedure, 470–472
- gain function
 - bias function and, 39
 - remapping unit interval with, 38–39
- gain_g function
 - defined, 340
 - for tuning fuzzy regions, 339–340
- gamma correction functions
 - bias, 37–38
 - for CRT display systems, 37, 38
- gamma correction texture, 196
- gases. *See also* animating solid spaces; clouds; fire; volumetric procedural modeling and texturing
 - animating gaseous volumes, 235–243
 - animating solid textured transparency, 233–235

- basic gas shaping, 214–224
- breeze effect using attractor, 247, 248, 252–253, 254
- flow into hole in wall, 253–254, 255
- geometry of, 211–224
- hardware acceleration for smoke and fog, 297–298
- helical path for, 228
- historical overview of modeling approaches, 204–205
- illumination model for, 207
- noise and turbulence functions for, 211–214
- particle systems for, 209
- patchy fog modeling, 216
- power and function effects on, 214–216
- rolling fog animation, 238–240
- self-shadowing, 207–208
- sine function effects on, 215, 216
- smoke column examples, 219–224, 239–243
- solid spaces framework for modeling, 209–211
- steam rising from teacup examples, 216–219, 220, 236–238
- three-dimensional tables for controlling animation, 244–254
- turbulence for simulating air currents, 238
- usage in this book, 203
- GDM (goal determination module), 394–395
- generalized Impressionistic texture. *See* GIT (generalized Impressionistic texture) texturing system
- genes, 551, 552–555
 - genetic algorithms, genetic programming vs., 558–559
 - genetic programming. *See also* genetic textures
 - biological analogy for, 550–552
 - DNA and, 551–552
 - examples, 557–558
 - expression trees, 553–556
 - further information, 558
 - future directions, 560–562
 - genes, 551, 552–555
 - genetic algorithms vs., 558–559
 - genomes, 551
 - genotypes, 551
 - implementation, 555
 - mutation, 551, 554
 - parameter proliferation problem and, 548
 - phenotypes, 551, 554
 - sexual reproduction, 551, 554
 - unnatural selection in, 552
 - genetic textures, 547–563. *See also* genetic programming
 - aesthetic *n*-space model, 548–549
 - basis vectors, 556
 - control vs. automaticity, 549–550
 - evolutionary biological model, 550–555
 - expression trees for, 553–556
 - future directions, 560–562
 - genetic programming and genetic art examples, 557–558
 - genetic programming vs. genetic algorithms, 558–559
 - library of genetic bases, 556–557
 - parameter proliferation problem, 547–548
 - root node interpretation, 555–556
- genomes, 551
- genotypes, 551
- geometric atmospheric density distribution models. *See* GADDs (geometric atmospheric density distribution models)
- geometric modeling
 - advanced techniques, 2–3
 - bump map geometry, 42
 - evolution of, 2–3
 - star geometry, 47
- geometric normal, GPU support lacking for, 106
- geometrical calculations, 7
- geometry mapping with PGI, 326–327
- geometry of gases, 211–224
- geometry, procedural synthesis of. *See* procedural synthesis of geometry
- gesturing. *See* textural limb animation
- GIT (generalized Impressionistic texture) texturing system, 478–480
 - Impressionistic image processing filter, 479–480
 - mathematical model underlying, 478–479
 - overview, 478–479
- global shading models, 8
- noise function. *See* gradient noise
- goal determination module (GDM), 394–395
- GPUs. *See also* hardware acceleration of procedural techniques
 - assembly languages for, 100–101
 - blending capability, 290
 - commercially supported programming languages, 111

GPUs (*continued*)

- computational precision issues, 288
- CPUs vs., 100, 102, 109
- data types, 108–109
- dot product capability, 290
- filtering capability, 290
- flexibility issues, 288–289
- fragment processing in, 287, 289
- future hardware and programming languages, 130–131, 410
- interpolation capability, 290
- language interface, 288
- levels of operation in the pipeline, 289
- limitations and restrictions, 102–103, 106–110
- memory bandwidth and performance tuning, 110
- migrating procedural techniques from CPUs to, 287–289
- parallelism in, 102, 103, 106–108
- real-time procedural solid texturing on, 421–425
- real-time procedural techniques and, 1
- resource limits, 109
- REYES shading contrasted with, 103–106
- separation of surface and light properties and, 117
- SIMD computation model in, 107–108
- SPMD computation model in, 108
- storage issues, 103, 289
- vertex processing in, 287, 289
- virtualization support lacking in, 109

gradient noise

- 2D slice of, 74
- flow noise, 384–387
- generating value for single integer lattice point, 76
- graph of, 73
- initializing table of pseudorandom gradient vectors, 73–75
- lattice noise, 69–70
- lookup table for, 181
- overview, 72–77
- Perlin's function, 69, 73, 75–76, 340–348
- for perturbed regular patterns, 89–90
- power spectrum of, 75, 77
- trilinear interpolation, 76–77
- value-gradient noise, 77–78

Graphics Gems III, 180

graphics processors. *See* GPUs; hardware acceleration of procedural techniques

grass modeling

- bounding volumes for, 331
- geometry mapping for, 326–327
- iterative instancing for, 322–323
- meadows, 327–329
- PGI for, 322–323, 326–329, 331

grid tracing, height fields and, 494

GUI for textures. *See* user interface for textures

hardware acceleration of procedural techniques, 287–302. *See also* GPUs

animated real-time effects, 291

common acceleration techniques, 289–291

dummy geometry and

depth culling, 290

example accelerated/real-time textures and models, 291–301

general issues, 287–289

marble texture, 292–297

migrating procedural techniques from CPUs to GPUs, 287–289

multilevel procedural models, 290–291

noise functions, 291

precomputation for, 289–290

real-time clouds and procedural detail, 298–301, 302

real-time procedural solid texturing, 416, 421–425

smoke and fog, 297–298

tasks beyond designers' intentions, 290

texture mapping, 208, 209

turbulence functions, 291–292

hardware texture mapping

for modeling and rendering gases, 209

for shadowing gases, 208

hashing in lattice noise generation, 69–70

Heidrich/Banks anisotropic shader

example using explicit computations, 113–115, 116

example using implicit computations, 115–116

explicit vs. implicit version, 116

overview, 112–113

height fields

defined, 491

as displacement maps, 511

file formats, 494–495

- for light diffusion effects simulation, 204
 - post spacing side effects, 509
 - in QAEB tracing, 511, 516–517
 - ray tracing and, 494
 - single altitude value per grid point in, 493
 - speedup scheme for, 513, 516–517
 - as storage scheme for terrains, 491
- helical paths
 - for animating gaseous volumes, 228, 236–243
 - for fog animation as solid textured transparency, 233
 - for marble animation, 232–233
 - for rolling fog animation, 238–239
 - for smoke rising animation, 239–243
 - for smoke simulation, still image, 220, 222–224
 - for space-filling fog, 298
 - for steam rising from tea-cup animation, 236–238
- Heney-Greenstein functions for illumination of gaseous phenomena, 207
- Hermite blending functions for bump mapping ridge, 186
- Hermite noise
 - 2D slice of, 74
 - described, 78
 - graph of, 73
 - power spectrum of, 75, 78
- Hermite spline interpolation in noise functions, 490, 498
- heterofractal function in MojoWorld, 601, 602
- heterogeneous terrain models, 498–506
 - hybrid multifractal terrain, 502–505
 - multiplicative multifractal terrains, 505–506
 - real landscapes and, 498, 500
 - smooth valleys at all altitudes, 502–505
 - smoother low-lying areas, 500–502
 - statistics by altitude, 500–502
- Hetero_Terrain procedure, 500–501
- hierarchy of surflets, 381–382
- high frequencies. *See also*
 - aliasing; antialiasing; Nyquist frequency
 - aliasing and, 53, 54, 370
 - edge events as sources of, 370
 - if function generation of, 47
 - non-edge high-frequency events, 370
 - step function generation of, 54, 57
- high-albedo reflection models
 - for clouds, 266
 - low-albedo models vs., 205
- high-contrast filter for
 - antialiasing, 371–372, 373–374
- HighEnd3D Web site, 284
- homogenous, defined, 438
- homogenous fBm terrain models, 495–498
 - basis function effects, 497–498
 - fractal dimension, 495–497
- homogenous procedural fBm, 438
- HSV color, transforming RGB to, texture for, 196
- Human Genome Project, 551
- humanlike figures
 - textural limb animation for emotive gesturing, 387, 391–395
 - texture for facial movement, 395–409
- hurricane, ontogenetic modeling of, 456–458
- hybrid multifractal terrains, 502–505
 - HybridMultifractal procedure, 502–504
 - RidgedMultifractal procedure, 504–505
- hyperspace in MojoWorld, 588–590
 - complexity and, 588–589
 - defined, 588
 - dimension defined, 588
 - Parametric Hyperspace, 589–590
- hypertextures
 - animating, 254–256
 - architecture, 359–362, 363
 - defined, 338
 - editing levels for, 352–353
 - egg shape for, 354
 - explosions example, 354–356
 - interaction with, 352–353
 - life forms example, 355, 356, 357
 - liquid flowing through hole in wall, 254–255
 - methods of combining
 - fuzzy shapes with solid textures, 338
 - NYU torch example, 362, 364
 - QAEB-traced, 520–525, 526
 - raymarcher renderer for, 348–352
 - smoke examples, 364–368
 - solid textures as precursor to, 337–338
 - space-filling fractals example, 357, 358–359
 - sphere shape for, 353–354
 - surflets for storing, 376–384
 - volumetric marble formation, 255–256

