

Image Sampler

This page provides information on the Image Sampler rollout under the [Sampler tab](#) in V-Ray's [Render Settings](#).

Overview

In V-Ray, an image sampler refers to an algorithm for calculating a pixel's color based on the colors within and around it.

Each pixel in a rendering can have only one color. To get the color of a pixel, V-Ray calculates it based on the object's material, direct light striking the object, and indirect lighting in the scene. But within a single pixel there might be multiple colors, which may come from multiple objects whose edge intersect at the same pixel, or even difference in brightness on the same object due to changes in object shape or falloff and/or shadowing of light sources.

To determine the right color for such a pixel, V-Ray looks at (or *samples*) colors from different parts of the pixel itself as well as the pixels around it. This process is called *image sampling*. Different image sampling methods are discussed below.

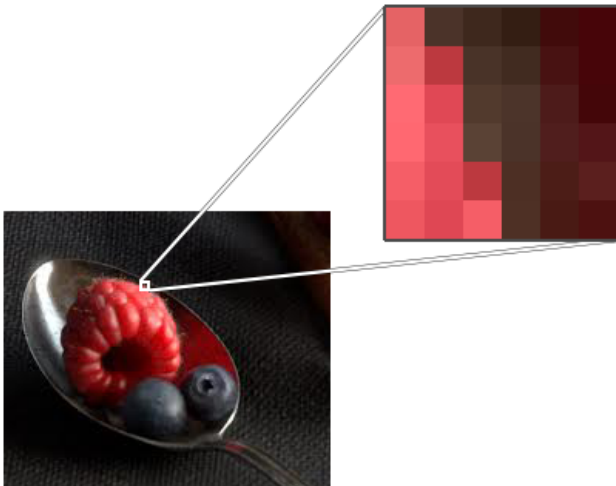


Image courtesy of Tuna Unalan

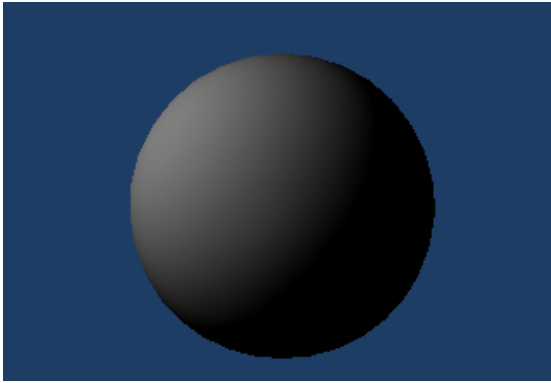
Multiple colors within a single pixel. What color should the pixel be?

UI Paths

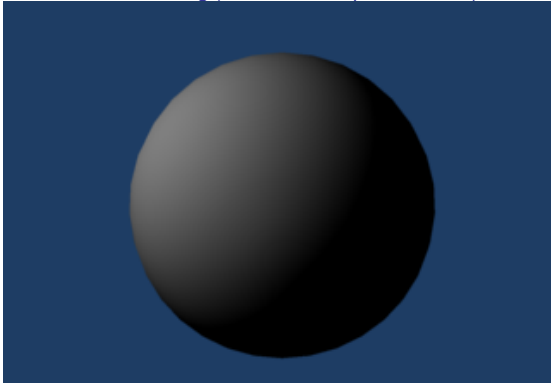
||Properties editor|| > **Render** > **Sampler** tab > **Image Sampler** rollout

What is Anti-aliasing?

