

INDEX OF FRAGMENTS

Bold numbers indicate the first page of a fragment definition, *bold italic* numbers indicate an extension of the definition, and roman numbers indicate a use of the fragment.

- ⟨Accumulate contribution of direction ω to transfer coefficients⟩, 973, 973
- ⟨Accumulate irradiance value from nearby photons⟩, 832, **833**
- ⟨Accumulate M_o from leaf node⟩, **911**, 911
- ⟨Accumulate outgoing radiance due to reflected irradiance⟩, 816, **817**
- ⟨AdaptiveSampler Declarations⟩, **386**
- ⟨AdaptiveSampler Method Definitions⟩, **387**, **387**, **388**
- ⟨AdaptiveSampler Private Data⟩, **386**, 386
- ⟨Add bias compensation ray contribution to radiance sum⟩, 783, **784**
- ⟨Add computed irradiance value to cache⟩, 791, **796**
- ⟨Add contribution from VirtualLight v⟩, 780, **782**
- ⟨Add contribution of each light source⟩, 43, **45**
- ⟨Add contribution of escaped ray, if any⟩, 847, **849**, 849
- ⟨Add contribution of light due to scattering at p⟩, **884**, 884
- ⟨Add contributions of constant segments before/after samples⟩, 268, **269**
- ⟨Add emitted light from vertex if appropriate⟩, 847, **848**, 850
- ⟨Add InfiniteArealLight texel's contribution to SH coefficients⟩, **945**, 945
- ⟨Add IrradiancePoint ip to interior octree node⟩, 899, **900**, 900
- ⟨Add IrradiancePoint to leaf octree node⟩, 899, **900**
- ⟨Add light contribution from BSDF sampling⟩, 750, 750
- ⟨Add light sample contribution to MC estimate of SH coefficients⟩, **943**, 943
- ⟨Add light's contribution to reflected radiance⟩, 749, 750
- ⟨Add photon to unordered array of photons⟩, 822, **823**
- ⟨Add primitive to already-allocated voxel⟩, 200, **201**
- ⟨Add primitive to overlapping voxels⟩, 199, **200**
- ⟨Add primitive to scene or current instance⟩, 1065, **1068**
- ⟨Add primitives to grid voxels⟩, 197, **199**
- ⟨Add splat value at pixel⟩, 412, **413**
- ⟨Adjust normal based on orientation and handedness⟩, 102, **103**
- ⟨Advance to first relevant wavelength segment⟩, 268, **269**
- ⟨Advance to next best-candidate sample table position⟩, 383, **384**
- ⟨Advance to next child node, possibly enqueue other child⟩, 242, **244**
- ⟨Advance to next pixel for sampling for AdaptiveSampler⟩, 387, **388**
- ⟨Advance to next pixel for stratified sampling⟩, 351, 357
- ⟨Advance to next voxel⟩, 205, **207**
- ⟨Advance to sample at t_0 and update T⟩, 880, **881**
- ⟨Aggregate Declarations⟩, **192**
- ⟨AggregateTest Method Definitions⟩, **247**
- ⟨AggregateTest Private Data⟩, **247**
- ⟨AggregateVolume Private Data⟩, 597, 597
- ⟨Allocate and initialize sample⟩, 27, 27
- ⟨Allocate and request samples for sampling all lights⟩, 743, 743, 774, 788, 803
- ⟨Allocate and request samples for sampling one light⟩, 743, **744**
- ⟨Allocate BSDF, possibly doing bump mapping with bumpMap⟩, 485, 485, 487, 489
- ⟨Allocate BucketInfo for SAH partition buckets⟩, 219, 219
- ⟨Allocate film image storage⟩, 405, **407**
- ⟨Allocate IrradianceSample, get write lock, add to octree⟩, 796, **797**
- ⟨Allocate kd-tree node and continue recursively⟩, 1030, **1031**
- ⟨Allocate local variables for isect and T if needed⟩, 34, 35
- ⟨Allocate LoopSubdiv vertices and faces⟩, 151, 151
- ⟨Allocate new voxel and store primitive in it⟩, 200, **201**

- (Allocate next level of children in mesh tree), 161, 162
- (Allocate octree node if needed and continue recursive traversal), 1024, 1025
- (Allocate SH coefficient vector pointers for sample points), 959, 960
- (Allocate space for samples and intersections), 30, 31
- (Allocate storage for sample pointers), 345, 345
- (Allocate storage for sample values), 345, 346
- (Allocate working memory for kd-tree construction), 232, 236, 240
- (Allow integrators to do preprocessing for the scene), 27, 27
- (AnimatedTransform Method Definitions), 97, 99, 100
- (AnimatedTransform Private Data), 96
- (AnimatedTransform Public Methods), 96, 100
- (Anisotropic Public Methods), 458, 458, 460
- (API Cleanup), 1052, 1053, 1056
- (API Function Declarations), 1055, 1058, 1061, 1064
- (API Function Definitions), 1051, 1052, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1063, 1065, 1069, 1070, 1071, 1072
- (API Global Variables), 1051
- (API Initialization), 1051, 1053, 1056, 1060
- (API Local Classes), 1054, 1054, 1056, 1059, 1060
- (API Macros), 1053, 1055, 1063
- (API Static Data), 1052, 1054, 1055, 1056, 1060, 1066
- (Apply boundary rule for even vertex), 162, 165
- (Apply box filter to checkerboard region), 549, 551
- (Apply edge rules to compute new vertex position), 167, 167
- (Apply large step to *i*th camera PathSample), 840, 841
- (Apply one-ring rule for even vertex), 162, 163
- (Apply SH definitions to compute final (*l*, *m*) values), 936, 941
- (Apply sWeights to zoom in *s* direction), 532, 534
- (AreaLight Interface), 623, 625
- (Bail out if we found a hit closer than the current node), 242, 242
- (BBox Method Definitions), 71, 74, 194
- (BBox Public Data), 70, 71
- (BBox Public Methods), 70, 70, 71, 72, 73
- (Begin the KdTree building process), 1029, 1029
- (BestCandidate Sampling Constants), 380
- (BestCandidateSampler Declarations), 381
- (BestCandidateSampler Method Definitions), 383, 383
- (BestCandidateSampler Private Data), 381, 381, 383, 384
- (BestCandidateSampler Public Methods), 381, 381
- (BilerpTexture Declarations), 522
- (BilerpTexture Private Data), 522, 522
- (BilerpTexture Public Methods), 522, 522, 523
- (Blinn Public Methods), 456, 457
- (BlockedArray Private Data), 1017, 1018
- (BlockedArray Public Methods), 1017, 1018, 1019, 1020
- (BoundEdge Public Methods), 235, 235, 237
- (Box Filter Declarations), 396
- (Box Filter Method Definitions), 396
- (BRDFToBTDF Public Methods), 431, 431, 431
- (BSDF Declarations), 428, 478, 482, 705, 706
- (BSDF Inline Functions), 426, 426, 427, 695
- (BSDF Inline Method Definitions), 479
- (BSDF Method Definitions), 478, 481, 706
- (BSDF Private Data), 478, 479, 479, 488
- (BSDF Private Methods), 478, 483
- (BSDF Public Data), 478, 479
- (BSDF Public Methods), 478, 479, 480, 482, 708
- (BSDFSAMPLE Public Methods), 705, 705, 705
- (BSSRDF Declarations), 598
- (BSSRDF Private Data), 598, 599
- (BSSRDF Public Methods), 598, 598, 599
- (Build BVH from primitives), 209, 210
- (Build kd-tree for accelerator), 229, 232
- (Build kd-trees for indirect and caustic photons), 804, 813
- (BVHAccel Local Declarations), 211, 211, 216, 217, 222, 226
- (BVHAccel Method Definitions), 209, 213, 224
- (BVHAccel Private Data), 210, 223
- (BVHBuildNode Public Methods), 211, 212, 212
- (BxDF Declarations), 428, 431, 436, 437, 439, 440, 444, 447, 454, 456, 458, 460, 464, 467
- (BxDF Interface), 428, 429, 429, 430
- (BxDF Local Definitions), 466
- (BxDF Method Definitions), 431, 432, 437, 441, 444, 447, 451, 454, 455, 461, 462, 465, 466, 469, 694, 695, 696, 699, 700, 701, 703, 704
- (BxDF Public Data), 428, 429
- (BxDF Utility Functions), 435, 435
- (Camera Declarations), 302, 305
- (Camera Interface), 302, 302, 303
- (Camera Method Definitions), 303, 305
- (Camera Public Data), 302, 302
- (Check for intersection in current voxel and advance to next), 205, 205
- (Check for intersections inside leaf node), 242, 244
- (Check for ray intersection against *x* and *y* slabs), 226, 226
- (Check one primitive inside leaf node), 244, 244
- (Check ray against BVH node), 225, 225
- (Check ray against overall grid bounds), 202, 203
- (Check sample against crop window, goto again if outside), 383, 385
- (Checkerboard2DTexture Private Data), 547, 547, 548
- (Checkerboard2DTexture Public Methods), 547, 547, 548
- (Checkerboard3DTexture Private Data), 552, 553
- (Checkerboard3DTexture Public Methods), 552, 552, 553
- (CheckerboardTexture Declarations), 547, 552
- (Choose level of detail for EWA lookup and perform EWA filtering), 540, 541
- (Choose light and call EstimatedDirect() for Metropolis vertex), 848, 848
- (Choose light and sample ray to start light path), 846, 846
- (Choose light source to trace virtual light path from), 777, 777
- (Choose light to shoot photon from), 807, 807
- (Choose point on disk oriented toward infinite light direction), 714, 715
- (Choose random rays, rayAccel and rayAll for testing), 247, 247
- (Choose ray direction for testing accelerator), 247, 248
- (Choose ray epsilon for testing accelerator), 247, 248