- hypertextures (*continued*)
 - woven cloth example, 357, 359, 360, 361
 - z-slicing, 35
- if function (RenderMan)
 - antialiasing and, 28
 - high frequencies generated by, 47
 - replacing with step function, 28–29
- ifpos pseudofunction, 371, 372–373
- illumination models. *See* shading models
- image maps
 - converting 3D textures to, 197–199
 - textures vs., 196–197
- “Image Synthesizer, An,” 590
- image textures
 - perturbed, 90–91
 - PhotoRealistic RenderMan antialiasing for, 56
 - procedural textures vs., 14–15
- implicit functions for cloud modeling, 269, 270–272
- implicit models, 10
- implicit primitive animation for clouds, 280–283
- implicit procedures
 - defined, 12–13
 - explicit procedures vs., 12–14
 - in RenderMan shading language, 15
- implicit surfaces, 3
- imposters
 - for cloud modeling, 268–269
 - defined, 268
 - for gases, 209
- index aliasing
 - defined, 159
 - modeling input distribution, 161–162
- planetary rings example, 163–166
- reducing, 160–163
- sources of, 158–159
- sum table for, 162–163
- transformation of scalar functions and, 159–160
- inductive instancing, 322–323
- init_density_function for raymarcher renderer, 350
- instances
 - defined, 315
 - inductive instancing, 322–323
 - parameter passing and, 321–322
 - of scene graphs, 315
- integrals, antialiasing using, 63–64, 65
- integration schemes
 - numerical quadrature with bounded error for radial GADDs, 542–544
 - as requirement for atmospheric models, 529, 530
 - trapezoidal quadrature for radial GADDs, 539–542
- intelligent textures, 200
- interactivity
 - cloud modeling approaches for, 267–268
 - for hypertextures, 352–353
 - in previewer for textures, 194
 - real-time vs. offline shading and, 98, 99, 100
- intermediate representation, 313
- interpolation capability of GPUs, 290
- inverse Fourier transform
 - defined, 49
 - domain change from, 49
 - noise generation using, 82–83
 - for spectral synthesis, 49–51
- irregular procedural textures, 67–94
 - noise functions for, 67–83
 - perturbed image textures, 90–91
 - perturbed regular patterns, 89–90
 - pseudorandom number generation and, 67–68
 - random placement patterns, 91–94
 - RenderMan noise functions for, 83–85
 - spectral synthesis, 85–89
 - time-dependent textures, 84
 - white noise and, 67–68
- isocurves, 13
- isosurfaces
 - defined, 3
 - implicit models, 10
 - overview, 13
 - surflets and, 383
- isotropic property
 - of cellular texturing function, 142, 143, 145–147
 - of noise function, 68, 180
 - of procedural fBm, 438
- iterated function systems, 320
- iteration
 - in fractal solid texture design, 447–448
 - iterative instancing, 322–323
 - in scientific discovery, 447–448
 - in shader development process, 99, 130
- jaggies. *See* aliasing; antialiasing
- Java, procedural geometric modeling and, 333

- jet stream warping for clouds, 279–280
- Kelvin-Helmholtz shearing in clouds, 266
- KISS (Keep it simple, stupid) principle, 443
- knot values in spline function (RenderMan), 34
- Koch snowflake. *See* von Koch snowflake
- labels in L-system, 312
- lacunarity
 - defined, 89, 583
 - of fractals, 433, 444, 583, 585–586
 - noise artifacts and, 180
- Lambertian (diffuse) model, 708
- landscapes. *See* atmospheric models; earth textures; MojoWorld; plant modeling; procedural fractal terrains
- language interface, CPUs vs. GPUs, 288
- lapped textures, 11
- lattice convolution noise
 - 2D slice of, 74
 - graph of, 73
 - implementation of, 78–80
 - power spectrum of, 75, 80
- lattice noises, 69–80
 - artifacts, 180
 - frequency multiplier and lattice artifacts, 88–89
 - gradient noise, 72–77
 - hashing technique, 69–70
 - integer lattice, 69
 - lattice convolution noise, 78–80
 - for modeling gases, 211–214
 - optimizing, 214
 - overview, 69–70
 - PRNs for, 69
 - value noise, 70–72
 - value-gradient noise, 77–78
- laughing, 407–408. *See also* texture for facial movement
- layering
 - RGBA textures and, 26
 - for texture pattern generation, 25–26
- lazy evaluation algorithms for procedural geometry
 - client-server relationship and, 314
 - described, 305, 314
 - L-system and, 315
 - for spatial coherence data structures, 314–315
- LED display, texture for, 196–197
- length-weighted atlas, 418
- lerping
 - layering texture patterns and, 26
 - LERP function in antialiasing, 371, 373
 - in rasterization phase for real-time procedural solid texturing, 414
- level of detail. *See* LOD (level of detail)
- life forms hypertexture example, 355, 356, 357
- light properties, 116–117
- light shaders, surface shaders vs., 117
- lighting
 - photographic texture images and, 22–23
 - surflets for, 376–377
- lighting models. *See* shading models
- limb animation. *See* textural limb animation
- linear basis function (MojoWorld), 596
- linear interpolation. *See* lerping
- liquid. *See also* water
 - flow into hole in wall, 253–254, 255
 - flow noise for, 384–387, 388–389
 - volumetric marble formation, 255–256
- LISP, functional composition in, 26
- Listerine (RenderMan examples), 18–19
- local dimensionality in fractals, 432
- local shading models, 8
- LOD (level of detail)
 - in fractals, 433–434, 437–438, 511
 - procedural fBm and, 437–438
- low-albedo reflection models
 - for clouds, 266
 - high-albedo models vs., 205
- lower crossover scale
 - of fractals, 434
 - in MojoWorld, 607
- low-pass filtering
 - analytic prefiltering techniques, 56–57, 61–66
 - antialiasing and, 54
 - clamping, 56, 59–61
 - defined, 54
 - in lattice noise generation, 69
 - sampling rate determination, 57–59
 - of white noise, 68
- L-system, 307–312. *See also* procedural geometric instancing (PGL); scene graphs
 - axioms, 308
 - bicubic patch modeling, 311
 - bush modeling, 318–320
 - color and texture specification, 311–312
 - context-free and context-sensitive aspects, 308