The following example shows the basic difference between an image with anti-aliasing, and one without:

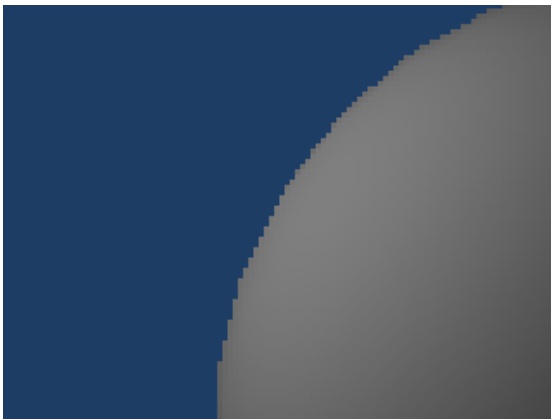


No antialiasing (Fixed rate sampler, 1 subdiv)



Antialiasing on (Adaptive subdivision sampler, rate $-1/2$)

The left images are jagged around the edges of the sphere, while the right are smooth. Here are close-ups of the two images:



No antialiasing (Fixed rate sampler, 1 subdiv)

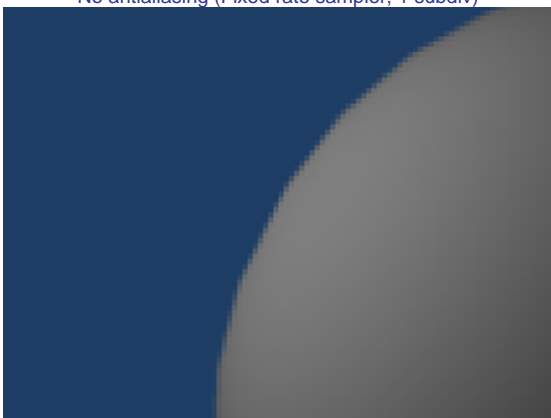


Image Sampler

Type – Specifies the image sampler type:

Fixed – Takes the same number of samples per pixel;

Adaptive DMC – Takes a variable number of samples per pixel depending on the difference in the intensity of the pixels;

Adaptive subdivision – Divides the image into an adaptive grid-like structure and refines it depending on the difference in pixel intensity;

Progressive – Progressively samples the entire image.

Min shading rate – Controls the number of primary rays shot for AA (anti-aliasing) versus secondary rays for other effects like glossy reflections, GI, area shadows, etc. Higher values mean that less time is spent on AA, and more effort is put in the sampling of shading effects. This approach is especially useful with the Progressive image sampler.

Min subdivs – Controls the minimum number of samples that each pixel in the image receives. The actual number of the samples is the square of the subdivs.

Max Subdivs – Controls the maximum number of samples that each pixel in the image receives. The actual number of the samples is the square of the subdivs.

Noise threshold – The desired noise level in the image. If this is 0.0, the entire image is sampled uniformly until either the **Max. subdivs** value is reached or the **Render time** limit is reached.

Max render time in min – The maximum render time in minutes. When this number of minutes is reached, the renderer stops. This is the render time for the whole frame; it includes any GI prepasses like light cache, irradiance map, etc. If this is 0.0, the render is not limited in time.

Ray bundle size – Useful for distributed rendering in order to control the size of the chunk of work that is handed to each machine. When using distributed rendering, higher values may help to utilize CPUs on the render servers better.

Filter Type – Specifies the filter type to be used for anti-aliasing. V-Ray supplies eight types of Anti-aliasing filters: Box, Area, Triangle, Lanczos, Sinc, CatmullRom, Gaussian and Cook Variable. Each has advantages and disadvantages, which make them useful for different tasks.

Size – Determines the size of the filter in pixels. Higher values yield blurrier results. *For more information, see the [Anti-aliasing Filters example](#) and the [Anti-aliasing Filters and Moire Effects example](#) below.*

Avoid using the Progressive sampler with sharpening image filters (Catmull-Rom, Mitchell-Netravali). This might slow down the rendering because additional image samples will be required to resolve sharpening filters properly. In such a case, V-Ray will print a warning in the V-Ray messages window.

Unlike the other bucket image samplers, the Progressive sampler doesn't have a mechanism to recover if a DR server goes offline during rendering before sending back the results from its calculations. The render process may hang indefinitely waiting for the missing data to arrive. This will be corrected in a future release.

Fixed Sampler

This is the simplest image sampler. It takes a fixed number of samples for each pixel.

