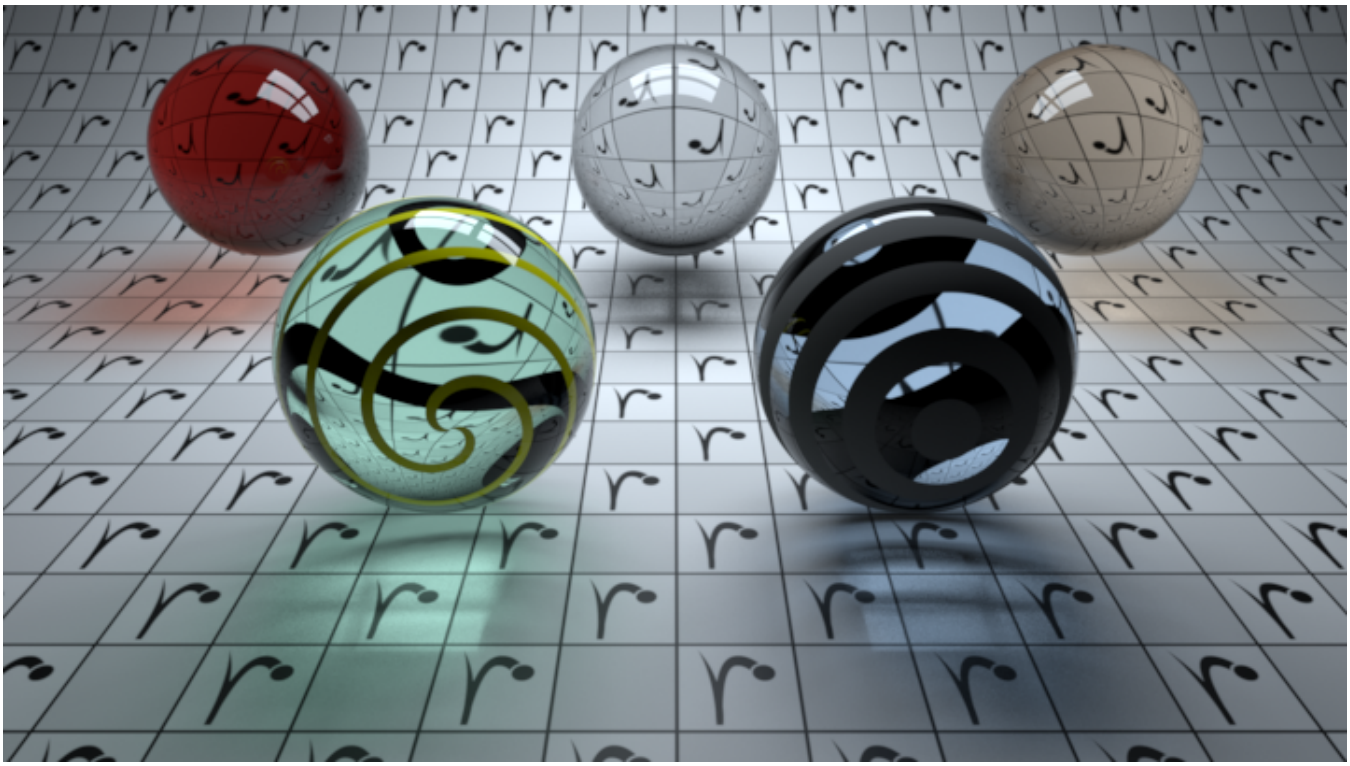
PxrPathTracer



PxrPathTracer is a core final-quality integrator in RenderMan. It implements the forward path tracing algorithm, which excels in outdoor, highly specular scenes. The simplicity of the algorithm generally makes it easy to use and to implement. Shortcomings may include slow convergence speeds, especially for scenes with significant caustic illumination (see the *clampLuminance* and *allowCaustics* parameters, below, for more information).

For direct illumination, the numLightSamples and numBxdfSamples parameters determine the number of light vs. Bxdf (material) direct illumination samples, and the resulting direct illumination contributions are combined using multiple importance sampling.

For indirect illumination, the sampleMode parameter controls the strategy for how indirect ray counts are apportioned between diffuse, glossy/specular, subsurface, and refraction/transmission Bxdf lobes.