- L-system (*continued*)
 - data amplification paradigm in, 312–314
 - described, 305
 - development of, 307
 - intermediate representation in, 313
 - labels for debugging or annotation, 312
 - lazy evaluation and, 315
 - operating system command execution, 312
 - for particle systems, 313–314
 - PGI compared to, 329–330
 - PGI enhancements, 321–330
 - for plant modeling, 308–310, 311, 313
 - polygon modeling, 311
 - productions, 307, 308
 - scene graphs, 315–321
 - shortcomings of, 305–306
 - turtle graphics symbols in, 305, 306, 308–312
 - von Koch snowflake modeling, 308, 310–311
- luminosity, volume rendering
 - with surface textures and, 196
- luna procedure, 474–476
- Mandelbrot set, texture using, 197
- “Manhattan” distance metric in cellular texturing, 148
- mapping from unit interval to itself, 37–39
- marble function, 229, 292
- marble texture
 - animated volumetric marble formation, 255–256
 - animating by changing solid space over time, 229–232
 - animating by moving through solid space, 232–233
 - formation from banded rock, 229–232
 - frequency multiplier for, 88–89
 - hardware acceleration for, 292–297
 - NVIDIA Cg language implementation, 292–296
 - for solid texture animation, 229–232
 - spectral synthesis for, 87–89
- marble_color function, 88, 229
- marble_forming procedure, 230–231
- marble_forming2 procedure, 231–232
- Marble_Vertex.cg shader, 294–296
- master, defined, 315
- matte shading model, plastic
 - shader and, 21
- max function (RenderMan), 28–29
- Maya animation system
 - cloud modeling in, 267, 284, 285
 - MEL scripts, 284, 285
 - volumeGas plug-in, 284, 285
 - Web sites, 284
- meadows, PGI for modeling, 327–329
- memory
 - GPU resource limits, 109
 - GPU restrictions on access, 107
 - performance tuning and bandwidth, 110
- meta-balls, 3
- MetaCreations Skunk Works, 493
- metal shading model, plastic
 - shader and, 21–22
- “Methods for Realistic Landscape Imaging,” 576
- Mie scattering, aerial perspective and, 530
- min function (RenderMan), 28–29
- MIP mapping, atlases supporting, 418–419
- mist, exponential, 532, 534
- mix function (RenderMan), 25
- mod function (RenderMan), 31–33
- MojoWorld, 565–615
 - aliasing from function fractals, 604
 - basis functions, 450, 582, 583, 590–597
 - building a virtual universe, 571
 - checkerboard basis function, 596–597
 - for clouds, 611
 - crossover scales, 606–607
 - dimensions of domain and range functions, 587–588
 - displacement maps, 610–611
 - distorted fractal functions, 606
 - domain distortion, 604–606
 - driving function parameters with functions, 607
 - ease of, 447
 - experimenting with, 613–614
 - fractal dimension and, 582–584
 - fractals and visual complexity, 567–571
 - fractals overview, 571–575
 - function fractals, 602–604
 - future work, 581–582, 614–615
 - Graph Editor, 606, 607
 - heterofractal function, 601, 602
 - history of fractal terrains, 575–582
 - hyperspace, 588–590
 - lacunarity of fractals, 583, 585–586

- linear basis function, 596
- MojoWorld Generator, 590
- MojoWorld Texture Editor, 589, 590, 604
- monofractals, 599–600
- mountain building, 567–568
- multifractals, 600–602, 603
- for nebulae, 611, 612
- octaves of fractals and, 583, 584–585
- Perlin basis function, 590–591, 592, 593, 594
- photorealistic renderer, 585, 617–618
- planet building, 568–570, 611, 612
- Pro UI, 566–567, 606
- random fractals, 582–586
- real-time renderer, 585, 617
- reason for inclusion in this book, 565–566
- ridged basis function, 590–591, 593, 594
- seed tables, 597–599
- SIGGRAPH 2001 presentation of, 200
- sine wave basis function, 596
- sparse convolution basis function, 593, 595
- steps basis function, 596, 597
- surface textures, 589, 608–609
- for terrains, 609–610
- texture engine, 589
- transporter coordinates, 589
- Voronoi basis function, 591, 593, 594, 597, 599
- MojoWorlds, 467
- molten_marble procedure, 255–256
- monofractals in MojoWorld, 599–600
- Monte Carlo sampling method
 - in Perlin's noise function, 342
 - in volume-rendering algorithm, 206
- Moon texture. *See* Selene (Moon) texture
- motion blur, object space vs. screen space shading and, 105–106
- mouse abdomen visualization, 122
- moving_marble procedure, 232–233, 236
- “multicolor” texture, 482–485
 - goal of, 482–483
 - multicolor shader code, 484
 - “naturalness” of, 485
 - steps for, 483–484
- multifractals, 438, 440–442. *See also* MojoWorld; procedural fractal
 - terrains
 - defined, 440
 - described, 569
 - fBm variation, 440–441
 - hybrid multifractal terrain, 502–505
 - MojoWorld, 600–602, 603
 - multiplicative cascades and, 440
 - multiplicative multifractal terrains, 505–506
 - procedural textures and, 442
 - realism and, 438, 440
 - terrain patch example, 441
- multipass pixel shader implementation of noise function, 422–425
- multiplicative cascades, additive cascades vs., 440
- mutation, 551, 554
- Natural Graphics' Natural Scene Designer, 580
- Navier-Stokes solutions for modeling gases, 204
- nearest-neighbor filter, seam artifacts and, 420
- nebulae, MojoWorld for modeling, 611, 612
- Ng variable, RenderMan vs. real-time hardware rendering and, 106
- noise function (RenderMan), 83–84
- noise functions, 67–83. *See also* cellular texturing; Perlin's noise function
 - for 2D cloud textures, 267–268
 - 2D slices of, 74
 - 4D functions, 84, 181
 - artifacts from, 180–181
 - cellular texturing compared to, 135–136
 - in cloud modeling, 270, 298–299
 - color splines and, 189
 - distorted (DistNoise), 450–453
 - enhancements and modifications, 179–182
 - explicit noise algorithms, 82–83
 - flame shader based on, 124–129
 - flow noise, 384–387
 - Fourier spectral synthesis, 82–83
 - for gases, 211–214
 - generating, 67–68
 - gradient noise, 72–77
 - graphs of, 73
 - hardware acceleration for, 291
 - Hermite spline interpolation in, 490, 498
 - lacunarity, 89, 180
 - lattice convolution noise, 78–80
 - lattice noises, 69–70
 - in layering facial movement, 405–408

- noise functions (*continued*)
 - lookup table for, 181
 - for marble texture, 87–89
 - multipass pixel shader implementation, 422–425
 - need for, 67
 - noise value range and distribution for, 85
 - normalizing noise levels, 189–190
 - performance issues, 194–195
 - for perturbed image textures, 90–91
 - for perturbed regular patterns, 89–90
 - with PGI, 327
 - as pink noise, 68
 - power spectra of, 75
 - properties of ideal function, 68
 - random placement patterns using, 91–94
 - Rayshade implementation, 422–425
 - in real-time procedural solid texturing, 421–425
 - RenderMan functions, 83–85
 - rotation matrices for hiding artifacts, 180, 181
 - for smoke and fog, 297, 298
 - sparse convolution noise, 80–82
 - spectral synthesis with, 85–89
 - as surflet, 379
 - value noise, 70–72
 - value-gradient noise, 77–78
 - vector-valued (*VecNoise*), 450–451
 - versatility of, 135
 - for volumetric cloud modeling, 270
 - Ward's Hermite noise function, 73, 74, 75
 - white noise and, 67–68
- noise textures, development of, 11
- nonprocedural textures, procedural vs., 12
- normal vectors, transforming between texture spaces, 46
- normalizing noise levels, 189–190
- n*-space model, 548–549, 559
- n*transform function (*RenderMan*), transform function vs., 46
- NVIDIA's Cg language, 111, 292–296
- Nyquist frequency
 - aliasing and, 53–54
 - clamping and, 60
 - defined, 53
 - MojoWorld function fractals and, 602–603
 - PhotoRealistic *RenderMan* antialiasing scheme and, 56
 - QAEB tracing and, 511, 512
 - white noise and, 68
- NYU torch hypertexture example, 362, 364
- object space
 - described, 24
 - screen space vs., for shading, 103–106
 - transformation to world space with PGI, 323–324
- Occam's Razor, 443, 530
- octaves in fractals, 433–434, 444, 583, 584–585
- offline programmable shading, real-time programmable shading vs., 98–100
- ontogenetic modeling
 - defined, 442
 - fractals and, 442–444
 - of hurricane, 456–458
 - Occam's Razor and, 443
 - realism and, 444
 - semblance in, 443
 - teleological modeling vs., 442
- opacity
 - plastic shader output, 21
 - solid textured transparency, 233–235
 - in volume-rendering algorithm for gases, 206
- OpenGL
 - Direct3D vs., 102
 - lighting model, 118
 - pixel texture extension, 422
 - for real-time cloud rendering, 267
- OpenGL Programming Guide*, 102
- OpenGL Shader project, 131
- operands in GPUs, precision of, 288
- optical depth, 531
- optical paths, 531, 534
- optimizing. *See also* performance
 - hypertexture smoke, 367–368
 - lattice noises, 214
 - spot size, 173–175
- outscattering
 - defined, 531
 - Rayleigh, 536
- oversampling. *See* supersampling
- parallelism
 - CPUs vs. GPUs and, 102
 - GPU restrictions due to, 103, 106–108
- parameter passing in PGI, 321–322