Subdivs - Determines the number of samples per pixel. When this is set to 1, one sample at the center of each pixel is taken. If this is greater than 1, the samples are distributed within the pixel. The actual number of pixels is the square of this parameter (e.g. 4 subdivs produce 16 samples per pixel).

Adaptive DMC Sampler

This sampler makes a variable number of samples per pixel based on the difference in intensity between the pixel and its neighbors. This is the preferred sampler for images with lots of small details (like V-RayFur, for example) and/or blurry effects (DOF, motion blur, glossy reflections etc). It also takes up less RAM than the *Adaptive Subdivision* sampler.

Min subdivs - Determines the initial (minimum) number of samples taken for each pixel. You will rarely need to set this to more than 1, except if you have very thin lines that are not captured correctly, or fast moving objects if you use motion blur. The actual number of pixels is the square of this number (e.g. 4 subdivs produce 16 samples per pixel).

Max subdivs - Determines the maximum number of samples for a pixel. The actual maximum number of samples is the square of this number (e.g. 4 subdivs produces a maximum of 16 samples). Note that V-Ray may take less than the maximum number of samples, if the difference in intensity of the neighboring pixels is small enough.

Threshold - The threshold that will be used to determine if a pixel needs more samples. This is ignored if the Use DMC sampler threshold option is on.

Use DMC sampler threshold - When this is on, V-Ray will use the threshold specified in the [DMC sampler](#) to determine if more samples are needed for a pixel. When this is off, the Color threshold parameter will be used instead.

Adaptive Subdivision Sampler

This is an advanced image sampler capable of undersampling (taking less than one sample per pixel). In the absence of blurry effects (direct GI, DOF, glossy reflection/refraction, etc.) this is the best preferred image sampler in V-Ray. On average, it takes fewer samples (and thus less time) to achieve the same image quality as the other image samplers. However, with detailed textures and/or blurry effects, it can be slower and produce worse results than the other two methods.

Min rate - Controls minimum number of samples per pixel. A value of zero means one sample per pixel; -1 means one sample every two pixels; -2 means one sample every 4 pixels, etc.

Max rate - Controls maximum number of samples per pixel; zero means one sample per pixel, 1 means four samples, 2 means eight samples etc.

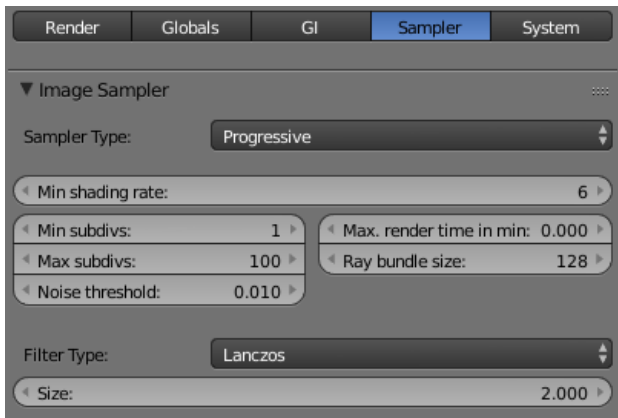
Threshold - Determines the sensitivity of the sampler to changes in pixel intensity. Lower values will produce better results, while higher values will be faster, but may leave some areas of similar intensity undersampled.

Normals threshold - Determines the sensitivity of the sampler to the variation of normals.

Jitter - Displaces the samples slightly to produce better antialiasing of nearly horizontal or vertical lines.

Edges - Causes the image sampler to always supersample object edges (regardless of whether they actually need to be supersampled). This option has no effect if DOF or motion blur is enabled.