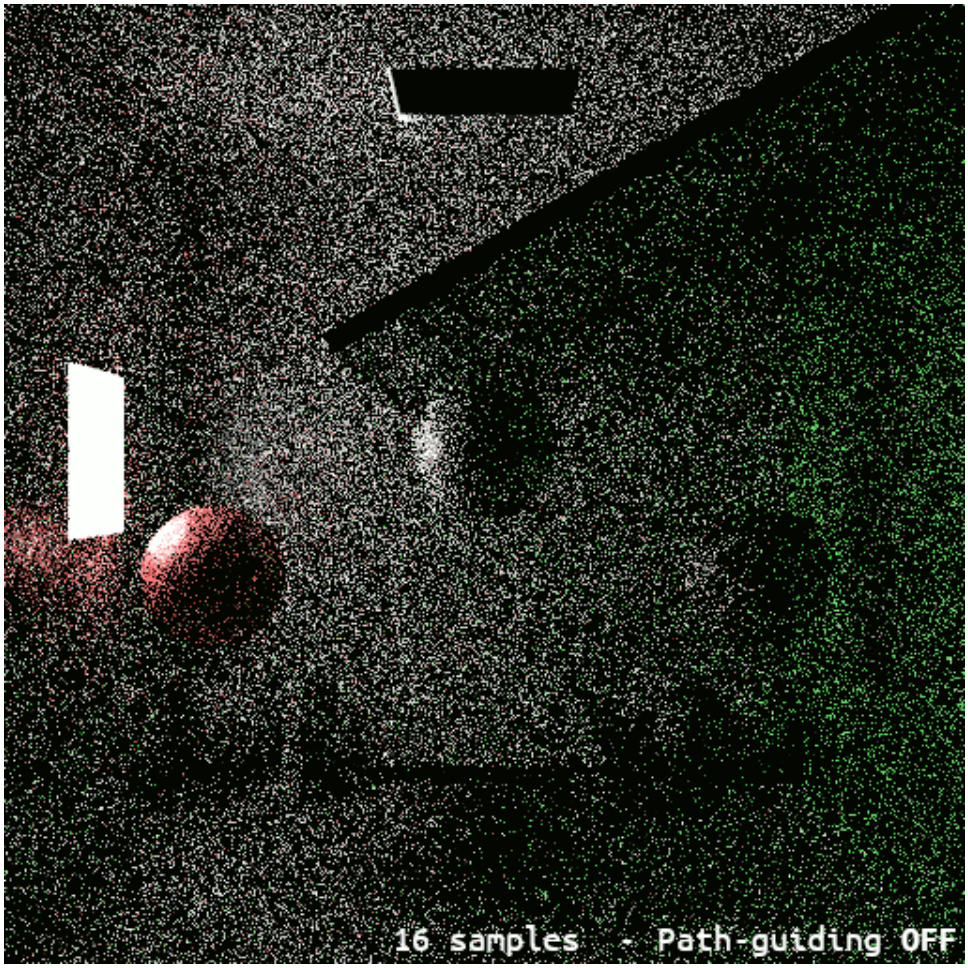
When the sampleMode parameter is set to a value of "bxdf" (the default value), then the number of indirect illumination rays spawned at each camera hit point is controlled by the value of the numIndirectSamples parameter, and the implementation of the Bxdf at each camera hit point controls the probability of generating indirect rays that are sampled from the diffuse vs. glossy vs. specular lobes of the Bxdf (or, in other words, the Bxdf controls how many indirect diffuse vs. glossy vs. specular rays are spawned at each of the camera hit points).