- (Choose ray origin for testing accelerator), 247, 248
- (Choose split axis position for interior node), 234, 236
- (Choose split direction and partition data), 1030, 1031
- (Choose which axis to split along), 236, 236
- (Choose which BxDF to sample), 706, 707
- (Clamp ellipse eccentricity if too large), 540, 540
- (Classify primitives with respect to split), 234, 239
- (Clean up after rendering), 1071, 1071
- (Clean up after rendering and store final image), 27, 29
- (ClosePhoton Public Methods), 822, 822, 822
- (CoefficientSpectrum Protected Data), 264, 264
- (CoefficientSpectrum Public Methods), 264, 264, 264, 265, 266
- (Compare contrast of sample differences to threshold), 388, 389
- (Compute (u, v) offsets at auxiliary points), 507, 508
- (Compute acceptance probability for proposed sample), 857, 860
- (Compute α' for RGB component, compute scattering properties), 913, 913
- (Compute auxiliary intersection points with plane), 507, 507
- (Compute average value of given SPD over *i*th sample's range), 267, 267
- (Compute barycentric coordinates for point), 146, 147
- (Compute bound of primitive centroids, choose split dimension *dim*), 213, 214
- (Compute bounding box of region used to generate random rays), 247, 247
- (Compute bounds and choose grid resolution), 197, 198
- (Compute bounds for kd-tree construction), 232, 232
- (Compute bounds of all primitives in BVH node), 213, 213
- (Compute bounds of data from start to end), 1031, 1031
- (Compute bump-mapped differential geometry), 495, 496
- (Compute caustic lighting for photon map integrator), 818
- (Compute cell indices for dots), 559, 559
- (Compute closed-form box-filtered Checkerboard2DTexture value), 548, 549
- (Compute coefficients for fundamental forms), 122, 122, 128
- (Compute conditional sampling distribution for \vec{v}), 672, 672
- (Compute contribution between camera and light vertices), 851, 851
- (Compute contribution for random sample for MLT bootstrapping), 855, 856
- (Compute contribution of proposed sample), 857, 860
- (Compute coordinate system for sphere sampling), 720, 721
- (Compute cosine of cone surrounding projection directions), 616, 617
- (Compute cosine term of Oren-Nayar model), 451, 451
- (Compute cost for split at *i*th edge), 237, 238
- (Compute cost of all splits for axis to find best), 236, 237
- (Compute costs for splitting after each bucket), 219, 221
- (Compute current voxel for axis), 204, 204
- (Compute cylinder $\partial \mathbf{n} / \partial u$ and $\partial \mathbf{n} / \partial v$), 128, 128
- (Compute cylinder $\partial \mathbf{p} / \partial u$ and $\partial \mathbf{p} / \partial v$), 128, 128
- (Compute cylinder hit point and ϕ), 127, 127, 128
- (Compute deltas for triangle partial derivatives), 143, 143
- (Compute differential changes in origin for ortho camera rays), 306, 307
- (Compute differential changes in origin for perspective camera rays), 310, 312
- (Compute differential reflected directions), 512, 513
- (Compute direct illumination for Metropolis path vertex), 847, 848, 850
- (Compute direct lighting for DirectLightingIntegrator integrator), 744
- (Compute direct lighting for irradiance cache), 789
- (Compute $\partial \mathbf{n} / \partial u$ and $\partial \mathbf{n} / \partial v$ from fundamental form coefficients), 122, 122, 128
- (Compute double spherical integral for BSDF matrix), 978, 979
- (Compute ellipse coefficients to bound EWA filter region), 541, 542
- (Compute ellipse minor and major axes), 540, 540
- (Compute emission-only source term at *p*), 880, 882
- (Compute emitted and reflected light at ray intersection point), 42, 43
- (Compute emitted light if ray hit an area light source), 43, 43
- (Compute environment camera ray direction), 319, 319
- (Compute estimate error term and possibly use sample), 793, 794
- (Compute exitant radiance from photons for glossy surface), 831, 832
- (Compute exitant radiance L_{indir} using radiance photons), 824, 824, 827
- (Compute falloff inside spotlight cone), 614, 614
- (Compute film image extent), 405, 406
- (Compute final average value for probe and cleanup), 963, 965
- (Compute final coefficients c_o using BSDF matrix), 981, 982
- (Compute first barycentric coordinate), 141, 142
- (Compute Fresnel reflectance for dielectric), 437, 438
- (Compute general sphere weight), 722, 722
- (Compute glossy BSDF matrix for PRT), 980, 980
- (Compute gradient weights), 555, 556
- (Compute image resampling weights for *i*th texel), 532, 533
- (Compute incident direction by reflecting about ω_h), 696, 697, 699
- (Compute incident radiance from radiance probe coefficients), 966, 967
- (Compute incident radiance sample from light, update SH coeffs), 943, 943
- (Compute index into measured BRDF tables), 469, 469
- (Compute indices of refraction for dielectric), 438, 438
- (Compute indices $wh\Theta$ Index, $wd\Theta$ Index, $wd\Phi$ Index), 469, 470
- (Compute indirect illumination with virtual lights), 780
- (Compute indirect lighting for irradiance cache), 789
- (Compute indirect lighting for photon map integrator), 818
- (Compute InfiniteAreaLight ray PDF), 729, 730
- (Compute initial parametric range of ray inside kd-tree extent), 240, 240
- (Compute integral of product of incident radiance and transfer function), 972, 973