- parameter proliferation problem, 547–548
- parametric control, 2, 14
- Parametric Hyperspace, 589–590
- parametric patches, 8
- particle systems
 - animation using, 257–261
 - attributes of particles, 257
 - for cloud modeling, 269, 282–283
 - initial shape, 257–258
 - L-system for, 313–314
 - movement of particles, 258
 - overview, 3
 - particle creation procedure, 257
 - probabilistic rendering, 259, 260
 - processes for each time step, 257
 - rendering problems, 259, 261
 - structured, 257, 258–259, 260
 - uses for, 209, 257, 259
 - for wall of fire, 257, 258
- pattern generation
 - advantages of procedural generation, 23
 - brick texture example, 39–41
 - bump-mapped brick example, 41–46
 - checkerboard pattern, 39
 - defined, 20
 - functional composition technique, 26
 - hiding regularity and periodicity, 51
 - irregular patterns, 83–94
 - layering technique, 25–26
 - perturbed image textures, 90–91
 - perturbed regular patterns, 89–90
 - photographic texture images for, 22–23
 - primitive functions as building blocks, 27–51
 - random placement patterns, 91–94
 - spectral synthesis technique, 48–51, 85–89
 - star texture example, 46–48, 49
 - writing procedural generators, 23–24
- pelting, described, 10–11
- penumbra, self-shadowing with, 382–383
- percentage closer filtering, described, 9
- performance. *See also* hardware acceleration of procedural techniques; optimizing
 - antialiasing efficiency issues, 157
 - assembly language vs. high-level language for shading and, 101
 - in cellular texturing, speed vs. isotropy, 146–147
 - as critical for real-time shading, 98, 99–100
 - early vs. late binding and, 117–118
 - hardware improvements and, 97
 - height field speedup scheme, 513, 516–517
 - memory bandwidth and performance tuning, 110
 - procedural texture efficiency issues, 194–195
 - of QAEB tracing, 510, 511, 518, 520
 - of ray-traced self-shadowing, 207–208
 - real-time procedural solid texturing issues, 425–426
 - target hardware and, 130
 - texture size and, 110
- periodic functions (RenderMan)
 - ceil, 33
 - cos, 31, 32
 - floor, 32–33
 - lacunarity and, 180
 - making other functions periodic, 32
 - mod, 31–32
 - sin, 31, 32
- Perlin basis function (MojoWorld), 590–591, 592, 593, 594
- Perlin's noise function
 - construction of, 340–347
 - DistNoise function based on, 450
 - finding current cubical “cel” location, 341
 - finding the pseudorandom wavelet at each cel vertex, 341–342
 - folding function, 343
 - as fractal basis function, 432–433
 - gradient distribution, 348
 - gradient table, 342–343
 - interpolation polynomial improvement, 347
 - in MojoWorld basis function, 590–591, 592, 594
 - nonbiased index of G , 343–344
 - overview, 340–341
 - performance improvements, 348
 - point evaluation in, 489–490
 - procedural fBm and, 437
 - pseudorandom permutation table, 343
 - in QAEB-traced hypertextures, 521
 - in real-time procedural solid texturing, 421–423

- Perlin's noise function
 - (*continued*)
 - reducing grid-oriented artifacts, 348
 - as seminal function, 69, 73
 - steps in computation, 341
 - uniformly distributed unit gradients generated by, 75–76
 - wavelet coefficients, 342–344
 - wavelet evaluation, 344–347
 - wavelet properties, 342
- perm array in lattice noise generation, 69–70
- perturbance
 - perturbed image textures, 90–91
 - perturbed regular patterns, 89–90
 - texture for, 197
 - volume perturbation for cloud modeling, 299–301
- PGI. *See* procedural geometric instancing (PGI)
- phenotypes, 551, 554
- photographic texture images, 22–23
- PhotoRealistic RenderMan
 - antialiasing in, 55–56
 - REYES algorithm, 103
 - sampling interval changes in, 59
 - screen space use by, 103
- pink noise, 68
- pipelines
 - computational precision issues, 288
 - levels of operation in, 289
 - REYES (PRMan) vs. real-time graphics hardware, 103–104
- PixelFlow project, 131
- planetary atmosphere. *See* atmospheric models
- planetary rings, fractal solid textures for, 485–487
- planetary rings antialiasing example, 163–166
- planetclouds procedure, 453–455
- plant modeling. *See also* MojoWorld
 - bounding volumes for, 331–332
 - bushes, 318–320
 - crop circles, 326
 - grass, 322–323, 326–329, 331
 - L-system for, 308–310, 311, 313, 318–320
 - meadows, 327–329
 - PGI for, 322–323, 324–329, 331–332
 - trees, 309–310, 324, 325, 331–332
 - tropism, 324–326
- plastic shader, 20–22
- plateau width parameter for bump mapping ridge, 184
- pnoise function (RenderMan), 84–85
- point evaluation
 - context-independence of, 490
 - Fourier synthesis vs., 490
 - in Perlin noise-based procedural fractal construction, 489–490
- polygons
 - polyhedron subdivision vs., 490
 - rounded terrain capabilities of, 490–491
- Poisson distribution in cellular texturing, 145–147
- polyhedron subdivision
 - basis function for, 497
 - for mountain modeling, 489
- point evaluation vs., 490
- terrain creases as artifacts, 498
- polygons
 - cloud modeling using, 268
 - L-system modeling of, 311
- object space vs. screen space shading and, 104–105
- post spacing of height fields, 509
- power functions
 - gas shape and, 215–216
 - for tuning procedural textures, 339
- power spectra
 - of fractional Brownian motion (fBm), 433
 - of noise functions, 75
- precomputation for procedural techniques, 289–290
- previewer for textures, 194
- Primitive Itch's ShaderLab2 package, 130
- PRMan. *See* PhotoRealistic RenderMan
- PRNs (pseudorandom numbers)
 - in cellular texturing, 143
 - for lattice noises, 69
 - table for noise functions, 69
 - for value noise, 70–71
 - for white noise, 67–68
- Pro UI of MojoWorld, 566–567, 606
- procedural, defined, 12
- procedural city technique, 200
- procedural cloud animation, 279–280
- procedural evaluation phase for real-time procedural solid texturing, 414, 425
- procedural fBm, 436–438, 440
- procedural fractal terrains, 489–506. *See also* earth textures; MojoWorld
 - advantages of point evaluation, 489–491
 - basis function effects, 497–499
 - fractal dimension, 495–497
 - height fields, 491–495

- heterogeneous terrain models, 498–506
- homogenous fBm terrain models, 495–498
- hybrid multifractal terrain, 502–505
- multiplicative multifractal terrains, 505–506
- rounded terrain capabilities, 490–491, 498
- statistics by altitude, 500–502
- procedural geometric instancing (PGI), 321–330
 - accessing world coordinates, 323–324
 - bounding volumes, 330–332
 - crop circles example, 326
 - described, 321
 - fractional Brownian motion, 327–329
 - front-to-back ordering, 330
 - geometry mapping example, 326–327
 - inductive instancing example, 322–323
 - levels of detail, 329
 - L-systems compared to, 329–330
 - meadows example, 327–329
 - noise function with, 327
 - parameter passing, 321–322
 - random number function, 327
 - trees example, 331–332
 - tropism example, 324–326
- procedural modeling of gases. *See* volumetric procedural modeling and texturing
- procedural shape synthesis, 387, 390
- procedural synthesis of geometry, 305–334
 - applications, 305
 - bounding volumes, 330–332
 - data amplification algorithms, 305, 312–314
 - flow of data, 312
 - future work, 333–334
 - lazy evaluation algorithms, 305, 314–315
 - L-system, 305–306, 307–312
 - overview, 332–333
 - paradigms governing, 312–315
 - procedural geometric instancing (PGI), 321–330
 - scene graphs, 307, 315–321
 - Web and, 333
- procedural techniques. *See also specific techniques*
 - defined, 1
 - implicit vs. explicit, 12–14
 - migrating from CPUs to GPUs, 287–289
 - overview, 1–2
 - precomputation for, 289–290
- procedural textures (overview). *See also* texture design methods; volumetric procedural modeling and texturing
 - advantages of, 14
 - aliasing in, 55–56
 - antialiasing methods, 56–67
 - antialiased rendering of, 369–376
 - brick texture example, 39–41
 - bump-mapped brick example, 41–46
 - checkerboard pattern, 39
 - defining characteristics, 12
 - disadvantages of, 14–15
 - efficiency issues, 194–195
 - explicit vs. implicit methods, 12–14
 - functional composition technique, 26
 - hiding regularity and periodicity, 51
 - historical overview, 11–12
 - image texture vs., 14–15
 - irregular, 67–94
 - isocurve or isosurface method, 13
 - layering technique, 25–26
 - multifractal models, 442
 - nonprocedural vs., 12
 - pattern generation, 20, 22–24, 25–26
 - primitive functions as building blocks, 27–51
 - renderer antialiasing schemes and, 55
 - shading models, 20–22
 - spectral synthesis technique, 48–51
 - star texture example, 46–48, 49
 - texture spaces, 24–25
 - volume rendering with surface textures, 195–196
- productions in L-system, 307, 308
- pseudoadvection in flow noise function, 386–387
- pseudorandom numbers. *See* PRNs (pseudorandom numbers)
- psychedelic clouds, 525, 526
- puffycLOUDS procedure, 448–449
- pyroclastic flow, 523–524
- QAEb rendering for procedural models, 509–526
 - antialiasing, 516
 - C code implementation, 625–626

