Renderer Options

- Common
- Shading
- Dicing
- Display
- Statistics
- Configuration
- RIB
- User

Common

Name	Type	Default	Description
hider: adaptall	*int*	0	Enable adaptive sampling of all AOVs. When enabled, all channels (beauty pass + AOVs) are considered when deciding if a pixel is converging. The default (off) will ignore the AOVs and only the beauty pass is considered for pixel convergence. So turning off "adaptall" may speed up your render if some of your AOVs never reach the specified variance threshold. Value range: true/false.
hider: darkfalloff	*float*	0.025	De-prioritize low luminance samples in the adaptive sampler. Use this parameter to avoid oversampling dark areas while maintaining overall image quality. Higher values cull more samples, speeding up render time while increasing noise in darker areas. Value range: 0 to 1.
hider: extreme motiondof	*int*	0	Enable improved sampling accuracy. In some cases where a large amount of motion blur is used with depth of field, sampling artifacts could cause an undesirable result. This flag causes a slower but more accurate sampling method to be used to alleviate the artifacts. Due to the performance degradations of this sampling method, this flag should only be used when necessary. Value range: true/false.
hider: incremental	*int*	0	Enables the progressive display of results (rather than bucketed results). Value range: true/false.
hider: maxsamples	*int*	0	An explicit limit for the number of samples. If maxsamples is equal to its default then the maximum number of samples will be set equal to 64. Note that minsamples defaults to the square root of the maxsamples setting. Value range: 0 or more.
hider: minextrasamples	*int*	-1	This is the minimum number of extra samples that a pixel should continue shooting with after a sample fails to meet its convergence test with the adaptive sampler. Defaults to matching minsamples.
hider: minsamples	*int*	-1	Specifies the minimum number of samples for adaptive sampling. The default value is the square root of maxsamples.
hider: pixelfiltermode	*string*	weighted	Specifies how pixel filtering of camera samples is performed. "Weighted" is the standard method and blends together samples from neighboring pixels. "Importance" warps the samples in screen space around the pixel center and does not share samples between pixels. It may be noisier, but is needed by some postprocessing algorithms. It is also needed to produce variance AOVs. Supported values: weighted, importance.
hider: samplemotion	*int*	1	This option toggles the sampling of motion blurred micropolygon in the hider, and is thus only meaningful if motion blur is being used in your scene. By default, this hider option is enabled (set to 1), meaning that micropolygons will be blurred and sampled in the hider. When disabled (set to 0), micropolygons will no longer be blurred before sampling; however, the dPdtime variable will still be computed for all objects. This variable can be output into a separate image using an arbitrary output variable (aov) and used to perform motion blur as a postprocess. Value range: true/false.
hider: sampleoffset	*int*	0	This allows several images to be rendered in parallel (with different sampleoffset values) and then combined. With non-adaptive sampling: Let's say you render four images with 256 samples each, with sampleoffsets 0, 256, 512, and 768. If you combine those four images, you'll get exactly the same image as if you had rendered a single image with 1024 samples. With adaptive sampling: Let's say you again render four images, each with "maxsamples" 256, with sampleoffsets 0, 256, 512, and 768. Let's say that due to adaptive sampling, some given pixel only gets 64 pixel samples in each of the four images. Then the combined image has been rendered with sample numbers 0-63, 256-319, 512-575, and 768-831. Due to the stratification of the samples, this is not quite as good as if you had rendered a single image with 256 consecutive samples. However, it is still better than rendering a single image with only 64 samples.
hider: type	*string*	raytrace	Render mode. Supported values: raytrace, bake.
Ri: Frame	*int*	0	Frame number.
Ri: PixelVariance	*float*	0.001	Controls adaptive sampling. This sets the upper bound on the acceptable estimated variance of the pixel values from the true pixel values. Value range: 0 to 1.
Ri: Shutter	*float[2]*	0 0	Scene motion shutter.
Shutter: offset	*float*	0	Offset shutter open/close.