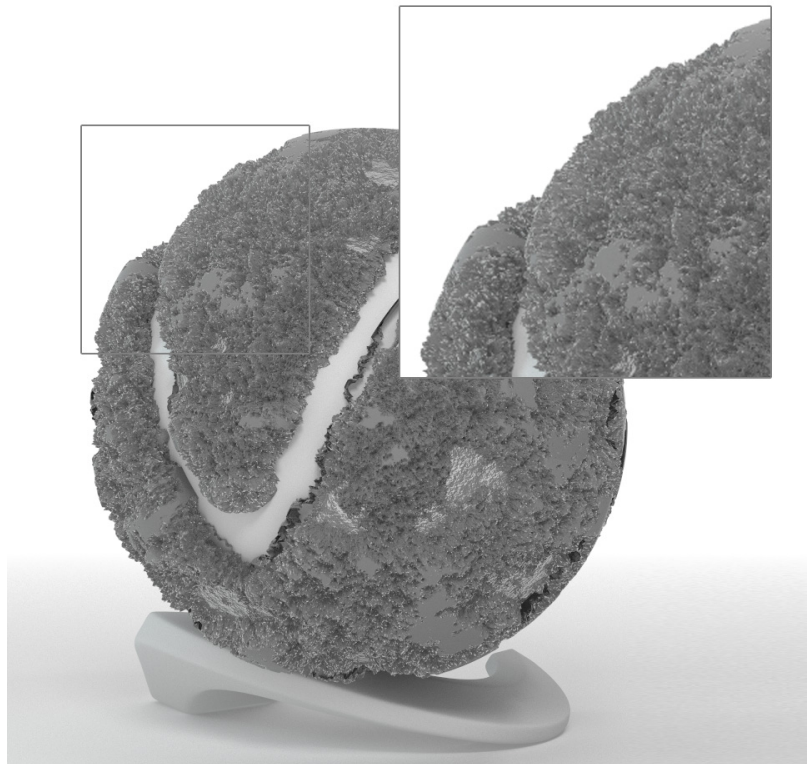
Normals - Supersamples areas with sharply varying normals. This option has no effect if DOF or motion blur is enabled.



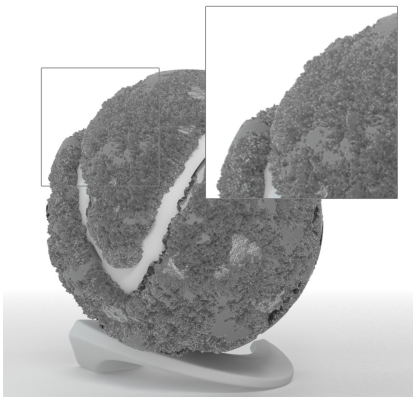
Example: Anti-aliasing Filters

Here is an example briefly demonstrating the effect of different anti-aliasing filters on the final result.

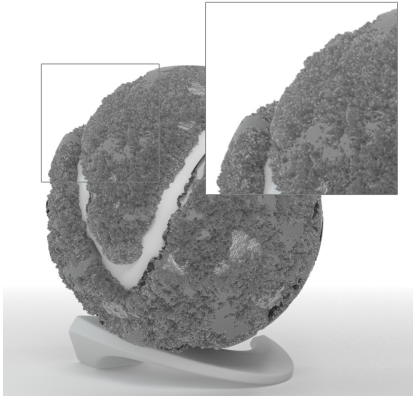
Note that rendering with a particular filter is **not** the same as rendering without a filter and then blurring the image in a post-processing program like Adobe Photoshop. Filters are applied on a sub-pixel level, over the individual sub-pixel samples. Therefore, applying the filter at render time produces a much more accurate and subtle result than applying it as a post effect. The zoomed in images below have been zoomed in and cropped 300%.