- QAEB rendering for procedural models (*continued*)
 - clouds, 520–525, 526
 - error in the algorithm, 513–514
 - fireballs, 525
 - implicit models and, 510
 - intersection point calculation, 515
 - meaning and pronunciation of acronym, 510, 511
 - near and far clipping planes, 514–515
 - Nyquist limit and, 511, 512
 - performance, 510, 511, 518, 520
 - prior art, 512–513
 - problem statement, 511–512
 - pyroclastic flow, 523–524
 - QAEB algorithm, 513–514
 - QAEB tracing overview, 510–511
 - QAEB-traced
 - hypertextures, 520–523
 - reflection and refraction, 517–518
 - shadows, 517
 - speedup scheme for height fields, 513, 516–517
 - stride length, 521–522
 - surface normal construction, 516
- quad tree spatial subdivision, height fields and, 494
- Quake III game engine, 131
- quasi-analytic error-bounded ray tracing. *See* QAEB rendering for procedural models
- radar texture, 197
- radial coordinate distance metric in cellular texturing, 148
- radiosity as global shading model, 8
- random coloring methods, 477–485
 - fBm coloring, 477–478
 - GIT texturing system, 478–480
 - “multicolor” texture, 482–485
- random fractals
 - basis function, 582, 583
 - constructing, 582–586
 - expressive vocabulary of, 611, 613
 - fractal dimension, 582–584
 - lacunarity of, 585–586, 593
 - octaves, 583, 584–585
 - overview, 574–575
- random placement patterns
 - defined, 91
 - noise function for, 91–94
 - one-star-per-cell version, 92–93
 - storing bomb positions in table, 91
 - version improving clipping and randomness, 93–94
- randomness. *See also* noise functions; PRNs (pseudorandom numbers)
 - deterministic nature of
 - computers and, 574
 - MojoWorld seed tables for, 597–599
 - PGI functions, 327
 - in setting texture parameters, 193
 - true vs.
 - pseudorandomness, 67, 574–575
- range function dimensions in MojoWorld, 587
- rasterization phase for real-time procedural solid texturing, 414, 416, 419, 425–426
- ray tracing. *See also* QAEB rendering for procedural models
 - antialiasing by stochastic ray tracer, 55
 - as global shading model, 8
 - height fields and, 494
 - implicit geometric models in, 13
 - implicit vs. explicit procedures in, 13
 - raymarcher renderer, 348–352
 - ray-surface intersection point calculation, 515
 - reflection mapping as, 9
 - scanline rendering vs., 511
 - for self-shadowing of
 - gases, 207–208
 - in surflets, 379–380, 382–383
 - for volume rendering of
 - gases, 205–206
- Rayleigh scattering
 - aerial perspective and, 530
 - color perspective due to, 529
 - in *Fractal Mandala*, 533
 - minimal approximation, 536, 539
- raymarcher renderer, 348–352
 - overview, 348–349
 - system code, 349–350
 - user-defined functions, 350–352
- Rayshade implementation of noise function, 422–425
- reaction-diffusion textures, intelligent, 200
- realism
 - complexity and, 434–435
 - heterogeneous terrains for, 498, 500
 - multifractal functions and, 438, 440
 - “naturalness” of fractals, 485

- ontogenetic modeling and, 444
- real-time clouds, 298–301
- real-time procedural solid texturing, 413–427
 - algorithm, 413–416
 - applications, 425–426
 - area-weighted mesh atlas, 417–418
 - atlas based on clusters of proximate triangles, 418, 419
 - atlas construction for, 416–419, 425
 - bilinear filtering, 420–421
 - hardware acceleration, 416, 421–425
 - implementing, 421–425
 - length-weighted atlas, 418
 - multipass pixel shader implementation of noise function, 422–425
 - nearest-neighbor filter, 420
 - noise functions in, 421–425
 - parameterization, 413
 - performance issues, 425–426
 - procedural evaluation phase, 414, 425
 - rasterization phase, 414, 416, 419, 425–426
 - scaling component of distortion, 416
 - seam artifacts, avoiding, 419–421
 - solid texture coordinates, 413–414
 - spatial coordinates, 413
 - texture filtering, 419–421
 - texture mapping phase, 415–416
 - uniform meshed atlas, 416–417
 - view independence, 425
- real-time programmable shading, 97–132
 - flame shader, noise-based procedural, 124–129
 - future GPU hardware and programming languages, 130–131
 - GPU architecture, 102–103
 - hardware data types, 108–109
 - Heidrich/Banks
 - anisotropic shader, 112–116
 - high-level shading language advantages, 100–101
 - interactivity and, 98, 99, 100
 - interface between shaders and applications, 118–121
 - iterative nature of development process, 99, 130
 - knowledge required for the reader, 101–102
 - literature review, 131
 - memory bandwidth and performance tuning, 110
 - object space shading vs. screen space shading, 103–106
 - offline programmable shading vs., 98–100
 - parallelism, 106–108
 - performance as critical issue for, 98, 99
 - resource limits, 109
 - simple examples, 111–116
 - Stanford shading language example, 111–112
 - strategies for developing shaders, 129–130
 - surface and light shaders, 116–118
 - volume-rendering shader, 121–124
- rectangular pulse, step function for, 28, 29
- recurrent iterated function systems, 320
- reflection, QAEB tracing and, 517–518
- reflection mapping, 9
- reflection models. *See* shading models
- refraction, QAEB tracing and, 517–518
- renderers
 - antialiasing schemes and procedural textures, 55
 - flow of data and, 312
 - image display while rendering, 194
 - raymarcher, 348–352
 - spot size calculation by, 166–167
- rendering
 - antialiased rendering of procedural textures, 369–376
 - efficiency issues, 194–195
 - image display during, 194
 - particle systems, 259, 261
 - previews, 194
 - texture as 2D image, 176
 - volume, 121–124, 195–196
 - volumetric rendering system, 205–208
- RenderMan
 - development of, 1
 - PhotoRealistic, 55–56, 59, 103
- RenderMan Companion, The*, 102
- RenderMan shading language. *See also specific functions*
 - brick texture example, 39–41
 - bump-mapped brick texture example, 43–46
 - cloud simulation using spectral synthesis, 50–51
 - conditional functions, 27–31
 - du and dv variables, 58
 - ease of, 447
 - examples using, 16–19