- (Compute integral of step function at x_i), 648, 648
- (Compute integrator samples for best-candidate sample), 383, 385
- (Compute interpolated matrix as product of interpolated components), 99, 100
- (Compute intersection distance along ray), 118, 119
- (Compute intersections using accelerator and exhaustive testing), 247, 248
- (Compute irradiance at current point), 791, 795
- (Compute irradiance sample's contribution extent and bounding box), 796, 797
- (Compute irradiance values at sample points), 897, 897
- (Compute i th candidate point p in cell's bounding box), 963, 964
- (Compute K_l^m coefficients), 936, 939
- (Compute Legendre polynomial values for $\cos \theta$), 936, 936
- (Compute light power CDF for photon shooting), 804, 805
- (Compute $L[\text{current}]$ for initial sample), 858, 858
- (Compute $m = 0$ Legendre values using recurrence), 936, 938
- (Compute $m = 1, \dots, l - 2$ values using Legendre recurrence), 936, 939
- (Compute $m = l$ edge using Legendre recurrence), 936, 938
- (Compute $m = l - 1$ edge using Legendre recurrence), 936, 939
- (Compute marginal sampling distribution $p[\bar{v}]$), 672, 673
- (Compute MIPMap level for trilinear filtering), 536, 537
- (Compute MIS weight for BSDF-sampled gather ray), 824, 825
- (Compute MIS weight for photon-sampled gather ray), 827, 827
- (Compute Monte Carlo estimate of i th sample for transfer matrix), 976, 977
- (Compute M_o at node if error is low enough), 911, 911
- (Compute new odd edge vertices), 162, 166
- (Compute new transformation matrix M without translation), 97, 97
- (Compute next digit of radical inverse), 362, 362
- (Compute next matrix R_{next} in series), 98, 98
- (Compute noise cell coordinates and offsets), 555, 555
- (Compute norm of difference between R and R_{next}), 98, 98
- (Compute number of octaves for antialiased FBM), 561, 562, 564
- (Compute number of SamplerRendererTasks to create for rendering), 28, 29
- (Compute number of specular vertices for each path length), 849, 850
- (Compute odd vertex on k th edge), 166, 167
- (Compute offset along CDF segment), 649, 649
- (Compute offset positions and evaluate displacement texture), 495, 495
- (Compute offset rays for PerspectiveCamera ray differentials), 313
- (Compute overall PDF with all matching BxDFs), 706, 707
- (Compute parametric distance along ray to split plane), 242, 242
- (Compute PDF for photon-sampling of direction w_i), 825, 827, 827
- (Compute PDF for sampled infinite light direction), 727, 728
- (Compute PDF for sampled offset), 649, 649
- (Compute PDF for ω_i from anisotropic distribution), 699, 700, 701
- (Compute PDF for ω_i from Blinn distribution), 696, 699, 699
- (Compute perfect specular reflection direction), 441, 441
- (Compute permutation tables for each base), 367, 368
- (Compute pixel spacing in world space at intersection point), 789, 790
- (Compute plane intersection for disk), 131, 131
- (Compute point on plane of focus), 315, 318
- (Compute probe coordinates and offsets for lookup point), 966, 966
- (Compute projective camera screen transformations), 305, 306
- (Compute projective camera transformations), 305, 305
- (Compute proposed mutation to current sample), 857, 859
- (Compute quadratic cylinder coefficients), 127, 127
- (Compute quadratic sphere coefficients), 117, 118, 123
- (Compute quadratic t values), 118, 119
- (Compute radiance along path using path tracing), 845, 846, 846
- (Compute radiance along paths using bidirectional path tracing), 846, 846
- (Compute radiance for radiance photon i), 816, 816
- (Compute random points on surfaces of scene), 959, 962
- (Compute randomly permuted table of pixel indices for large steps), 858, 859
- (Compute range of radiance photons to process in task), 816, 816
- (Compute raster sample from table), 383, 385
- (Compute ray differential rd for specular reflection), 46, 512
- (Compute rayEpsilon for quadric intersection), 117, 123, 127, 131
- (Compute reflectance SampledSpectrum with $rgb[0]$ as minimum), 278, 278
- (Compute reflected light at camera path vertex), 849, 850
- (Compute reflected lighting using radiance probes), 966, 966
- (Compute reflected radiance due to irradiance and BSDF), 790, 791
- (Compute reflected radiance using diffuse PRT), 972
- (Compute reflected radiance with glossy PRT at point), 981
- (Compute reflection for radiance probes integrator), 966
- (Compute region in which to compute incident radiance probes), 963, 963
- (Compute representation of depth-first traversal of BVH tree), 210, 223
- (Compute sample patterns for single scattering samples), 884
- (Compute sample's raster extent), 408, 408
- (Compute sampled half-angle vector ω_h for Blinn distribution), 696, 697
- (Compute samples for emitted rays from lights), 776, 777
- (Compute sampling distributions for rows and columns of image), 725, 727
- (Compute sampling rate in each dimension), 959, 959
- (Compute scalar-valued image img from environment map), 725, 727
- (Compute scale S using rotation and original matrix), 97, 98
- (Compute scene bounds and initialize probe integrators), 959, 959
- (Compute second barycentric coordinate), 141, 142
- (Compute SH coefficients of incident radiance at point p), 963, 964

- (Compute SH radiance transfer matrix at point and SH coefficients), 981, 981
- (Compute sine and tangent terms of Oren-Nayar model), 451, 451
- (Compute single-scattering source term at p), 884
- (Compute $\sin i\phi$ and $\cos i\phi$ using recurrence), 940, 941
- (Compute $\sin \phi$ and $\cos \phi$ values), 936, 940
- (Compute sint using Snell's law), 438, 438
- (Compute sphere $\partial \mathbf{n} / \partial u$ and $\partial \mathbf{n} / \partial v$), 120, 122
- (Compute sphere $\partial p / \partial u$ and $\partial p / \partial v$), 120, 121
- (Compute sphere hit position and ϕ), 117, 119, 120, 123
- (Compute squared radius and filter texel if inside ellipse), 543, 544
- (Compute sum of octaves of noise for FBm), 561, 563
- (Compute sum of octaves of noise for turbulence), 564, 565
- (Compute t to intersection point), 141, 142
- (Compute tangents of boundary face), 171, 173
- (Compute tangents of interior face), 171, 172
- (Compute texel (s, t) accounting for boundary conditions), 535, 535
- (Compute texel (s, t) in s -zoomed image), 534, 534
- (Compute texture coordinate differentials for sphere (u, v) mapping), 516, 517
- (Compute texture differentials for 2D identity mapping), 515, 515
- (Compute the ellipse's (s, t) bounding box in texture space), 541, 542
- (Compute total area of shapes in ShapeSet and area CDF), 723
- (Compute total number of sample values needed), 345, 345
- (Compute transmitted ray direction), 444, 445
- (Compute triangle partial derivatives), 141, 143
- (Compute trilinear interpolation of weights), 555, 558
- (Compute valence of boundary vertex), 158, 159
- (Compute valence of interior vertex), 158, 158
- (Compute value of BSDF for sampled direction), 706, 708
- (Compute s_1), 141, 141
- (Compute vertex tangents on limit surface), 161, 171
- (Compute virtual light's tentative contribution LLight), 780, 782
- (Compute voxel coordinates and offsets for Pobj), 593, 594
- (Compute voxel widths and allocate voxels), 197, 199
- (Compute weight for bidirectional path, pathWt), 851, 852
- (Compute ω_h and transform ω_i to halfangle coordinate system), 469, 469
- (Compute world-to-object transformation for instance), 190, 191
- (Compute x and y pixel sample range for sub-window), 341, 342
- (Compute XYZ matching functions for SampledSpectrum), 271, 272
- (ConstantTexture Declarations), 520
- (ConstantTexture Public Methods), 520, 520
- (Convert EWA coordinates to appropriate scale for level), 541, 541
- (Convert image to RGB and compute final pixel values), 412, 412
- (Convert infinite light sample point to direction), 727, 728
- (Convert leaf node to interior node, redistribute points), 900, 900
- (Convert light sample weight to solid angle measure), 717, 717
- (Convert pixel XYZ color to RGB), 412, 412
- (Convert reflectance spectrum to RGB), 277, 278
- (Convert texels to type Tmemory and create MIPMap), 525, 526
- (Convolve incident radiance to compute irradiance function), 966, 968
- (Copy integrator samples into samples[i]), 376, 376
- (Copy photon directions to local array), 819, 823
- (Cosine-sample the hemisphere, flipping the direction if necessary), 694, 694, 701
- (Create and initialize new odd vertex), 167, 167
- (Create and launch SamplerRendererTasks for rendering image), 27, 28
- (Create float texture and store in floatTextures), 1063, 1063
- (Create initial Shape for animated shape), 1067, 1068
- (Create leaf BVHBuildNode), 213, 213, 215, 221
- (Create leaf if no good splits were found), 234, 239
- (Create leaf node of kd-tree if we've reached the bottom), 1030, 1030
- (Create mipmap for GoniophotometricLight), 619, 620
- (Create octree of clustered irradiance samples), 897, 898
- (Create one-valued MIPMap), 525, 527
- (Create primitive for animated shape), 1065, 1067
- (Create primitive for static shape), 1065, 1066
- (Create ProjectionLight MIP-map), 616, 616
- (Create reader-writer mutex for grid), 197, 205
- (Create scene and render), 1071, 1071
- (Create scene bounding sphere to catch rays that leave the scene), 890, 891, 962
- (Create virtual light and sample new ray for path), 777, 778
- (Create virtual light at ray intersection point), 778, 779
- (Create interior flattened BVH node), 224, 224
- (CreateRadianceProbes Method Definitions), 959, 963
- (CreateRadianceProbes Private Data), 958
- (CreateRadianceProbes Public Methods), 958
- (Cylinder Declarations), 124
- (Cylinder Method Definitions), 125, 126, 127, 129, 719
- (Cylinder Private Data), 124, 126
- (CylindricalMapping2D Private Methods), 517, 517
- (Declare basic MLTTask variables and prepare for sampling), 857, 858
- (Declare common path integration variables), 767, 767
- (Declare common variables for SurfacePointTask::Run()), 892, 892
- (Declare local variables for PhotonShootingTask), 806, 806, 806, 815
- (Declare local variables used for rendering loop), 30, 31
- (Declare sample parameters for light source sampling), 742, 743, 788, 803