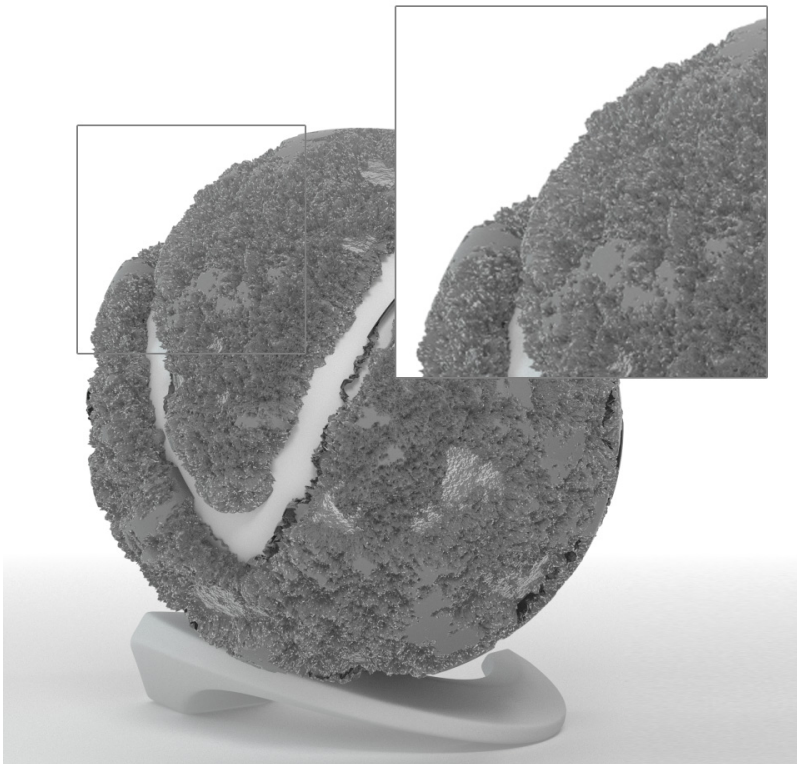
Filtering is off.