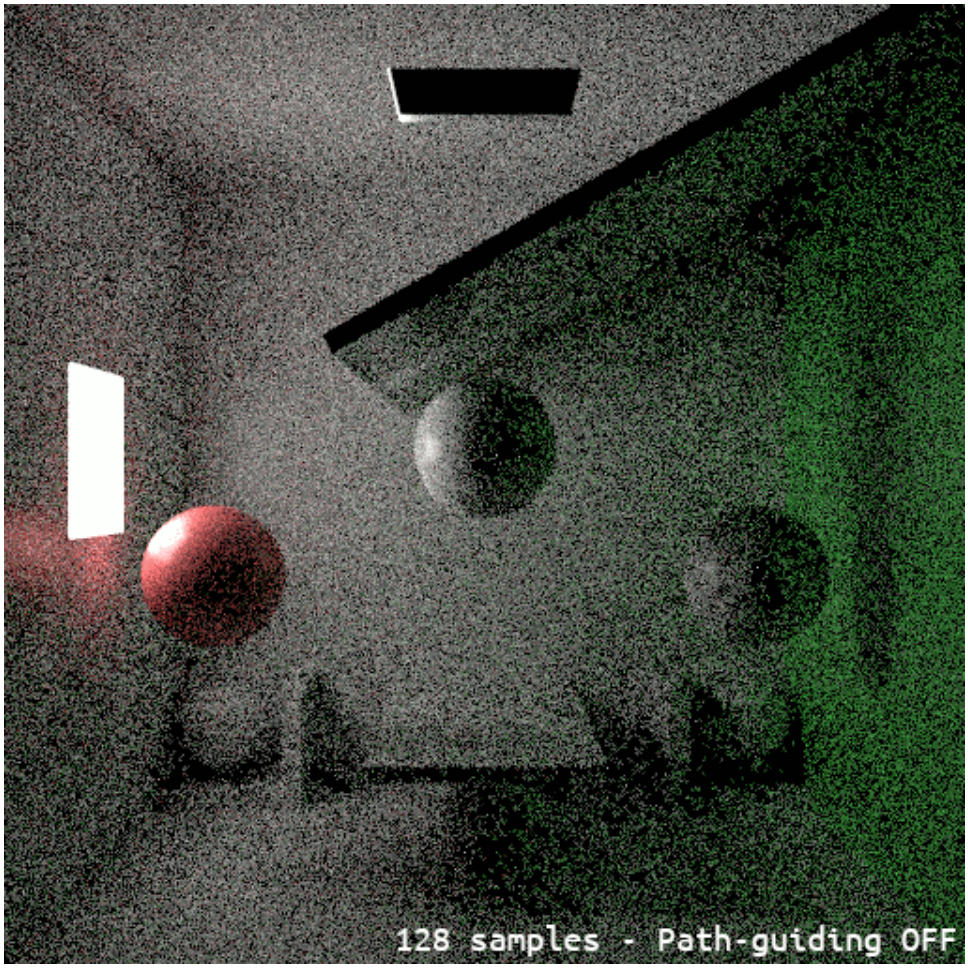
When the sampleMode parameter is set to a value of "manual", then the value of the numIndirectSamples parameter is ignored and the user is responsible for manually controlling the total number of indirect rays per lobe of the Bxdf. Specifically, when sampleMode is set to "manual" then the numDiffuseSamples parameter explicitly controls the number of indirect diffuse rays that should be spawned at each camera hit point, the numSpecularSamples parameter controls the number of indirect glossy/specular rays that should be created at each camera hit point, the value of the numSubsurfaceSamples parameter is used to determine the number subsurface rays that are launched from each camera hit point, and the value of the numRefractionSamples parameter is equal to the number of refraction/transmission rays that are created at each camera hit point.

Parameters

Parameter	Description
maxIndirect Bounces	Controls the absolute upper bound on the maximum ray depth. For example, a value of 0 for this parameter will allow direct illumination only, while a value of 4 will permit up to 4 bounces of global illumination. Default value is 10.