- ⟨Declare shared variables for photon shooting⟩, 804, 804, 815
- ⟨Declare shared variables for Poisson point generation⟩, 890, 890
- ⟨Declare variables for storing and computing MLT samples⟩, 858, 858
- ⟨Decorrelate sample dimensions⟩, 352, 353
- ⟨DensityRegion Protected Data⟩, 591, 592
- ⟨DensityRegion Public Methods⟩, 591, 592, 592
- ⟨Deposit photon at surface⟩, 808, 810
- ⟨Determine bases b_i and their sum⟩, 367, 367
- ⟨Determine how many tiles to use in each dimension, n_x and n_y ⟩, 341, 342
- ⟨Determine which children the item overlaps⟩, 1024, 1026
- ⟨Determine which octree child node p is inside⟩, 1027, 1027
- ⟨DifferentialGeometry Declarations⟩, 102
- ⟨DifferentialGeometry Method Definitions⟩, 102, 505
- ⟨DifferentialGeometry Public Data⟩, 102, 102, 505
- ⟨DiffuseAreaLight Declarations⟩, 625
- ⟨DiffuseAreaLight Method Definitions⟩, 626, 627, 718
- ⟨DiffuseAreaLight Protected Data⟩, 625, 626
- ⟨DiffuseAreaLight Public Methods⟩, 625, 626, 627
- ⟨DiffusePRTIntegrator Method Definitions⟩, 972, 972
- ⟨DiffusePRTIntegrator Private Data⟩, 972
- ⟨DiffusionReflectance Data⟩, 907, 907
- ⟨DiffusionReflectance Public Methods⟩, 907, 907, 908
- ⟨DipoleSubsurfaceIntegrator Helper Declarations⟩, 898
- ⟨DipoleSubsurfaceIntegrator Local Declarations⟩, 898, 907
- ⟨DipoleSubsurfaceIntegrator Method Definitions⟩, 897, 911
- ⟨DipoleSubsurfaceIntegrator Private Data⟩, 888, 897, 898
- ⟨DipoleSubsurfaceIntegrator Public Methods⟩, 887
- ⟨DirectLightingIntegrator Declarations⟩, 742, 742
- ⟨DirectLightingIntegrator Method Definitions⟩, 743
- ⟨DirectLightingIntegrator Private Data⟩, 742, 742, 742
- ⟨Discard directions behind projection light⟩, 617, 617
- ⟨Disk Declarations⟩, 129
- ⟨Disk Method Definitions⟩, 130, 131, 132, 719
- ⟨Disk Private Data⟩, 129, 130
- ⟨DistantLight Declarations⟩, 621
- ⟨DistantLight Method Definitions⟩, 621, 622, 623, 714
- ⟨DistantLight Private Data⟩, 621, 622
- ⟨Distribution1D Private Data⟩, 648, 648
- ⟨Distribution1D Public Methods⟩, 648, 648, 649, 650
- ⟨Distribution2D Private Data⟩, 672, 672, 673
- ⟨Distribution2D Public Methods⟩, 672, 673, 673
- ⟨Do atomic add with gcc x86 inline assembly⟩, 1036, 1037
- ⟨Do bias compensation for bounding geometry term⟩, 780, 783
- ⟨Do emission-only volume integration in vr ⟩, 880, 880
- ⟨Do large step mutation of cameraSample⟩, 840, 840
- ⟨Do one-bounce final gather for photon map⟩, 818, 819
- ⟨Do photon map lookup at intersection point⟩, 831, 831
- ⟨Do trilinear interpolation to compute SH coefficients at point⟩, 967, 967
- ⟨DotsTexture Declarations⟩, 559
- ⟨DotsTexture Private Data⟩, 559, 559
- ⟨DotsTexture Public Methods⟩, 559, 559, 559
- ⟨dtrace.d*⟩, 1008
- ⟨Either create leaf or split primitives at selected SAH bucket⟩, 219, 221
- ⟨EmissionIntegrator Declarations⟩, 877
- ⟨EmissionIntegrator Method Definitions⟩, 877, 879, 880
- ⟨EmissionIntegrator Private Data⟩, 877, 877, 877
- ⟨EmissionIntegrator Public Methods⟩, 877, 877
- ⟨Enqueue secondChild in todo list⟩, 244, 244
- ⟨Ensure there are no pushed graphics states⟩, 1071, 1071
- ⟨EnvironmentCamera Declarations⟩, 318
- ⟨EnvironmentCamera Method Definitions⟩, 319
- ⟨EnvironmentCamera Public Methods⟩, 318, 319
- ⟨Erase primitives, lights, and volume regions from RenderOptions⟩, 1072, 1072
- ⟨Estimate direct lighting from light samples⟩, 745, 745
- ⟨Estimate indirect lighting with irradiance cache⟩, 789, 789
- ⟨Estimate one term of ρ_{hd} ⟩, 703, 704
- ⟨Estimate one term of ρ_{hh} ⟩, 704, 704
- ⟨Estimate reflected radiance due to incident photons⟩, 831, 831
- ⟨Estimate screen space change in p and (u, v) ⟩, 505, 507
- ⟨Evaluate BSDF at hit point⟩, 43, 43
- ⟨Evaluate BSSRDF and possibly compute subsurface scattering⟩, 908
- ⟨Evaluate filter value at (x, y) pixel⟩, 409, 410
- ⟨Evaluate irradiance function and accumulate reflection⟩, 966, 969
- ⟨Evaluate outgoing radiance function for ω_o and add to L ⟩, 981, 982
- ⟨Evaluate radiance along camera ray⟩, 32, 33
- ⟨Evaluate single check if filter is entirely inside one of them⟩, 549, 550
- ⟨Evaluate textures for MatteMaterial material and allocate BRDF⟩, 485, 486
- ⟨Exit task if enough photons have been found⟩, 806, 813
- ⟨ExponentialDensity Declarations⟩, 595
- ⟨ExponentialDensity Private Data⟩, 595, 596
- ⟨ExponentialDensity Public Methods⟩, 595, 595, 596
- ⟨Extract rotation R from transformation matrix⟩, 97, 98
- ⟨Extract translation T from transformation matrix⟩, 97, 97
- ⟨FBmTexture Declarations⟩, 565
- ⟨FBmTexture Private Data⟩, 565, 566
- ⟨FBmTexture Public Methods⟩, 565, 566, 566
- ⟨Figure out which η is incident and which is transmitted⟩, 444, 445
- ⟨Fill in DifferentialGeometry from triangle hit⟩, 141, 145
- ⟨Film Declarations⟩, 403
- ⟨Film Interface⟩, 403, 403, 403, 404
- ⟨Film Method Definitions⟩, 404
- ⟨Film Public Data⟩, 403, 403
- ⟨Filter Declarations⟩, 393
- ⟨Filter four texels from finer level of pyramid⟩, 536, 536
- ⟨Filter Interface⟩, 393, 393, 394
- ⟨Filter Public Data⟩, 393, 394
- ⟨Find (u, v) sample coordinates in infinite light texture⟩, 727, 727
- ⟨Find bucket to split at that minimizes SAH metric⟩, 219, 221