- RenderMan shading language (*continued*)
 - further information, 15, 102
 - multiplying colors in, 26
 - noise functions, 83–85
 - OpenGL lighting model and, 118
 - overview, 15–16, 19
 - periodic functions, 31–33
 - plastic shader in, 20–22
 - real-time shading and, 101–102
 - star texture example, 47–48
 - surface and light properties separated in, 117
 - texture spaces, 24–25
 - translating into C code, 19
- reptile hide texture, 139
- repulsors, 246
- resolution
 - image textures vs. procedural textures and, 14
 - low-resolution previews, 194
 - surflet hierarchy, 381–382
- resource limits in GPUs, 109
- REYES algorithm (PRMan), 103–106
- RGB color. *See also* color
 - multiplying colors together, 26
 - random fBm coloring, 477–478
 - in RenderMan shading language, 118
 - transforming to HSV using a texture, 196
- RGBA color
 - layering texture patterns and, 26
 - in OpenGL lighting model, 118
 - in Stanford shading language, 118
- ridge for bump mapping, 183–187
 - algorithm, 187
 - bevel shape parameters, 184–186
 - Hermite blending functions, 186
 - non-geometric applications, 187
 - plateau width parameter, 184
 - ridge shapes for different slope controls, 185
 - ridge width parameter, 183–184
 - simple ridge, 183, 184
- ridged basis function (MojoWorld), 590–591, 593, 594
- RidgedMultifractal procedure, 504–505
- Rings procedure, 485, 487
- ripples procedure, 462–463
- rising_smoke_stream procedure, 239–243
- rotation along helical path, 228
- rotation matrices, noise artifacts and, 180, 181
- roughness. *See* fractal dimension
- RTSL. *See* Stanford shading language
- sample point, shading, 15
- sampling
 - defined, 52
 - Monte Carlo method, 206
 - stochastic, 55–56, 67
 - supersampling, 54
- sampling interval
 - clamping and, 60
 - defined, 57–58
 - Du and Dv functions for de-termining, 58–59
 - du and dv variables for de-termining, 58
 - variations in
 - PhotoRealistic RenderMan, 59
- sampling rate
 - aliasing and, 53, 54
 - clamping and changes in, 60
 - defined, 52
 - for low-pass filtering, de-termining, 57–59
 - Nyquist frequency and, 53
 - as sampling interval reciprocal, 57–58
 - supersampling or oversampling, 54
- sampling theorem, 52
- Saturn’s rings, 485–487
- scanline rendering, ray tracing vs., 511
- scattering, 531
- scattering models
 - aerial perspective and, 530
 - color perspective and, 529
 - Rayleigh scattering, 529, 530, 533, 536, 539
- scene description modules (SDMs), 393–394
- scene graphs, 315–321. *See also* procedural geometric instancing (PGI)
 - benefits of, 307
 - cyclic, 320, 321
 - defined, 315
 - instances, 315
 - iterated function systems, 320
 - libraries and languages for, 315
 - limitations of, 307, 321
 - L-system with additional iteration, 318–320
 - named node syntax, 315–316
 - overview, 307
 - PGI augmentations, 321–330
 - single-production L-system implementation, 317–318
 - terminology, 315
 - tree-structured, 320–321
- science
 - computer graphics vs., 448
 - iterative method in, 447–448

- Science of Fractal Images, The*, 429, 433, 438, 489, 575, 579
- snoise function. *See* sparse convolution noise
- screen space
 mapping for 3D tables for
 controlling animation, 245
 moving point through
 solid space, 227, 228, 232–233, 235–243
 object space shading vs.,
 for shading, 103–106
 relating texture space to,
 168
- SDMs (scene description modules), 393–394
- sea surface texture, 140, 141
- sedimentary rock strata,
 466–467, 468
- seed tables for MojoWorld,
 597–599
- Selene (Moon) texture, 473–477
 highlands/maria, 473
 luna procedure, 474–476
 rayed crater, 473–474
- self-shadowing. *See also*
 shadowing
 of clouds, 267
 of gases, 207–208
 of particle systems, 259
 with penumbra, 382–383
 surflets for, 376, 382–383
- self-similarity
 defined, 572
 in fractal geometry, 621
 fractal terminology and,
 496–497
 in fractals and nature,
 572–573
 statistical vs. exact, 435, 575
- semblance
 in ontogenetic modeling,
 443
 veracity vs., 448
- serendipity
 image textures vs. procedural textures and, 15
- procedural techniques and,
 2
- sexual reproduction, 551, 554
- “shade trees” system, 11
- shader space (RenderMan)
 described, 24
 for solid textures, 25
- ShaderLab2 package (Primitive
 Itch), 130
- shading, 7
- shading models. *See also specific types*
 anisotropic, 8
 for clouds, 266–267
 defined, 20
 diffuse, 7–8
 for gaseous phenomena,
 205, 207
 historical overview, 7–8
 local vs. global, 8
 low-albedo vs. high-albedo, 205
 object space vs. screen
 space, 103–106
 plastic shader, 20–22
 for procedural textures,
 20–22
 separation of surface and
 light properties, 117
 simplifying assumptions in,
 7
 specular reflection, 8
- shading sample, 15
- shading sample point, 15
- shadowing. *See also* self-shadowing
 cloud shadowing, 267
 in gases, 207–208
 hardware acceleration for,
 208
 in particle systems, 259
 in QAEB tracing, 517
 self-shadowing, 207–208,
 259, 267, 376
 volumetric, 207–208
- shape
 Boolean characteristic
 function for, 337–338
 continuous function for,
 338
- procedural shape synthesis,
 387, 390
- sign function (RenderMan),
 step function vs., 48
- signal processing
 aliasing in, 53–54
 Fourier analysis in, 52–53
 further information, 52
 Nyquist frequency, 53–54
 sampling theorem, 52
- SIMD (single instruction, multiple data) computation model, in GPUs,
 107–108
- simple_light shader, 115
- sine function
 as band-limited, 57
 gas shape and, 215, 216
 graph of, 32
 overview, 31
 for rotation on helical
 path, 228
- sine wave basis function
 (MojoWorld), 596
- sinf function (ANSI C implementation),
 RenderMan sin function vs., 31
- single program, multiple data
 (SPMD) computation
 model in GPUs, 108
- sintegral macro, 65
- size (area)
 fractal complexity and,
 431
 image textures vs. procedural textures and,
 14
 photographic texture image problems, 23
 spot size, 166–170
- size (storage)
 of image textures vs. procedural textures, 14
 texture size and performance, 110
- smoke. *See also* fog; gases; volumetric procedural modeling and texturing

- smoke (*continued*)
 - column, animated, 239–243
 - column, hypertexture examples, 364–368
 - column, still image, 219–224
 - drifting, 365, 366
 - hardware acceleration for space-filling smoke, 297–298
 - optimizing, 367–368
 - rings, 365, 366–367
- Smoke function, 297–298
- smoke_density_function for hypertextures, 367–368
- smoke_stream procedure, 220–224
 - described, 220
 - helical path for smoke, 220, 222–224
 - parameters, 224
 - rising_smoke_stream procedure and, 243
- smoothstep function (RenderMan)
 - aliasing and, 57
 - analytic prefiltering with, 62
 - C implementation, 30
 - graph of, 31
 - overview, 30–31
 - step function vs., 31
- snoise function
 - marble texture using, 87–89
 - perturbed image textures using, 90–91
 - perturbed regular patterns using, 89–90
- soft objects, 3
- solid spaces. *See also* animating solid spaces; volumetric procedural modeling and texturing
 - changing over time, 227, 229–232
 - development of framework, 209–210
 - mathematical description of framework, 210–211
 - moving screen space point through, 227, 228, 232–233, 235–243
 - overview, 210
 - three-dimensional tables for controlling animation, 244–254
 - uses for framework, 210
- solid textures. *See also* animating solid spaces; fractal solid textures; real-time procedural solid texturing
 - for clouds in Maya animation system, 267
 - color texture, 196
 - defined, 337
 - described, 209
 - fractal, 447–487
 - fuzzy shapes combined with, 338
 - as hypertexture precursor, 337–338
 - overview, 10
 - real-time procedural solid texturing, 413–427
 - solid spaces framework for modeling gases, 209–211
 - solid textured transparency, 233–235
 - as space-filling functions, 337–338
 - transparency, 233–235
 - wood-grain example, 10
- space-filling fractals, hypertexture example, 357, 358–359
- spaces. *See* coordinate systems; texture spaces; *specific spaces*
- sparse convolution basis function (MojoWorld), 593, 595
- sparse convolution noise
 - 2D slice of, 74, 82
 - as fractal basis function, 450, 497
 - graph of, 73
 - implementation of, 80–82
 - power spectrum of, 75, 82
 - spatial frequency of fractal basis function, 433
- spectral synthesis
 - clamping method for antialiasing, 56, 59–61, 86–87
 - cloud simulation, 50–51
 - inverse Fourier transform for, 49
 - irregular pattern generation using, 85–89
 - marble synthesis using, 87–89
 - for modeling gases, 204
 - noise generation using, 82–83
 - overview, 48–51
 - turbulence function, 85–87
- specular function (RenderMan), 21
- specular reflection models, 8
- specularcolor parameter of plastic shader, 21, 22
- sphere
 - for hypertexture creation, 353–354
 - for hypertexture explosion example, 354
 - implicit formulation of, 270
- spherical attractors, 247–249
 - animating center of attraction, 249
 - breeze effect using, 247, 248, 252–253, 254
 - effect of increasing over time, 247
 - effect over time, 247
 - extensions of, 249
 - flow field function, 248–249
 - geometry of attraction, 249