sampleMode	Controls the strategy for how indirect ray counts are apportioned between diffuse vs. glossy vs. specular Bxdf lobes. Valid values for this parameter are: "bxdf" or "manual". If "sampleMode" is set to "bxdf", then the Bxdf is responsible for balancing the number of indirect rays that will be spawned at each camera hit point between the diffuse vs. glossy vs. specular lobes, and the total number of indirect rays spawned at each camera hit point will be controlled by the numIndirectSamples parameter. If "sampleMode" is set to "manual" the user is responsible for explicitly specifying the number of indirect rays to spawn per Bxdf lobe at each camera hit point via the numDiffuseSamples, numSpecularSamples, numSubsurfaceSamples, and numRefractionSamples parameters. The default is "bxdf".
numLightSamples	Controls the number of light samples for direct illumination per camera hit point. The default is 1. Small numbers may improve performance of interactive and progressive updates. Larger numbers may appear to converge better while being slower to update.
numBxdfSamples	Controls the number of Bxdf samples for direct illumination per camera hit point. The default is 1. Small numbers may improve performance of interactive and progressive updates. Larger numbers may appear to converge better while being slower to update.
numIndirectSamples	When sampleMode is set to Bxdf this parameter controls the total number of indirect rays to spawn per camera hit point. When sampleMode is set to manual the value of this parameter is ignored. The default is 1.
numDiffuseSamples	When sampleMode is set to manual, this parameter controls the number of indirect diffuse reflection rays to spawn per camera hit point. Ignored when sampleMode is set to Bxdf. The default is 1.
numSpecularSamples	When sampleMode is set to manual, this parameter controls the number of indirect specular/glossy reflection rays to spawn per camera hit point. Ignored when sampleMode is set to Bxdf. The default is 1.
numSubsurfaceSamples	When sampleMode is set to manual, this parameter controls the number of subsurface rays to spawn per camera hit point. Ignored when sampleMode is set to Bxdf. The default is 1.
numRefractionSamples	When sampleMode is set to manual, this parameter controls the number of refraction/transmission rays to spawn per camera hit point. Ignored when sampleMode is set to "bxdf". The default is 1.
rouletteDepth	Controls the ray depth to begin performing Russian Roulette. Russian Roulette may terminate rays after this depth especially if they contribute little to the result. The default is 4.
rouletteThreshold	Controls the path throughput threshold below which to perform Russian Roulette. This controls at which point the contribution of the ray is unimportant enough to terminate the ray. The default is 0.2.
clampDepth	If a value for the clampLuminance parameter is specified, then clampDepth controls the ray depth at which to begin clamping based on the per-ray luminance. For example, setting this parameter to 2 and also specifying a value of 4 for clampLuminance will ensure that the luminance of each ray's contribution is no more than 4 for all indirect illumination, without affecting or clamping the direct illumination. The default is 2.
clampLuminance	By default the PxrPathTracer integrator clamps the luminance of each per-ray contribution to be at most 10.0. However, it is possible to change this behavior by specifying a different value for the clampLuminance parameter. Specifying a relatively low value for the clampLuminance parameter (for example, between 2 and 20) can greatly speed up convergence and, in many cases, will make PxrPathTracer converge more quickly than the more sophisticated PxrVCM or PxrUnified integrators. In some cases, indirect illumination lights paths may be noticeably dimmer due to clamping; this may be an acceptable trade-off in certain cases. Setting this parameter to a very large number (such as 1e30) will effectively disable all clamping. The default is 10.0.
allowCaustics	Controls whether illumination from caustic light paths (that is, specular illumination onto diffuse surfaces) is allowed or disallowed. The default is 0 (off).
accumOpacity	Controls whether or not the path tracer will keep track of accumulated opacity along the path. The default is off. If you plan to render semi-opaque objects and care about alpha values for compositing, you might want to turn this on. This feature does not function when using "manual" sample mode. We recommend the default of BxDF sampling.
risPathGuiding	Uses learnt direct lighting results to guide directions for indirect path sampling. Can only be used if light selection learning is on and numIndirectSamples is set to 1. The default is off. In the image below, a PxrDirectionalLight outside of the room lights only a small portion the left wall through the top window: most of the illumination is indirect. In that case, path-guiding makes a very significant difference. Model by Onouris.


16 samples - Path-guiding OFF

	
maxNonStochasticOpacityEvents	This defines the depth at which true opacity is computed. Depths beyond this number are calculated as stochastic presence. Lower values may improve performance at the cost of increased noise. This only applies to camera/primary rays. All others are computed as stochastic presence.
maxContinuationLength	This is an advanced knob for rendering overlapping volumes. Similar to maxPathLength, this controls the upper bound on the maximum ray depth including continuation rays. A negative value will cause the path tracer to use an empirical heuristic to bound the ray depth. A positive value will override this heuristic and cause the path tracer to use the parameter value directly as the maximum continuation ray depth. Continuation rays are normally produced by volumes when no density exists within subsections of them. If there are many such zero density subsections overlapping then it is possible to quickly hit the maximum continuation depth. If you are rendering a scene with highly overlapping volumes, you might want to adjust this parameter to be generous enough for the ray to exit the scene.

Standard AOVs

On top of regular LPE-based AOVs, this integrator outputs a number of standard AOVs typically used by compositors.

Declaration	Contents	Channels
color __Pworld	P in world-space	__Pworld.r : x component __Pworld.g : y component __Pworld.b : z component
color __Nworld	Nn in world-space	__Nworld.r : x component __Nworld.g : y component __Nworld.b : z component
color __depth	Multi-purpose AOV	__depth.r : depth from camera in world-space __depth.g : height in world-space __depth.b : geometric facing ratio : abs(Nn.V)

color __st	Texture coords	__st.x : s __st.y : t __st.z : 0.0
color __Pref	Reference Position primvar (if available)	__Pref.r : x component __Pref.g : y component __Pref.b : z component
color __Nref	Reference Normal primvar (if available)	__Nref.r : x component __Nref.g : y component __Nref.b : z component
color __WPref	Reference World Position primvar (if available)	__WPref.r : x component __WPref.g : y component __WPref.b : z component
color __WNref	Reference World Normal primvar (if available)	__WNref.r : x component __WNref.g : y component __WNref.b : z component