- (Find camera ray for sample[i]), 32, 32
 (Find closest intersection of ray with shapes in ShapeSet), 723, 723
 (Find indirect photons around point for importance sampling), 819, 821
 (Find light and BSDF sample values for direct lighting estimate), 745, 746
 (Find next vertex of path), 767, 769
 (Find parametric representation of cylinder hit), 127, 128
 (Find parametric representation of disk hit), 131, 132
 (Find parametric representation of sphere hit), 117, 120
 (Find quadratic discriminant), 118, 118
 (Find ray after shifting one pixel in the x direction), 303, 303
 (Find ray intersection with scene geometry or bounding sphere), 893, 894
 (Find stepAxis for stepping to next voxel), 207, 208
 (Find surrounding CDF segments and offset), 649, 649, 650
 (Find voxel extent of primitive), 199, 200
 (Find voxelsPerUnitDist for grid), 198, 199
 (Finish vertex initialization), 151, 158
 (floatfile.h*), 1006
 (Follow path i from light to create virtual lights), 776, 777
 (Follow photon path through scene and record intersections), 807, 808
 (Follow photon paths for a block of samples), 806, 807
 (Follow ray path and attempt to deposit candidate sample points), 892, 893
 (Follow ray through BVH nodes to find primitive intersections), 224, 225
 (Free MemoryArena memory from computing image sample values), 32, 33
 (Fresnel Interface), 436, 436
 (FresnelBlend Private Data), 460, 461
 (FresnelBlend Public Methods), 460, 462
 (FresnelConductor Public Methods), 436, 437
 (FresnelDielectric Public Methods), 437, 437
 (Function Definitions), 2, 3
 (Gaussian Filter Declarations), 397
 (Gaussian Filter Method Definitions), 398
 (GaussianFilter Private Data), 397, 397
 (GaussianFilter Public Methods), 397, 397
 (GaussianFilter Utility Functions), 397, 398
 (Generate camera path from camera path samples), 845, 845
 (Generate camera rays and compute radiance along rays), 32, 32
 (Generate initial stratified samples into sampleBuf memory), 352, 352
 (Generate lens, time, and integrator samples for HaltonSampler), 364, 366
 (Generate LHS samples along diagonal), 356, 356
 (Generate low-discrepancy pixel samples), 375, 376
 (Generate photonRay from light source and initialize alpha), 807, 808
 (Generate random ray from intersection point), 893, 894
 (Generate random sample and path radiance for MLT bootstrapping), 855, 856
 (Generate raster and camera samples), 309, 309, 312
 (Generate sample with Halton sequence and reject if outside image extent), 364, 365
 (Generate stratified camera samples for (xPos, yPos)), 351, 352
 (Generate stratified samples for integrators), 354, 356
 (GeometricPrimitive Declarations), 187
 (GeometricPrimitive Method Definitions), 188, 189
 (GeometricPrimitive Private Data), 187, 188
 (GeometricPrimitive Public Methods), 187, 188
 (Geometry Declarations), 57, 63, 65, 66, 69, 70
 (Geometry Inline Functions), 59, 60, 61, 62, 63, 65, 66, 291, 292
 (Get animatedWorldToObject transform for shape), 1067, 1068
 (Get irradiance E and average incident direction wi at point p), 790, 791
 (Get new block of memory for MemoryArena), 1016, 1016
 (Get next free node from nodes array), 233, 233
 (Get node children pointers for ray), 242, 243
 (Get one-ring vertices for boundary vertex), 164, 165
 (Get one-ring vertices for interior vertex), 164, 164
 (Get outgoingBSDFSample for sampling new path direction), 768, 769
 (Get radiance probe coefficients around lookup point), 966, 966
 (Get samples from Sampler and update image), 30, 32
 (Get sub-Sampler for SamplerRendererTask), 30, 31
 (Get SurfacePoints for translucent objects in scene), 897, 897
 (Get triangle vertices in p1, p2, and p3), 139, 140, 140, 141, 145, 719
 (Global Constants), 1000, 1002, 1012
 (Global Forward Declarations), 263, 1051
 (Global Include Files), 999, 1000, 1005
 (Global Inline Functions), 118, 1000, 1001, 1002, 1005
 (Global Macros), 1009
 (GlossyPRTIntegrator Method Definitions), 980
 (GlossyPRTIntegrator Private Data), 980, 980
 (GlossyPRTIntegrator Public Methods), 980
 (GonioPhotometricLight Declarations), 618
 (GonioPhotometricLight Method Definitions), 619, 621
 (GonioPhotometricLight Private Data), 618, 620
 (GonioPhotometricLight Public Methods), 618, 621
 (Grab next node to process from todo list), 242, 245
 (Graphics State), 1059, 1064, 1064, 1065
 (Graphics State Methods), 1059, 1067
 (GraphicsState Constructor Implementation), 1060, 1064
 (GridAccel Declarations), 196
 (GridAccel Method Definitions), 197, 202, 206
 (GridAccel Private Data), 196, 198, 198, 199, 201, 205
 (GridAccel Private Methods), 196, 200, 200, 201
 (GridAccel Public Methods), 196, 208
 (HaltonSampler Method Definitions), 364
 (HaltonSampler Private Data), 364
 (Hand kd-tree node to processing function), 1032, 1033
 (Handle boundary conditions for matrix interpolation), 99, 99
 (Handle cases with out-of-bounds range or single sample only), 268, 268
 (Handle degenerate parametric mapping), 147, 147
 (Handle first region of disk), 668, 668

- (Handle new edge)*, 156, 156
- (Handle photon/surface intersection)*, 808, 808
- (Handle previously seen edge)*, 156, 156
- (Handle ray that doesn't intersect any geometry)*, 34, 35
- (Handle ray that leaves the scene during path generation)*, 843, 844
- (Handle ray with negative direction for voxel stepping)*, 204, 205
- (Handle ray with positive direction for voxel stepping)*, 204, 204
- (Handle total internal reflection)*, 438, 439
- (Handle total internal reflection for transmission)*, 445, 446
- (Handle zero determinant for triangle partial derivative matrix)*, 143, 143
- (HomogeneousVolumeDensity Declarations)*, 589
- (HomogeneousVolumeDensity Private Data)*, 589, 590
- (HomogeneousVolumeDensity Public Methods)*, 589, 590, 590, 591
- (IGIIntegrator Local Structures)*, 779
- (IGIIntegrator Method Definitions)*, 774, 776
- (IGIIntegrator Private Data)*, 773, 774, 775, 779
- (ImageFilm Declarations)*, 404
- (ImageFilm Method Definitions)*, 405, 408, 411, 412
- (ImageFilm Private Data)*, 404, 405, 406, 407
- (ImageIO Declarations)*, 1004, 1004
- (ImageTexture Declarations)*, 524
- (ImageTexture Method Definitions)*, 524, 525, 527
- (ImageTexture Private Data)*, 524, 524, 525
- (ImageTexture Private Methods)*, 524, 526, 527
- (ImageTexture Public Methods)*, 524, 527
- (Incorporate results from +z face to coefficients)*, 946, 948
- (InfiniteAreaCube Public Methods)*, 948, 948, 949
- (InfiniteAreaLight Declarations)*, 629
- (InfiniteAreaLight Method Definitions)*, 630, 631, 727, 729, 944
- (InfiniteAreaLight Private Data)*, 629, 631, 727
- (InfiniteAreaLight Public Methods)*, 629, 631
- (InfiniteAreaLight Utility Classes)*, 948
- (Init SubsurfaceOctreeNode leaf from IrradiancePoints)*, 901, 901
- (Initialize A and C matrices for barycentrics)*, 147, 147
- (Initialize A, Bx, and By matrices for offset computation)*, 508, 508
- (Initialize basic variables for camera path vertex)*, 847, 847, 849
- (Initialize BucketInfo for SAH partition buckets)*, 219, 220
- (Initialize buildData array for primitives)*, 210, 210
- (Initialize common variables for CreateRadProbeTask::Run())*, 963, 963
- (Initialize common variables for Whitted integrator)*, 43, 43
- (Initialize depth of field parameters)*, 305, 314
- (Initialize DifferentialGeometry from parameters)*, 102, 102, 505
- (Initialize DifferentialGeometry from parametric information)*, 117, 122, 127, 131
- (Initialize edges for axis)*, 236, 236
- (Initialize EWA filter weights if needed)*, 530, 544
- (Initialize first three columns of viewing matrix)*, 84, 85
- (Initialize fourth column of viewing matrix)*, 84, 84
- (Initialize Global Variables)*, 2, 2, 3
- (Initialize interior node and continue recursion)*, 233, 234
- (Initialize i th MIPMap level from i — 1st level)*, 534, 536
- (Initialize leaf node if termination criteria met)*, 233, 233
- (Initialize levels of MIPMap from image)*, 530, 534
- (Initialize light and bsdf samples for single light sample)*, 746, 747
- (Initialize matrices for chosen projection plane)*, 508, 509
- (Initialize most detailed level of MIPMap)*, 534, 535
- (Initialize primitives with primitives for grid)*, 197, 198
- (Initialize primNums for kd-tree construction)*, 232, 232
- (Initialize ProjectionLight projection matrix)*, 616, 616
- (Initialize samples with computed sample values)*, 375, 376
- (Initialize sampling PDFs for infinite area light)*, 630, 725
- (Initialize stratified samples with sample values)*, 352, 354
- (Initialize Triangle shading geometry with n and s)*, 145, 146
- (Initialize volumeRegion from volume region(s))*, 1072, 1072
- (Integrator Declarations)*, 740, 740
- (Integrator Interface)*, 740, 740, 740
- (Integrator Utility Functions)*, 46, 709, 745, 746, 749
- (Interpolate (u, v) triangle parametric coordinates)*, 141, 143
- (Interpolate rotation at dt)*, 99, 99
- (Interpolate scale at dt)*, 99, 99
- (Interpolate translation at dt)*, 99, 99
- (Intersect ray with primitives in leaf BVH node)*, 225, 226
- (Intersect sample ray with area light geometry)*, 717, 717
- (Intersection Declarations)*, 186
- (Intersection Method Definitions)*, 484, 625
- (Intersection Public Data)*, 186, 186
- (Intersection Public Methods)*, 186, 484
- (IrradianceCacheIntegrator Declarations)*, 786
- (IrradianceCacheIntegrator Local Declarations)*, 792, 793
- (IrradianceCacheIntegrator Method Definitions)*, 788, 788, 790, 792, 793
- (IrradianceCacheIntegrator Private Data)*, 786, 787, 788
- (IrradProcess Data)*, 792, 793
- (IrradProcess Public Methods)*, 792, 794, 794
- (IrregIsoProc Public Methods)*, 466, 466, 466
- (IrregIsotropicBRDF Private Data)*, 464, 465
- (IrregIsotropicBRDF Public Methods)*, 464, 465
- (KdAccelNode Methods)*, 229, 231, 231
- (KdNode Data)*, 1028, 1029
- (KdTree Declarations)*, 1028, 1029, 1031
- (KdTree Method Definitions)*, 1029, 1030, 1032
- (KdTree Private Data)*, 1029, 1029
- (KdTreeAccel Declarations)*, 228, 241
- (KdTreeAccel Local Declarations)*, 229, 235
- (KdTreeAccel Method Definitions)*, 229, 230, 233, 240
- (KdTreeAccel Private Data)*, 228, 229, 232, 234
- (KdTreeAccel Public Methods)*, 228, 245
- (Lambertian Private Data)*, 447, 447
- (Lambertian Public Methods)*, 447, 447, 447

