Anti-aliasing Settings

Sample Clamping

Adaptive Settings

Progressive Sampling

Filtering

AA_samples

Supersampling control over the number of rays per pixel that will be traced from the camera. The higher the number of samples, the better the anti-aliasing quality, and the longer the render times. The exact number of rays per pixel is the square of this value. For example, a Camera (AA) samples value of 3 means 3x3 = 9-pixel samples. In practice, you may consider using a value of 4 for medium quality, 8 for high quality, and (rarely) 16 for super-high quality. This control acts as a global multiplier of all the different rays, multiplying the number of Diffuse and Specular rays. Motion blur and depth of field quality can only be improved by increasing Camera (AA) samples.
Sample Clamping

**indirect_sample_clamp**

The threshold to clamp away fireflies from indirect light samples and reduce noise. This works similar to AA_sample_clamp but preserves specular highlights from direct lighting. Lower values result in more aggressive noise reduction, possibly at the expense of dynamic range.
This can also help with diffuse scenes when a light is very close to the ceiling/walls.

**enable_AA_sample_clamp**

Enable/disable clamping.

**AA_sample_clamp**

If enabled, this control will clamp pixel samples to this specified maximum value. This can make it easier to anti-alias certain high-dynamic range effects such as bright motion blurred specular streaks (at the cost of reduced contrast). What is clamped is the resulting output value of the renderer computations of each pixel sample. Example: If the Max Clamp Value is set to 2, no pixels will have an RGB value greater than 2.0, 2.0, 2.0.

10 (default value) 1
Adaptive Settings

**enable_adaptive_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.
AA_samples_max

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 Sampling

enable_progressive_render

Enables/disables Progressive Rendering.
enableBlockyProgressive

All but the final pass sample settings will be automatically adjusted up or down to achieve a target frames-per-second display rate.

AA_progressive_samples_min

Preview and live rendering can be customized so that rendering isn't more blocky than desired when modifying the scene.

bucketCorners

Bucket corners can be turned disabled or enabled (default). They will not be displayed until a bucket's computation time has passed to avoid visual clutter.

Filtering

AA_filter

The filter type used for averaging individual subpixel samples into a final pixel color. Most of these filters are available for legacy purposes only. More details can be found on Wikipedia.

We recommend using the Gaussian (width 2.0) or Blackman-Harris filters.

The complete list of filters is:

- Blackman-Harris
- Box
- Catrom
- Closest
- Difference
- Farthest
- Gaussian
- Heatmap
- Mitnet
- Sinc
- Triangle
- Variance
- Contour
- Denoise Optix

AA_filter_width

The width of the subpixel sample averaging filter, in pixels. For circularly symmetric filters, this is the diameter of the filter's support.
This example demonstrates the effect filter width can have on moire effects in your images. Increasing the filter width setting helps to reduce the moire effect.

This is more noticeable on the left side and top of the cube.