Shading

Name	Type	Default	Description
lighting: selection learning scheme	*int*	1	Employs a machine learning algorithm to more accurately importance sample lights, taking into account occlusion and light filters. Value range: true/false.
trace: bvhcompression	*float*	0	Provides a hint to the system to favor memory utilization versus speed when constructing the ray accelerator. Valid values are in the range [0, 1], where the higher the values the more compact the memory representation. The control behaves almost linearly with some discontinuities in the compression rate as different strategies are activated. The result obtained is likely to be scene dependent. As a rule of thumb, leave this control to 0 unless you need to reduce memory utilization in order to render a given scene or sequence. When tuning, set the control to 1 at first to see how much memory can be saved through compression and what time overhead is generated, then dial the number back to a lower value (if possible) and reach the desired balance between speed and memory usage. Value range: 0 to 1.
trace: worldoffset	*float[3]*	0 0 0	Used as the origin when the worldorigin option is set to "worldoffset".
trace: worldorigin	*string*	camera	Set a new world origin to give better floating point precision. Supported values: world, worldoffset, camera.

Dicing

Name	Type	Default	Description
curve:minwidth	*float*	0	Minimum curve width in raster coordinates. Value range: 0 or more.
dice:offscreenmultiplier	*float*	1	Dicing rate multiplier for objects outside the viewing frustum. Value range: 0 or more.

Display

Name	Type	Default	Description
bucket: order	*string*	horizontal	Render bucket order. Supported values: horizontal, vertical, zigzag-x, zigzag-y, spacefill, random, spiral, circle.
bucket: orderorigin	*int[2]*	-1 -1	First bucket for "spiral" and "circle" order. Image center is [-1 -1].
checkpoint:asfinal	*int*	0	When enabled, this keeps the extra channels and the checkpoint tag when writing the final image for the render. The final image will be identical to any other checkpoint, rather than a slimmed down image. This means that once your image has reached the quality you've set and it completes, it can always be restarted by the user later. Value range: true/false.
checkpoint: command	*string*	*empty*	System command to be called after checkpoint. If system calls are enabled, then after a checkpoint has been written, the specified command will be called. This is synchronous; the rendering threads are quiescent while this runs and will not resume again until the process returns, avoiding possible race conditions if the command takes a token %i will be replaced with the current increment, zero-padded to 5 digits. The token %e will be replaced with the elapsed time in seconds, zero-padded to 6 digits. The token %r will be replaced with the reason for this update to the checkpoint files (either completely `finished`, `exiting` early due to exitat option, or a normal `checkpoint`). Literal % characters may be inserted with %%.
checkpoint:exitat	*string*	*empty*	How long render should run before writing a final checkpoint and terminating. Value can also be a string such as "10m" or "0.5h". Set to 0 or empty to disable.
checkpoint: interval	*string*	*empty*	Write framebuffers to disk every N progressive steps. If value is positive the units will be interpreted as seconds. If value is negative its units will be interpreted as increments. Value can also be a string such as "10m" or "0.5h". Setting option to 0 or empty will disable checkpoints.
checkpoint: keepfiles	*int*	0	If set to true then prman will write to a new, numbered image for each checkpoint. Note: This will disable the final write of a checkpoint file which means the render cannot be recovered using the 'recover' option. Value range: true/false.
Ri: CropWindow	*float[4]*	0 1 0 1	Crop window [left right top bottom] in NDC space.
Ri: FormatPixelAspectRatio	*float*	1	Output image pixel aspect ratio. Value range: 0 to 1.
Ri: FormatResolution	*int[2]*	640 480	Output image resolution.
Ri: FrameAspectRatio	*float*	-1	Output image aspect ratio.

Ri: Screen Window	*float[4]*	0 0 0 0	Screen window [left right top bottom] in screen space.