- (LanczosSincFilter Public Methods), 400, 402, 402
- (LDSampler Declarations), 373
- (LDSampler Method Definitions), 373, 374
- (LDSampler Private Data), 373, 374
- (LDSampler Public Methods), 373, 374, 374, 377
- (Light Declarations), 606, 608, 623, 710
- (Light Interface), 606, 606, 608, 711, 941
- (Light Method Definitions), 609, 609, 631, 710, 943
- (Light Protected Data), 606, 606
- (Light Public Data), 606, 606
- (Look for texture in texture cache), 525, 525
- (Lookup nearby photons at irradiance computation point), 832, 832
- (Loop over filter support and add sample to pixel arrays), 408, 409
- (Loop over light path vertices and connect to camera vertex), 850, 851
- (Loop over light sources and sum their SH coefficients), 949, 949
- (Loop over primitives in voxel and find intersections), 206, 207
- (Loop over wavelength sample segments and add contributions), 268, 269
- (LoopSubdiv Declarations), 151
- (LoopSubdiv Inline Functions), 158
- (LoopSubdiv Local Structures), 152, 153, 155
- (LoopSubdiv Macros), 153
- (LoopSubdiv Method Definitions), 151, 160, 161, 164, 166
- (LoopSubdiv Private Data), 151, 152
- (LoopSubdiv Private Methods), 151, 163, 171
- (main program), 20
- (Make first pass through candidate points with reader lock), 892, 894
- (Make second pass through points with writer lock and update octree), 892, 895
- (Map square to (r, θ)), 667, 668
- (Map uniform random numbers to $[-1, 1]^2$), 667, 667
- (MarbleTexture Declarations), 568
- (MarbleTexture Private Data), 568, 568
- (MarbleTexture Public Methods), 568, 568, 569
- (Material Declarations), 483
- (Material Interface), 483, 483, 484
- (Material Method Definitions), 495
- (Matrix4x4 Declarations), 1021
- (Matrix4x4 Method Definitions), 1020, 1021
- (Matrix4x4 Public Methods), 1021, 1021, 1021
- (MatteMaterial Declarations), 484
- (MatteMaterial Method Definitions), 485
- (MatteMaterial Private Data), 484, 485
- (MatteMaterial Public Methods), 484, 485
- (MeasuredMaterial Declarations), 489
- (MeasuredMaterial Method Definitions), 489
- (MeasuredMaterial Private Data), 489, 489
- (Memory Allocation Functions), 1013
- (Memory Declarations), 1010, 1011, 1013, 1015, 1017
- (MemoryArena Private Data), 1015, 1015
- (MemoryArena Public Methods), 1015, 1015, 1016, 1017
- (Merge indirect photons into shared array), 812, 812
- (Merge local photon data with data in PhotonIntegrator), 806, 812
- (Metropolis Declarations), 852
- (Metropolis Local Declarations), 838, 838, 839, 840, 841, 842, 843
- (Metropolis Method Definitions), 843, 845, 847, 849, 855, 856, 857
- (MetropolisRenderer Private Data), 852, 853
- (Microfacet Private Data), 454, 454
- (Microfacet Public Methods), 454, 455
- (MicrofacetDistribution Interface), 454, 454, 695
- (MIPMap Declarations), 530, 530
- (MIPMap Method Definitions), 530, 535, 536, 539, 540, 541
- (MIPMap Private Data), 530, 530, 532, 535, 544
- (MIPMap Private Methods), 530, 532
- (MIPMap Public Methods), 530, 535
- (Mitchell Filter Declarations), 398
- (Mitchell Filter Method Definitions), 399
- (MitchellFilter Public Methods), 398, 398, 400
- (MixMaterial Declarations), 488
- (MixMaterial Method Definitions), 488
- (MixMaterial Private Data), 488, 488
- (MixMaterial Public Methods), 488, 488
- (MixTexture Declarations), 521
- (MixTexture Public Methods), 521, 521, 522
- (Modify ray for depth of field), 309, 312, 315
- (Monte Carlo Function Definitions), 367, 652, 664, 666, 667, 671, 672, 713, 731
- (Monte Carlo Inline Functions), 693, 693
- (Monte Carlo Utility Declarations), 354, 362, 366, 367, 648, 669, 670, 672
- (Normal Public Methods), 65, 66
- (Normalize filter weights for texel resampling), 533, 533
- (Normalize pixel with weight sum), 412, 413
- (Object Creation Function Definitions), 1066
- (Octree Declarations), 1023, 1023, 1026
- (Octree Method Definitions), 1024, 1027
- (Octree Private Data), 1023, 1023
- (Octree Public Methods), 1023, 1023, 1024, 1027
- (OrenNayar Private Data), 451
- (OrenNayar Public Methods), 449
- (Orient shading normal to match geometric normal), 495, 496
- (OrthoCamera Private Data), 306, 307
- (OrthographicCamera Declarations), 306
- (OrthographicCamera Definitions), 306, 309, 310
- (Otherwise add data item to octree children), 1024, 1024
- (Otherwise compute M_0 from points in leaf or recursively visit children), 911, 911
- (Parallel Declarations), 1036, 1036, 1037, 1038, 1039, 1041
- (ParamSet Declarations), 1047, 1048
- (ParamSet Methods), 1049, 1050
- (ParamSet Private Data), 1047, 1047
- (ParamSet Public Methods), 1047, 1049, 1049, 1050, 1051
- (ParamSetItem Data), 1048, 1048
- (ParamSetItem Methods), 1048
- (Parse scene from input files), 21, 21
- (Parse scene from standard input), 21, 21
- (Partition primitives into equally-sized subsets), 217, 219
- (Partition primitives into two sets and build children), 213, 215
- (Partition primitives through node's midpoint), 215
- (Partition primitives using approximate SAH), 219
- (PathIntegrator Declarations), 766
- (PathIntegrator Method Definitions), 766, 767
- (PathIntegrator Private Data), 766, 766, 766
- (PathIntegrator Public Methods), 766, 766
- (Perform projective divide), 311, 311
- (Perform trilinear interpolation at appropriate MIPMap level), 536, 537