- SPHIGS, 322
- spiral paths. *See* helical paths
- spiral vortex functions,
 - 249–251
 - based on 2D polar coordinate function, 250
 - based on conservation of angular momentum, 251
 - based on frame number and distance from center, 250–251
 - example of effects, 252
 - vortex simulation vs., 249
- spline functions
 - C implementation, 34–35
 - as color map or color table, 36
 - for colors or points, 35–36
 - graph of, 35
 - knot values, 34
 - overview, 34–37
 - reflection texture example, 37
 - for tuning procedural textures, 339
 - for value noise interpolation, 71–72
- SPMD (single program, multiple data) computation model in GPUs, 108
- spot noise, 80
- spot size
 - calculating, 167–170
 - “correct” bump mapping using, 171–173
 - correcting renderer calculations, 167
 - defined, 166
 - exaggerating to reduce aliasing, 175
 - optimization and verification, 173–175
 - renderer calculation of, 166–167
 - stretched spots and, 169–170
- square waves as fractal basis function, 497
- Stanford shading language
 - API calls for one face of a cube, 119–120
 - API routines, 118–119
 - as current sole choice for GPUs, 288
 - development of, 131
 - early-binding model in, 118
 - Heidrich/Banks
 - anisotropic shader, 112–116
 - noise-based procedural flame shader, 124–129
 - OpenGL API and, 118
 - OpenGL lighting model and, 118
 - separation of surface and light properties in, 118
 - vertex/fragment programming model, 111–112
 - volume-rendering shader, 121–124
- star shader, 47–48
- star texture
 - feature space for, 46
 - geometry of star, 46–47
 - image rendered using, 49
 - procedural texture generator, 46–48
 - random placement pattern using, 92–94
 - testing whether point is inside star, 48
- Star Trek II: The Wrath of Khan*, 257, 258, 578
- static bounding volumes, 331
- stationary property of noise function, 68
- statistical self-similarity, exact self-similarity vs., 435, 575
- steam. *See also* fog; gases; volumetric procedural modeling and texturing
 - breeze effect using attractor, 247, 248, 252–253, 254
 - rising from teacup, animated, 236–238
 - rising from teacup, still image, 216–219, 220
 - steam_moving procedure
 - breeze effect using attractor, 247, 248, 252–253, 254
 - for steam rising from teacup, 236–238
 - steam_slab procedure
 - basic slab of steam, 216
 - confining steam within cup radius, 216–218
 - ramping off gas density for rising steam, 218–219
 - steam_moving procedure and, 238
 - variables, 219
- step function (RenderMan)
 - antialiasing and, 28, 61–62
 - box-filtering, 61–62, 64–65
 - C implementation, 27
 - graph of, 27
 - high frequencies generated by, 54, 57
 - overview, 27
 - for rectangular pulse, 28, 29
 - rewriting if functions using, 27–28
 - sign function vs., 48
 - smoothstep function vs., 31
- steps basis function (MojoWorld), 596, 597
- stochastic control of gesture, 387, 392–393
- stochastic models vs. fractal models, 495–497
- stochastic procedures
 - for particle creation, 257–258

- stochastic procedures
 (*continued*)
 for particle movement, 258
 in structured particle systems, 258–259
- stochastic sampling
 for antialiasing, 55–56, 67
 defined, 55
 by renderers, 55–56
- storage
 GPU issues, 103, 289
 height fields for terrains, 491, 493–495
 surflets for storing
 hypertextures, 376–384
- store-to-memory instruction,
 GPU lack of, 103
- strata procedure, 466–467
- stratus cloud models, 276,
 277–278, 282, 283
- stream processors, GPUs as, 110
- structured particle systems, 257,
 258–259
- subsumption architecture, 397
- sum table, antialiasing using,
 162–163
- summed-area table method of
 antialiasing, 63–64
- supersampling
 as antialiasing strategy, 54,
 66–67, 157
 in procedural textures,
 66–67
 stochastic, 55, 67
 by texture instead of ren-
 derer, 170–173
- surface normal construction for
 QAEB tracing, 516
- surface properties
 defined, 116
 GPUs and, 117
 separating from light prop-
 erties, 116–117
- surface shaders vs. light shaders,
 117
- surface textures
 MojoWorld, 589, 608–609
- procedural shape synthesis,
 387, 390
 volume rendering with,
 195–196
- surflets, 376–384
 advantages over surface-
 based techniques,
 383–384
 defined, 376, 378
 finding visible surfaces,
 379–380
 generator for, 381
 hierarchical model,
 381–382
 noise function as, 379
 ray tracing in, 379–380,
 382–383
 selective surface re-
 finement, 380–381
 self-shadow with penum-
 bra example,
 382–383
 shading by, 380
 singularities and, 381
 steps for generating, 379
 surflet model, 377–378
 uses for, 376–377, 384
 wavelet integration with,
 384
 as wavelets, 378–379
- symmetry, dilation, 431,
 573–574
- “Synthesis and Rendering of
 Eroded Fractal Ter-
 rains, The,”
 575–576
- synthetic texture models, 11
- talking in different moods,
 408–409. *See also*
 texture for facial
 movement
- teleological modeling, 442
- temporal antialiasing, verifying,
 175
- Terragen software, 580
- terrains. *See* atmospheric mod-
 els; earth textures;
 MojoWorld; proce-
 dural fractal terrains
- terran procedure, 470–472
- textural limb animation, 387,
 391–395
 basic notions, 392
 examples, 394, 395
 goal determination module
 (GDM), 394–395
 limb motion, 391
 overview, 387
 related work, 392
 scene description modules
 (SDMs), 393–394
 stochastic control of ges-
 ture, 387, 392–393
 system for, 393–395
 textural gesture defined,
 393
 uses for, 391
- texture. *See also* procedural tex-
 tures (overview)
 historical overview, 8–11
 L-system specification,
 311–312
- texture baking, 197–199
- texture design methods,
 179–201. *See also*
 user interface for
 textures
- 2D mapping methods,
 197–199
- bump-mapping methods,
 183–187
- color mappings, 181–182
- efficiency issues, 194–195
- future of, 199–202
- hiding noise artifacts,
 179–182
- toolbox functions,
 179–187
- user interface, 187–194
- utility textures, 196–197
- volume rendering with sur-
 face textures,
 195–196
- texture filtering for real-time
 procedural solid tex-
 turing, 419–421

- texture for facial movement, 395–409
 - addressing an audience, 407
 - background, 396
 - basic component movements, 398–399
 - blinking, 405
 - bottom-level movement vocabulary, 402–403
 - example component movement combinations, 399, 400
 - future work, 409
 - laughing, 407–408
 - mixing coherent jitter in component movements, 403, 405
 - movement layering, 401
 - movement model, 398–401
 - noise in layering facial movement, 405–408
 - opposites of component movements, 399–400
 - painting with actions, 403–405
 - problems addressed, 395
 - related work, 397
 - searching, 407
 - small constant head movements, 405–406
 - talking in different moods, 408–409
 - Web site, 408
 - winking, 408
- texture function (RenderMan)
 - built-in filtering for, 57
 - described, 22
- texture images, photographic, 22–23
- texture mapping
 - 2D methods, 197–199
 - 2D techniques, limitations of, 10
 - atlas for real-time procedural solid texturing, 416–419
 - for clouds, 267
 - evolution of, 10–11
 - GPU support lacking for vertices, 105
 - hardware acceleration for smoke and fog, 297–298
 - photographic texture image problems, 23
 - in real-time procedural solid texturing, 415–416
 - shader development and, 130
 - simple point-sampling approach, 8
 - texture baking, 197–199
 - UV mapping, 197–199
- texture spaces. *See also specific spaces*
 - built-in in RenderMan, 24
 - bump mapping and, 45–46
 - changing world space to, 168
 - choosing when defining textures, 24–25
 - feature spaces, 25
 - relating to screen space, 168
 - transforming normal vectors between, 46
 - user-defined in RenderMan, 24
- texturing, 7
- three-dimensional tables for controlling animation, 244–254
 - accessing table entries, 245
 - breeze effect using attractor, 247, 248, 252–253, 254
 - combinations of functions, 251–254
 - common dimensions for, 245
 - flow into hole in wall, 253–254, 255
 - functional flow field functions, 246–251
 - functional flow field tables, 245, 246
 - nonuniform spacing between entries, 244–245
 - overview, 244–245
 - vector field tables, 245
 - wind effects, 252–254
- time vs. space trade-off, 15
- time-animated fractal noise, 181
- time-dependent textures, 4D
 - noise function for, 84
- tintegral macro, 65
- tinting, solid color texture for, 196
- torch hypertexture example, 362, 364
- transform function
 - (RenderMan), ntransform function vs., 46
- transparency, solid textured, 233–235
- trees
 - bounding volumes for, 331–332
 - dilation symmetry in, 573
 - fractal modeling challenges, 506
 - L-system for modeling, 309–310
 - PGI for modeling, 324, 325, 331–332
- tree-structured scene graphs, 320–321
- tropism
 - defined, 324
 - PGI for modeling, 324–326
- truth, known vs. proven, 442
- turbulence
 - fBm-valued distortion to emulate, 454–456
 - fractal character of, 435–436
 - ontogenetic model of hurricane, 456–458
 - in QAEB-traced hypertextures, 521
 - viscous dumping and, 606