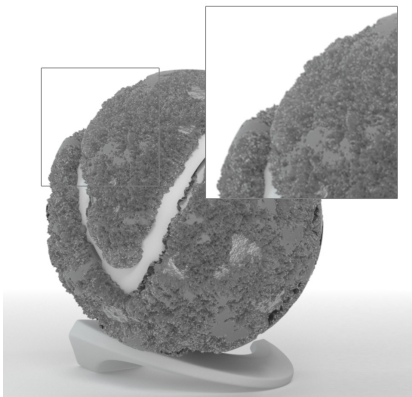
Lanczos filter, size 2.5



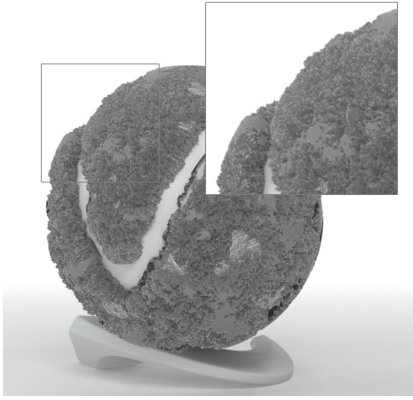
Triangle filter



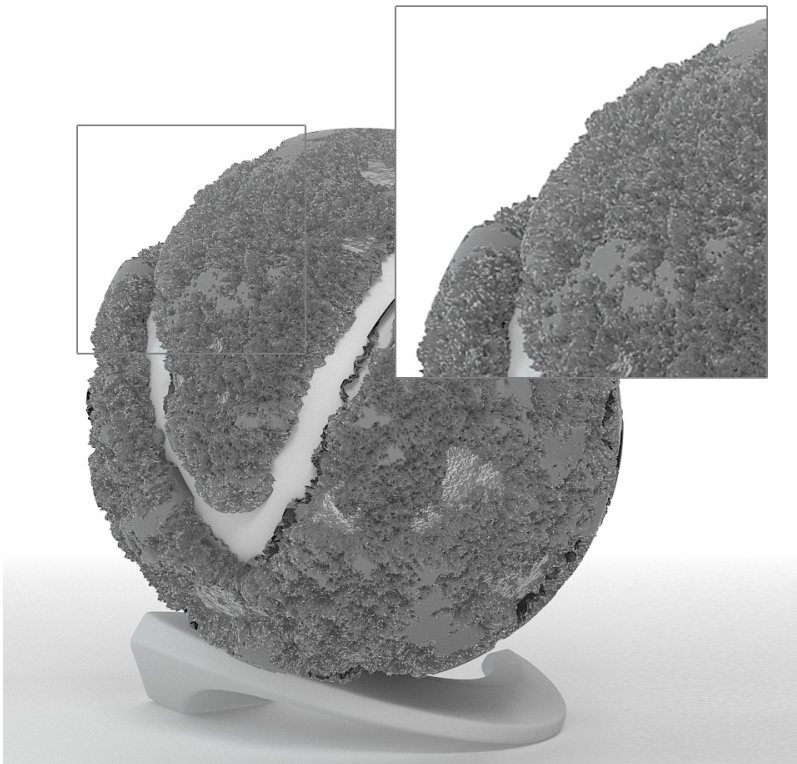
Sinc Filter, size 2.5



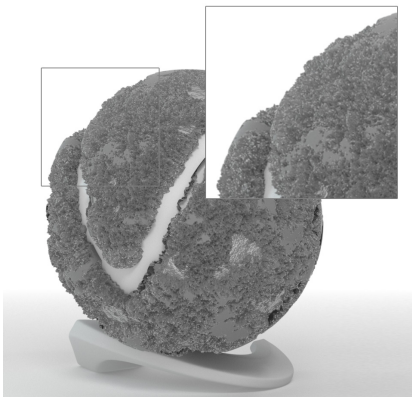
Box Filter, size 2.5



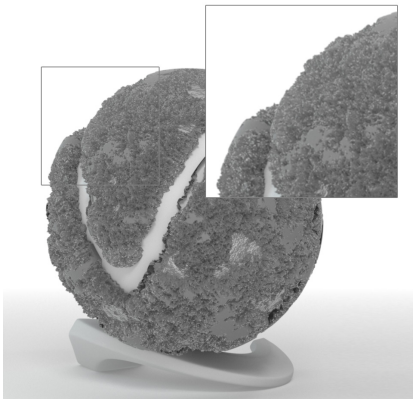
Area filter, size 2.5



Catmull-Rom



Cook Variable, size 2.5



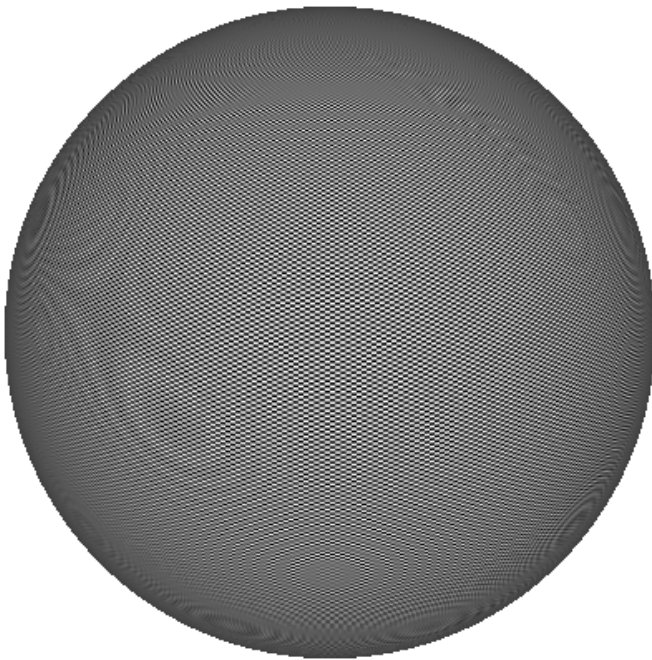
Gaussian, size 2.5

Example: Anti-aliasing Filters and Moire Effects

This example demonstrates the effect anti-aliasing filters have on moire effects in your images. Sharpening filters may enhance moire effects, even if your image sampling rate is very high. Blurring filters reduce moire effects.