- (Perlin Noise Data)*, 557
- (Permute LHS samples in each dimension)*, 356, 356
- (PermutedHalton Private Data)*, 367, 368
- (PermutedHalton Public Methods)*, 367, 368
- (PerspectiveCamera Declarations)*, 310
- (PerspectiveCamera Method Definitions)*, 310, 312
- (PerspectiveCamera Private Data)*, 310, 312
- (PerspectiveCamera Public Methods)*, 310, 312
- (PhotonIntegrator Local Declarations)*, 805, 815, 821, 822, 825
- (PhotonIntegrator Local Definitions)*, 822, 830, 831, 832
- (PhotonIntegrator Method Definitions)*, 803, 804, 806, 816
- (PhotonIntegrator Private Data)*, 803, 803, 812, 813, 815
- (PlanarMapping2D Public Methods)*, 518, 518
- (PlasticMaterial Declarations)*, 487
- (PlasticMaterial Method Definitions)*, 487
- (PlasticMaterial Private Data)*, 487, 487
- (PlasticMaterial Public Methods)*, 487, 487
- (Point Public Data)*, 63, 63
- (Point Public Methods)*, 63, 64, 64
- (Point sample Checkerboard2DTexture)*, 548, 548, 550
- (PointLight Declarations)*, 610
- (PointLight Method Definitions)*, 610, 611, 712, 942
- (PointLight Private Data)*, 610, 611
- (PointLight Public Methods)*, 610, 611
- (Possibly add data item to current octree node)*, 1024, 1024
- (Possibly add emitted light at path vertex)*, 767, 768
- (Possibly create area light for shape)*, 1066, 1067
- (Possibly create radiance photon at photon intersection point)*, 810, 814
- (Possibly skip virtual light shadow ray with Russian roulette)*, 780, 782
- (Possibly terminate photon path with Russian roulette)*, 810, 811
- (Possibly terminate ray marching if transmittance is small)*, 881, 882
- (Possibly terminate the path)*, 767, 769
- (Possibly terminate virtual light path with Russian roulette)*, 779, 780
- (Precompute directions ω and SH values for directions)*, 978, 978
- (Precompute filter weight table)*, 405, 407
- (Precompute information for light sampling densities)*, 776, 777
- (Precompute radiance at a subset of the photons)*, 804, 815
- (Precompute sine and cosine terms for z-axis SH rotation)*, 955, 955
- (Precompute x and y filter table offsets)*, 409, 410
- (Prepare for next level of subdivision)*, 161, 170
- (Prepare for volume integration stepping)*, 880, 881
- (Prepare temporary array pointers for low-discrepancy camera samples)*, 375, 375
- (Prepare temporary array pointers for low-discrepancy integrator samples)*, 375, 375
- (Prepare to traverse kd-tree for ray)*, 240, 241
- (Primitive Declarations)*, 185
- (Primitive Interface)*, 185, 185, 185, 186, 187
- (Primitive Method Definitions)*, 185, 186
- (Primitive Protected Data)*, 185, 185
- (Primitive Public Data)*, 185, 185
- (Process kd-tree interior node)*, 242, 242
- (Process kd-tree node's children)*, 1032, 1033
- (Process scene description)*, 20, 21
- (ProgressReporter Public Methods)*, 1006, 1006
- (Project diffuse transfer function at point to SH)*, 972, 972
- (Project InfiniteAreaLight to SH from cube map sampling)*, 944, 949
- (Project InfiniteAreaLight to SH from lat-long representation)*, 944, 945
- (Project InfiniteAreaLight to SH using Monte Carlo if visibility needed)*, 944, 944
- (Project point light source to SH)*, 942, 942
- (Project point onto projection plane and compute light)*, 617, 618
- (ProjectionLight Declarations)*, 614
- (ProjectionLight Method Definitions)*, 616, 617, 618
- (ProjectionLight Private Data)*, 614, 616, 617
- (ProjectiveCamera Protected Data)*, 305, 305, 306, 314
- (Push vertices to limit surface)*, 161, 171
- (Put far BVH node on todo stack, advance to near node)*, 225, 227
- (Put vert one-ring in Pring)*, 164, 164, 166
- (Quaternion Inline Functions)*, 93, 93
- (Quaternion Method Definitions)*, 96
- (Quaternion Public Data)*, 93
- (Quaternion Public Methods)*, 92, 93, 94
- (RadiancePhotonProcess Methods)*, 825, 825
- (Randomly accept proposed path mutation (or not))*, 857, 861
- (Randomly choose a single light to sample, light)*, 746, 747
- (Ray Public Data)*, 66, 67, 67
- (Ray Public Methods)*, 66, 68, 68
- (RayDifferential Public Data)*, 69, 69
- (RayDifferential Public Methods)*, 69, 69, 69, 70
- (Recheck candidate point and possibly add to octree)*, 895, 896
- (Record information for current path vertex)*, 843, 844
- (Recursively build BVH tree for primitives)*, 210, 211
- (Recursively initialize children nodes)*, 234, 240
- (Recursively visit children nodes to compute M_o)*, 911, 911
- (Reference Public Methods)*, 1011, 1011, 1011, 1012
- (Refine animated shape and create BVH if more than one shape created)*, 1067, 1068
- (Refine instance Primitives and create aggregate)*, 1070, 1070
- (Refine last primitive in todo list)*, 186, 187
- (Refine primitive prim if it's not intersectable)*, 206, 207
- (Refine primitives in voxel if needed)*, 206, 206
- (Reflection Declarations)*, 465, 905
- (RegularHalfangleBRDF Private Data)*, 467, 468
- (RegularHalfangleBRDF Public Methods)*, 467, 468
- (Release temporary image memory)*, 412, 413
- (Remove most distant photon from heap and add new photon)*, 822, 823
- (Renderer Declarations)*, 24
- (Renderer Interface)*, 24, 24, 25
- (RenderOptions Constructor Implementation)*, 1056, 1057, 1058, 1069
- (RenderOptions Public Data)*, 1056, 1057, 1057, 1058, 1065, 1068, 1069

- (RenderOptions Public Methods), 1056, 1072
- (Report any inconsistencies between intersections), 247, 249
- (Report sample results to Sampler, add contributions to image), 32, 33
- (Request samples for final gathering), 803, 803
- (Resample image in s direction), 531, 532
- (Resample image to power-of-two resolution), 530, 531
- (Return insideDot result if point is inside dot), 559, 560
- (Return radiance value for infinite light direction), 727, 729
- (Return uniform weight if point inside sphere), 722, 722
- (Return $x \in [0, 1)$ corresponding to sample), 649, 649
- (Reverse bits of n), 372, 372
- (RGBSpectrum Public Methods), 279, 279, 280
- (Rotate coefficients for band l about z), 955, 955
- (Rotate incident SH lighting to local coordinate frame), 981, 981
- (Round up sz to minimum machine alignment), 1016, 1016
- (Safely update $Lxyz$ and $weightSum$ even with concurrency), 410, 411
- (Sample BSDF to get new path direction), 767, 768
- (Sample BSDF with multiple importance sampling), 749, 750
- (Sample chosen $BxDF$), 706, 707
- (Sample direction for irradiance estimate ray), 795, 796
- (Sample direction for outgoing Metropolis path direction), 843, 844
- (Sample from first quadrant and remap to hemisphere to sample ω_h), 699, 699
- (Sample gather ray direction from photonNum), 826, 826
- (Sample illumination from lights to find path contribution), 767, 768
- (Sample i th direction and compute estimate for transfer coefficients), 973, 973
- (Sample light ray and apply bidirectional path tracing), 845, 846
- (Sample light source with multiple importance sampling), 749, 749
- (Sample Method Definitions), 344, 345
- (Sample new photon ray direction), 808, 810
- (Sample new ray direction and update weight for virtual light path), 778, 779
- (Sample point on lens), 315, 317
- (Sample Public Data), 343, 344
- (Sample Public Methods), 343, 344, 344, 346
- (Sample random direction from BSDF for final gather ray), 824, 824
- (Sample random direction using photons for final gather ray), 826, 826
- (Sample ray leaving light source for virtual light path), 777, 778
- (Sample sphere uniformly inside subtended cone), 720, 722
- (Sample uniformly on sphere if p is inside it), 720, 721
- (SampledSpectrum Private Data), 266, 271, 277
- (SampledSpectrum Public Methods), 266, 266, 267, 271, 272, 273, 275, 278
- (Sampler Interface), 340, 340, 341, 344
- (Sampler Method Definitions), 340, 341
- (Sampler Public Data), 340, 340
- (SamplerRenderer Declarations), 25
- (SamplerRenderer Method Definitions), 26, 27, 34, 35
- (SamplerRenderer Private Data), 25, 25, 26
- (SamplerRendererTask Declarations), 29
- (SamplerRendererTask Definitions), 30
- (SamplerRendererTask Private Data), 29, 30
- (SamplerRendererTask Public Methods), 29, 30
- (Sampling Declarations), 340, 342, 343
- (Sampling Function Definitions), 352, 352, 356, 374, 375
- (Sampling Inline Functions), 372, 372, 377
- (Sampling Local Definitions), 368
- (Scale to canonical viewing volume), 311, 312
- (ScaledBxDF Public Methods), 431, 432
- (ScaleTexture Declarations), 520
- (ScaleTexture Public Methods), 520, 521, 521
- (Scan over ellipse bound and compute quadratic equation), 541, 543
- (Scene Constructor Implementation), 24
- (Scene Declarations), 22
- (Scene Method Definitions), 24
- (Scene Public Data), 22, 23, 23, 24
- (Scene Public Methods), 22, 23, 24
- (SDEdge Comparison Function), 155, 155
- (SDEdge Constructor), 155, 155
- (SDFace Methods), 153, 159, 159, 168
- (SDVertex Constructor), 152, 152
- (See if any shape ids differ within samples), 388, 388
- (See if hit point is inside disk radii and ϕ_{\max}), 131, 132
- (See if point is visible to any element of surfacePoints), 964, 964
- (Select antialiasing method for Checkerboard2DTexture), 547, 548
- (Select initial sample from bootstrap samples), 855, 857
- (Set face to vertex pointers), 151, 154
- (Set neighbor pointers in faces), 151, 156
- (Set ray origin and direction for infinite light ray), 714, 715
- (Set up 3D DDA for ray), 202, 204
- (Shape Declarations), 108
- (Shape Interface), 108, 109, 113, 716
- (Shape Method Definitions), 108, 109, 110, 111, 113, 717
- (Shape Public Data), 108, 108, 109
- (ShapeSet Declarations), 626
- (ShapeSet Method Definitions), 723, 724
- (ShapeSet Private Data), 626, 626, 723
- (Shift dgEval du in the u direction), 495, 495
- (Shift dgEval dv in the v direction), 495, 496
- (Shift stratified image samples to pixel coordinates), 352, 353
- (Sinc Filter Declarations), 400
- (Sinc Filter Method Definitions), 402
- (SingleScatteringIntegrator Declarations), 882
- (Skip point p if not indirectly visible from camera), 963, 964
- (Solve quadratic equation for t values), 117, 118, 123, 127
- (Sort samples if unordered, use sorted for returned spectrum), 267, 267, 280
- (Spectral Data Declarations), 271, 276
- (Spectrum Declarations), 264, 266, 279
- (Spectrum Inline Functions), 265
- (Spectrum Method Definitions), 268, 277, 279, 281
- (Spectrum Utility Declarations), 266, 273, 274, 277
- (SpecularReflection Private Data), 440, 440
- (SpecularReflection Public Methods), 440, 440, 441, 703
- (SpecularTransmission Private Data), 444, 444
- (SpecularTransmission Public Methods), 444, 444, 444, 703
- (Sphere Declarations), 115