- turbulence functions
 - for air current simulation, 238
 - clamping version, 86–87
 - in cloud modeling, 270, 271, 281
 - derivative discontinuities in, 86
 - fabs function with, 369
 - in flame shader, 129
 - fractal sum function vs., 86
 - for gases, 211, 214
 - hardware acceleration for, 291–292
 - for marble formation from banded rock, 230–232
 - overview, 368–369
 - with particle systems, 261
 - power spectrum of, 85, 86, 87
 - for rolling fog animation, 238
 - sine function effects on, 215, 216
 - for smoke and fog, 297, 298
 - for smoke stream modeling, 220, 223
 - for transparency control, 234
 - for volumetric cloud modeling, 270
- turtle graphics symbols in L-system
 - for bicubic patch modeling, 311
 - for color and texture specification, 311–312
 - described, 305
 - difficulties for human processing, 306
 - geometric meanings of, 308
 - labels for debugging or annotation, 312
 - multicharacter symbols, 311
- operating system command execution, 312
- parameterized, 310–311
- for plant modeling, 308–310, 311
- for polygon modeling, 311
- query command, 326
- for von Koch snowflake modeling, 310–311
- type promotion in Stanford shading language, 112
- uniform meshed atlas, 416–417
- unit interval, remapping, 37–39
- upper crossover scale
 - of fractals, 434
 - in MojoWorld, 607
- user interface for cloud modeling, 278–279
- user interface for textures, 187–194
 - adding parameter range suggestions, 188
 - color table equalization, 189, 190–192
 - difficulties in manipulating parameters, 192–193
 - importance of, 187–188
 - library of settings, 192, 193
 - normalizing noise levels, 189–190
 - parameter proliferation problem, 547–548
 - polishing the texture, 193
 - previews (low-resolution), 194
 - random method for parameter settings, 193
 - remapping nonintuitive parameter ranges, 188
 - tracking previous settings, 192
 - user preset functionality, 193
- user-defined functions for raymarcher renderer, 350–352
- UV mapping, 197–199
- value noise
 - 2D slice of, 74
 - graph of, 73
 - interpolation schemes, 71–72
 - lookup table for, 181
 - overview, 70–72
 - power spectrum of, 72, 75
 - PRN table initialization, 70–71
 - value-gradient noise, 77–78
 - Wiener interpolation, 72
- value-gradient noise
 - 2D slice of, 74, 78
 - graph of, 73
 - overview, 77–78
 - power spectrum of, 75
 - Ward's Hermite noise, 73, 74, 75, 78
- vcnoise function. *See* lattice convolution noise
- vector-valued noise (VecNoise) function, 450–451
- Venus, Coriolis effect in modeling, 458–460
- venus procedure, 458–460
- vfBm procedure, 454–455
- Virtual Reality Modeling Language (VRML), 333
- virtual reality (VR), 581–582, 615
- virtualization, CPUs vs. GPUs and, 109
- viscous dumping, 606
- visibility function in deep shadow map, 208
- Vistapro software, 580
- vnoise function. *See* value noise
- volume density functions for modeling gases, 204
- volume perturbation for cloud modeling, 299–301
- volume rendering
 - algorithm for gases, 206
 - QAEB-traced
 - hypertextures, 520–525, 526
 - ray tracing for gases, 205–206

- shader for, 121–124
 - surface textures for, 195–196
- `volume_fog_animation` procedure, 238–239
- `volumeGas` plug-in, 284, 285
- volumetric cloud modeling and rendering. *See also* clouds
 - animating volumetric procedural clouds, 279–283
 - cirrus cloud models, 275–276, 277, 282, 283
 - cloud creatures, 273, 278
 - commercial packages, 284, 285
 - cumulus cloud models, 272–274, 282
 - dynamics and physics-based simulations in, 281–282
 - hardware acceleration for, 298–301
 - historical overview, 268–269
 - implicit functions for, 270–272
 - interactive cloud models, 283–284, 285
 - jet stream warping, 279–280
 - microstructure and macrostructure, 270
 - modeling system, 269–272
 - particle system for, 269, 282–283
 - physics-based approach, limitations of, 269–270
 - rendering, 272–279
 - simple cloud model, 271
 - stratus cloud models, 276, 277–278, 282, 283
 - turbulence and noise functions in, 270
 - user specification and control, 278–279
- volumetric implicit functions, for cloud modeling, 269, 270–272
- volumetric procedural modeling and texturing, 203–224. *See also* gases; volumetric cloud modeling and rendering
 - a-buffer rendering algorithm, 205
 - alternative rendering and modeling approaches, 208–209
 - basic gas shaping, 214–224
 - constant density medium for, 204
 - geometry of gases, 211–224
 - historical overview, 204–205
 - marble formation animation, 255–256
 - noise and turbulence functions, 211–214
 - patchy fog example, 216
 - power function effects on gas shape, 215–216
 - ray tracing for volume rendering, 205–206
 - rendering system, 205–208
 - sine function effects on gas shape, 215, 216
 - smoke column example, 219–224
 - solid spaces procedural framework, 209–211
 - steam rising from teacup example, 216–219, 220
 - stochastic functions used for, 203
 - uses for, 203
 - volume-rendering algorithm, 206
 - volumetric shadowing, 207–208
- `volumetric_procedural_implicit_function` for simple cloud model, 271
- von Koch snowflake
 - exact self-similarity in, 435
 - as locally Euclidean, 621
 - L-system modeling of, 308, 310–311
 - as simple fractal, 571–572
- Voronoi basis function (MojoWorld), 591, 593, 594, 597, 599
- vortex functions
 - based on 2D polar coordinate function, 250
 - based on conservation of angular momentum, 251
 - based on frame number and distance from center, 250–251
 - example of effects, 252
 - ontogenetic model of hurricane, 456–458
 - spiral, 249–251
 - vortex simulation vs., 249
- voxel automata algorithm, 359–360
- VR (virtual reality), 581–582, 615
- VRML (Virtual Reality Modeling Language), 333
- `vtransform` function, 36
- Vue d'Esprit software, 580
- wallpaper shader, 92–94
- water. *See also* liquid; volumetric procedural modeling and texturing
 - flow into hole in wall, 253–254, 255
 - fractal solid textures for, 461–465
 - noise ripples, 461–463

- water (*continued*)
 - sea surface texture, 140, 141
 - wind-blown, 463–465
- Wavefront's Maya animation system. *See* Maya animation system
- wavelets. *See also* surflets
 - defined, 341, 378
 - as fractal basis functions, 450, 497
 - in Perlin's noise function, 340–347
 - surflet integration with, 384
 - surflets as, 378–379
- weathering, texture for, 197
- Web sites
 - for cellular texturing information, 149
 - cloud rendering, 284
 - facial movement examples, 408
 - genetic programming and genetic art, 557–558
 - HighEnd3D, 284
 - Musgrave, F. Kenton, 492, 522, 557, 579, 611
 - Rooke, Steven, 557
 - Sims, Karl, 557
 - for this book, 149, 484, 524
- white noise
 - aliasing and, 68
 - defined, 67
 - irregular procedural textures and, 67
 - physical generation of, 67
 - repeatable, from PRNs, 67–68
- Wiener interpolation for value noise, 72
- wind
 - breeze effect using attractor, 247, 248, 252–253, 254
 - Coriolis effect, 458–460
 - jet stream warping for clouds, 279–280
 - ontogenetic modeling of hurricane, 456–458
 - turbulence for simulating air currents, 238
 - wind-blown waters, 463–465
- windywave procedure, 464–465
- winking, 408. *See also* texture for facial movement
- World Builder software, 580
- World Construction Set software, 580
- world space
 - changing to texture space, 168
 - changing world coordinate to image coordinate, 168
 - current space and, 24
 - described, 24
 - moving screen space point through solid space and, 228
 - PGI for object-to-world transformation, 323–324
 - placement of model into scene and, 315
 - solid textures and, 25
- World Wide Web. *See also* Web sites
 - procedural geometric modeling and, 333
- woven cloth, hypertexture example, 357, 359, 360, 361
- write_noise function, 212
- xfrog plant modeling program, 306
- z-slicing hypertextures, 35