Adaptive and Progressive Sampling

Arnold has the capability of adapting the sampling rate of each pixel when the `enable_adaptive_sampling` render option is enabled, allowing it to dedicate a greater number of camera samples (and thus also a greater amount of render time) to the pixels that show a greater variation in their sample values. When used, all pixels will receive a sampling rate of at least `AA_samples`, but no more than `AA_samples_max`. The adaptive sampler's sensitivity to noise may be controlled through the `adaptive_threshold` render option, where lower threshold values will apply higher sampling rates to a greater number of pixels.

Adaptive sampling is good for situations where small areas of the image have high levels of noise that require an impractical number of samples to clear up. For example scenes with bright, motion-blurred speculars, DOF, buzzing rim lights, or scenes with the hair shader.

Adaptive/Progressive and IPR

In Arnold plugins that perform a series of render passes at lower AA settings for the purpose of previewing the output before rendering at the final sampling rates, `enable_adaptive_sampling` should be DISABLED for all but the final render pass to enhance performance.

Likewise, when the `progressive_render` render mode is used, for best performance plugins may skip performing preview renders with AA settings of 1 or higher since the progressive rendering mode will
already display all of the intermediate AA sampling steps from 1 to the user-selected setting to the render view drivers.

**Adaptive Sampler and Custom Filters**

Adaptive sampling consists in placing samples in a non-uniform fashion over the rendered frame. Because of this, some pixels will have a higher density of samples than others, and this sample density must be taken into account by weighted-average filters otherwise the sample values of high-density regions will become over-represented and those from low-density regions will become under-represented in the filtered result.

Each sample's sampling density can be obtained through an AiAOVSAMPLEIteratorGetInvDensity() function call in each iteration of a sample filtering loop. The scalar return value can be used to properly adjust each sample's value in the weighted average filter through a multiplication.

The AA_inv_density AOV can help to visualize the sample density with *Adaptive Sampling*. Use it with a *Heatmap* filter.

Max. Camera (AA)

Sets the maximum amount of supersampling. It controls the per-pixel maximum sampling rate and is equivalent to the units used by AA_samples. Adaptive sampling is enabled when AA_samples_max >
AA_samples and AA_samples >= 2. Scenes with a large amount of depth of field or motion blur may require higher Max. Camera (AA) values. This parameter can also help with 'buzzing' speculars and hair shading as well.

Setting AA_samples to 1 or lower does not yet allow Arnold to calculate the measures needed for adaptive thresholding, and setting AA_samples_max to AA_samples or lower does not leave a margin for any adaptive AA samples.

Adaptive Threshold

The threshold which triggers/terminates adaptive-AA. This value controls how sensitive to noise the adaptive sampling algorithm gets. Lower numbers will detect more noise. The default value (0.05) should work well for most scenes.
Progressive Render

Completes the render call in multiple passes. During each of the intermediate passes, drivers that do not output to a file will be invoked after each tile has completed, which allows for display drivers to show a result whose noise progressively converges towards the result at the final Camera (AA) sample settings.

Note that this option slows down render times and increases memory usage, which is why it’s only supported during interactive renders for fast preview.