- (Sphere Method Definitions), 116, 116, 117, 123, 124, 720, 722, 1073
- (Sphere Private Data), 115, 116
- (Spherical Harmonics Declarations), 935, 935, 946, 950, 956, 981
- (Spherical Harmonics Definitions), 936, 949, 951, 953, 955, 968, 973, 976, 978
- (Spherical Harmonics Local Definitions), 936, 939, 940, 968
- (Splat current and proposed samples to Film), 857, 860
- (SpotLight Declarations), 612
- (SpotLight Method Definitions), 612, 613, 614, 713
- (SpotLight Private Data), 612, 613
- (Start recursive construction of kd-tree), 232, 233
- (Statistics Counters Probe Declarations), 1008
- (Statistics Counters Probe Definitions), 1007
- (Statistics Disabled Declarations), 1007
- (Stop following paths if not finding new points), 892, 896
- (Store candidate sample point at ray intersection if appropriate), 893, 894
- (Store primitive ids for leaf node), 230, 230
- (StratifiedSampler Declarations), 349
- (StratifiedSampler Method Definitions), 349, 350, 351
- (StratifiedSampler Private Data), 349, 349
- (StratifiedSampler Public Methods), 349, 349, 359
- (SubsurfaceMaterial Declarations), 599
- (SubsurfaceMaterial Method Definitions), 600
- (SubsurfaceMaterial Private Data), 599, 599
- (SubsurfaceOctreeNode Methods), 898, 899, 899, 901
- (SubsurfaceOctreeNode Public Data), 898, 899
- (SurfaceIntegrator Interface), 740, 741
- (SurfacePoint Data), 890, 891
- (SurfacePointsRenderer Declarations), 890
- (SurfacePointsRenderer Local Declarations), 895
- (SurfacePointsRenderer Method Definitions), 890, 892, 896
- (SurfacePointsRenderer Private Data), 889, 890
- (SurfacePointsRenderer Public Methods), 888
- (Take initial set of samples to compute b), 855, 855
- (Terminate path with RR or prepare for finding next vertex), 843, 844
- (Test cylinder intersection against clipping parameters), 127, 128
- (Test disk ϕ value against ϕ_{\max}), 132, 132
- (Test intersection against alpha texture, if present), 141, 144
- (Test sphere intersection against clipping parameters), 117, 120, 123
- (Texture Declarations), 514, 515, 516, 517, 518, 519, 533
- (Texture Inline Functions), 563
- (Texture Interface), 519, 520
- (Texture Method Definitions), 515, 515, 516, 517, 518, 519, 555, 557, 558, 561, 564
- (TextureMapping2D Interface), 514, 514
- (TextureMapping3D Interface), 519, 519
- (TextureParams Declarations), 1061
- (TextureParams Method Definitions), 1062
- (TextureParams Private Data), 1061, 1061
- (TextureParams Public Methods), 1061, 1061, 1062
- (Trace BSDF final gather ray and accumulate radiance), 824, 824
- (Trace photon-sampled final gather ray and accumulate radiance), 826, 827
- (Trace ray for bias compensation gather sample), 783, 783
- (Trace ray to sample radiance for irradiance estimate), 795, 796
- (Trace rays for specular reflection and refraction), 43, 46, 789
- (Transform Declarations), 76
- (Transform Inline Functions), 86, 86, 88
- (Transform instance's differential geometry to world space), 190, 191
- (Transform mesh vertices to world space), 137, 138
- (Transform Method Definitions), 79, 80, 81, 83, 84, 88, 89, 307, 311
- (Transform Private Data), 76, 76
- (Transform Public Methods), 76, 77, 77, 80
- (Transform Ray to object space), 117, 117, 123, 127, 131
- (Transform step function integral into CDF), 648, 649
- (TransformCache Private Data), 1066
- (TransformedPrimitive Declarations), 190
- (TransformedPrimitive Method Definitions), 190
- (TransformedPrimitive Private Data), 190, 190
- (TransformedPrimitive Public Methods), 190, 190, 191
- (TransformSet Public Methods), 1054, 1054, 1054, 1064
- (Traverse kd-tree nodes in order for ray), 240, 242
- (Triangle Filter Declarations), 396
- (Triangle Filter Method Definitions), 396
- (Triangle Private Data), 139, 139
- (Triangle Public Methods), 139, 139, 144
- (TriangleMesh Declarations), 135, 139
- (TriangleMesh Method Definitions), 137, 138, 139, 140, 141, 145, 719
- (TriangleMesh Protected Data), 135, 137
- (TriangleMesh Public Methods), 135, 138
- (Trilinearly interpolate density values to compute local density), 593, 594
- (Try to compute radiance probe contribution at i th sample point), 963, 963
- (Try to find enough BRDF samples around m within search radius), 466, 466
- (Try to generate next vertex of ray path), 843, 843
- (Update best split if this is lowest cost so far), 238, 238
- (Update BSDF matrix elements for sampled directions), 979, 979
- (Update child vertex pointer to new even vertex), 170, 170
- (Update child vertex pointer to new odd vertex), 170, 170
- (Update children f pointers for neighbor children), 169, 170
- (Update children f pointers for siblings), 169, 169
- (Update even vertex face pointers), 169, 169
- (Update f and v for next level of subdivision), 161, 161
- (Update face neighbor pointers), 169, 169
- (Update face vertex pointers), 169, 170
- (Update for rejected candidate point), 895, 895, 896
- (Update interval for i th bounding box slab), 194, 195
- (Update neighbor pointer for edgeNum), 156, 156
- (Update new mesh topology), 161, 169
- (Update parametric interval from slab intersection ts), 195, 195
- (Update pixel values with filtered sample contribution), 409, 410
- (Update ray for effect of lens), 315, 318
- (Update sample shifts), 384, 384

- (Update tHit for quadric intersection)*, 117, 123, 127, 131
- (Update transfer matrix for unoccluded direction)*, 977, 977
- (Update vertex positions and create new edge vertices)*, 161, 162
- (Update vertex positions for even vertices)*, 162, 162
- (Use BSDF to do final gathering)*, 819, 824
- (Use hierarchical integration to evaluate reflection from dipole model)*, 908, 910
- (Use n and s to compute shading tangents for triangle, ss and ts)*, 146, 147
- (Use nearby photons to do final gathering)*, 819, 826
- (UseRadianceProbes Private Data)*, 965
- (UVTexture Declarations)*, 545
- (UVTexture Public Methods)*, 545, 546
- (Vector Public Data)*, 57, 57
- (Vector Public Methods)*, 57, 58, 58, 59, 60, 62, 66
- (VisibilityTester Public Methods)*, 608, 609, 609
- (Volume Scattering Declarations)*, 585, 587, 591, 597, 600
- (Volume Scattering Definitions)*, 584, 585, 586, 588, 597, 598, 733, 913
- (Volume Scattering Local Definitions)*, 912, 912
- (VolumeGridDensity Declarations)*, 593
- (VolumeGridDensity Method Definitions)*, 593
- (VolumeGridDensity Private Data)*, 593, 593
- (VolumeGridDensity Public Methods)*, 593, 593, 594
- (VolumeIntegrator Interface)*, 876, 876
- (VolumeRegion Interface)*, 587, 587, 588
- (Voxel Declarations)*, 202
- (Voxel Public Methods)*, 202, 202, 202
- (Walk ray through voxel grid)*, 202, 205
- (WhittedIntegrator Declarations)*, 42
- (WhittedIntegrator Method Definitions)*, 42
- (WhittedIntegrator Private Data)*, 42, 42
- (WindyTexture Declarations)*, 567
- (WindyTexture Private Data)*, 567, 567
- (WindyTexture Public Methods)*, 567, 567, 567
- (Write RGB image)*, 412, 413