Statistics

Name	Type	Default	Description
shade:debug	*int*	0	Shading debug level. Value range: true/false.
statistics: displace_ratios	*float[2]*	0.1 1	Suppress reporting of displacements that, when divided by the max displacement, fall in the specified range.
statistics:filename	*string*	*empty*	File name for summary statistics reported in plain text.
statistics: iotracefilename	*string*	*empty*	
statistics:level	*int*	0	Enable statistics reporting. Value range: true/false.
statistics: maxdispwarnings	*int*	100	Modify the maximum reported displacement issues. If the value is set to 0, then all displacements issues are reported. Value range: 0 or more.
statistics:shaderprofile	*string*	*empty*	
statistics:stylesheet	*string*	*empty*	Stylesheet for XML detailed statistics.
statistics: texturestatslevel	*int*	0	Enable texture statistics reporting. Value range: true/false.
statistics: trackassetobjects	*int*	0	
statistics:xmlfilename	*string*	*empty*	File name for detailed statistics reported as XML.

Configuration

Name	Type	Default	Description
limits: brickmemory	*int*	2097152	Brickmap cache size in kB. Value range: 0 or more.
limits: bucketsize	*int[2]*	16 16	Size of render buckets in pixels.
limits: deepshadowerror	*float*	0.01	Control lossy compression scheme to reduce file size of deep output. Setting to a high value will result in lower numbers of samples stored for each pixel. Value range: 0 to 1.
limits: geocachememory	*int*	2097152	Geometry cache size in kB. Value range: 0 or more.
limits: gridsize	*int*	289	Value range: 1 or more.
limits: matrixcachememory	*int*	0	Matrix cache in kB. Value range: 0 or more.
limits: nurbcurvatureredicefactor	*int*	3	NURBs max curvature multiplier. Value range: 0 or more.
limits: opacitycachememory	*int*	1048576	Opacity cache in kB. Value range: 0 or more.
limits: othreshold	*float[3]*	0.99609375 0.99609375 0.99609375	Opacity Culling: When rendering scenes with a large number of semi-transparent layered objects (e.g. hair), the opacity culling threshold can be set for a significant time and memory savings. Essentially, a stack of visible points whose accumulated opacity is greater (in each channel) than the specified limit will be considered fully opaque by the hider, and objects behind the stack will be culled.
limits: proceduralbakingclumpsize	*int*	0	Value range: 0 or more.
limits: ptexturemaxfiles	*int*	128	Value range: 1 or more.

limits: ptexturememory	*int*	32768	PTex cache size in kB. Value range: 0 or more.
limits: rendermemory	*float*	0	Memory limit as percentage of machine memory. Value range: 0 to 1.
limits: rendertime	*int*	0	Maximum render time in minutes. Value range: 0 or more.
limits: shadesize	*int*	289	The maximum number of vertices, not grid. Value range: 1 or more.
limits: texturememory	*int*	2097152	Texture cache size in kB. Value range: 0 or more.
limits: texturepert hreadmemoryratio	*float*	0.5	
limits: threads	*int*	0	Set the number of threads that the renderer uses. If a value of 0 is specified then all CPUs will be utilized for rendering. If a negative value '-N' is specified then prman will use all but N CPUs for rendering.
searchpath:archive	*string*	*empty*	Search path for Rib archives and DelayedReadArchive.
searchpath:dirmap	*string*	*empty*	Search path remapping.
searchpath:display	*string*	*empty*	Search path for display plug-ins.
searchpath: procedural	*string*	*empty*	Search path for DynamicLoad procedural plug-ins.
searchpath:rifilter	*string*	*empty*	Search path for Rif plug-ins.
searchpath:rixplugin	*string*	*empty*	Search path for shading plug-ins.
searchpath:shader	*string*	*empty*	Search path for OSL pattern plug-ins.
searchpath:texture	*string*	*empty*	Search path for texture files.

RIB

Name	Type	Default	Description
ribparse:varsubst	*string*	*empty*	RIB parse only.
scenedump:filename	*string*	*empty*	Path to dump RIB scene for rendermode "catrib".

User

Name	Type	Default	Description
osl:batched	*int*	1	Allows batched execution of OSL patterns to improve performance, this in on by default, see more in the OSL page
osl:verbose	*int*	4	The verbosity level of OSL output, see more in the OSL page
osl:statisticslevel	*int*	0	The statistics level of detail, see more in the OSL page