Note that moire effects are not necessarily a result of poor image sampling. In general, moire effects appear simply because the image is discretized into square pixels. As such, they are inherent to digital images. The effect can be reduced through the usage of different anti-aliasing filters, but is not completely avoidable.

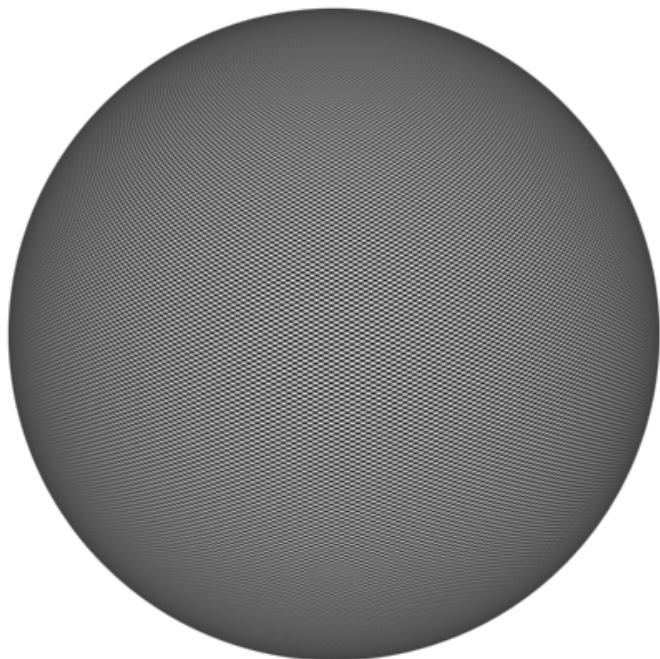
The scene is very simple: a sphere with a very fine checker map applied. The images were rendered with a very high sampling rate (15 subdivs, or 225 rays/pixel). This is enough to produce quite an accurate approximation to the pixel values. Note that the image looks quite different depending on the filter:



No Filter



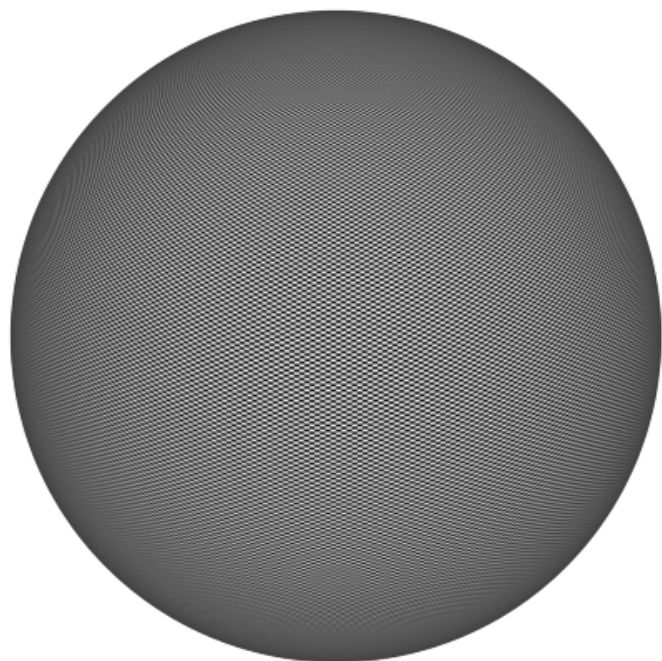
Box Filter



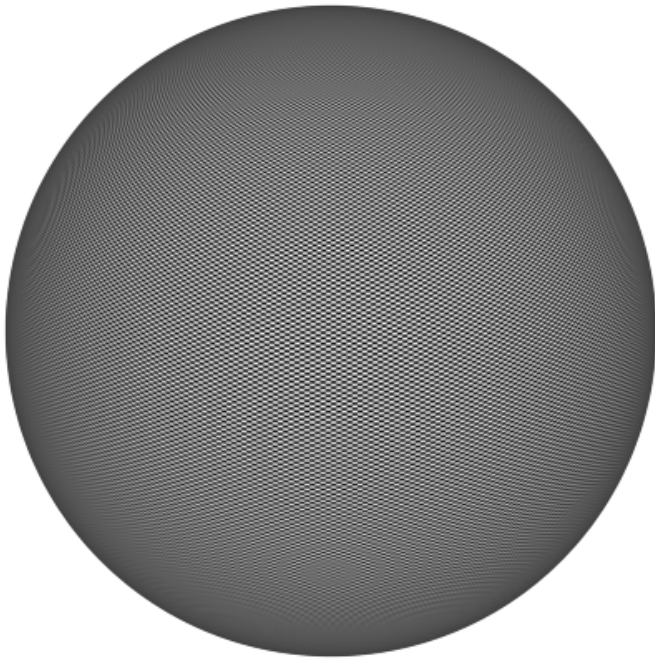
Area Filter, size 1.5



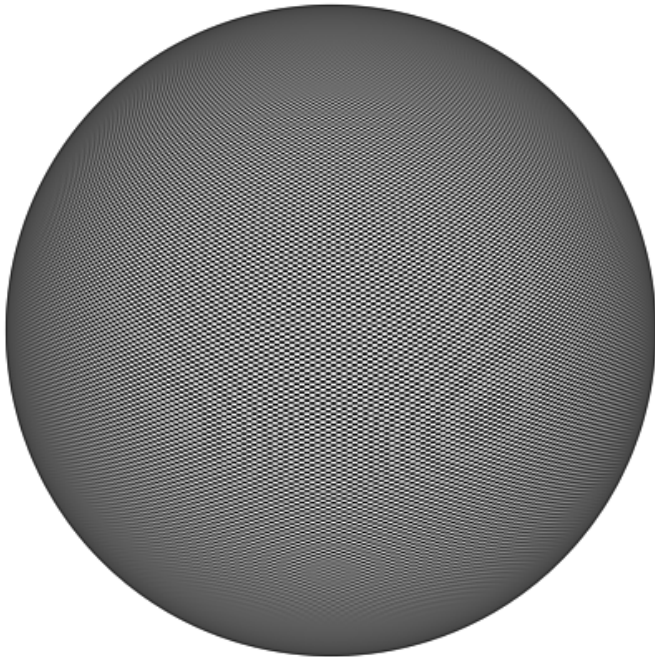
Area Filter, size 4.0



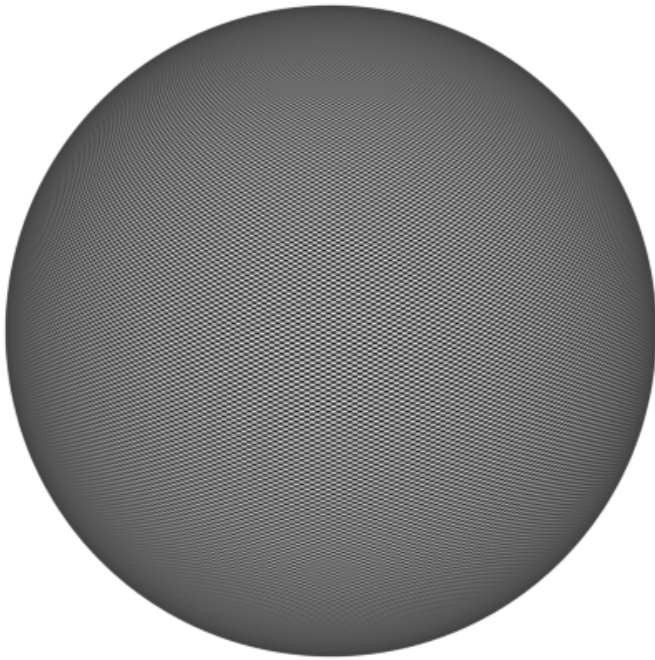
Triangle Filter, size 1.5



Lanczos Filter



Catmull Rom



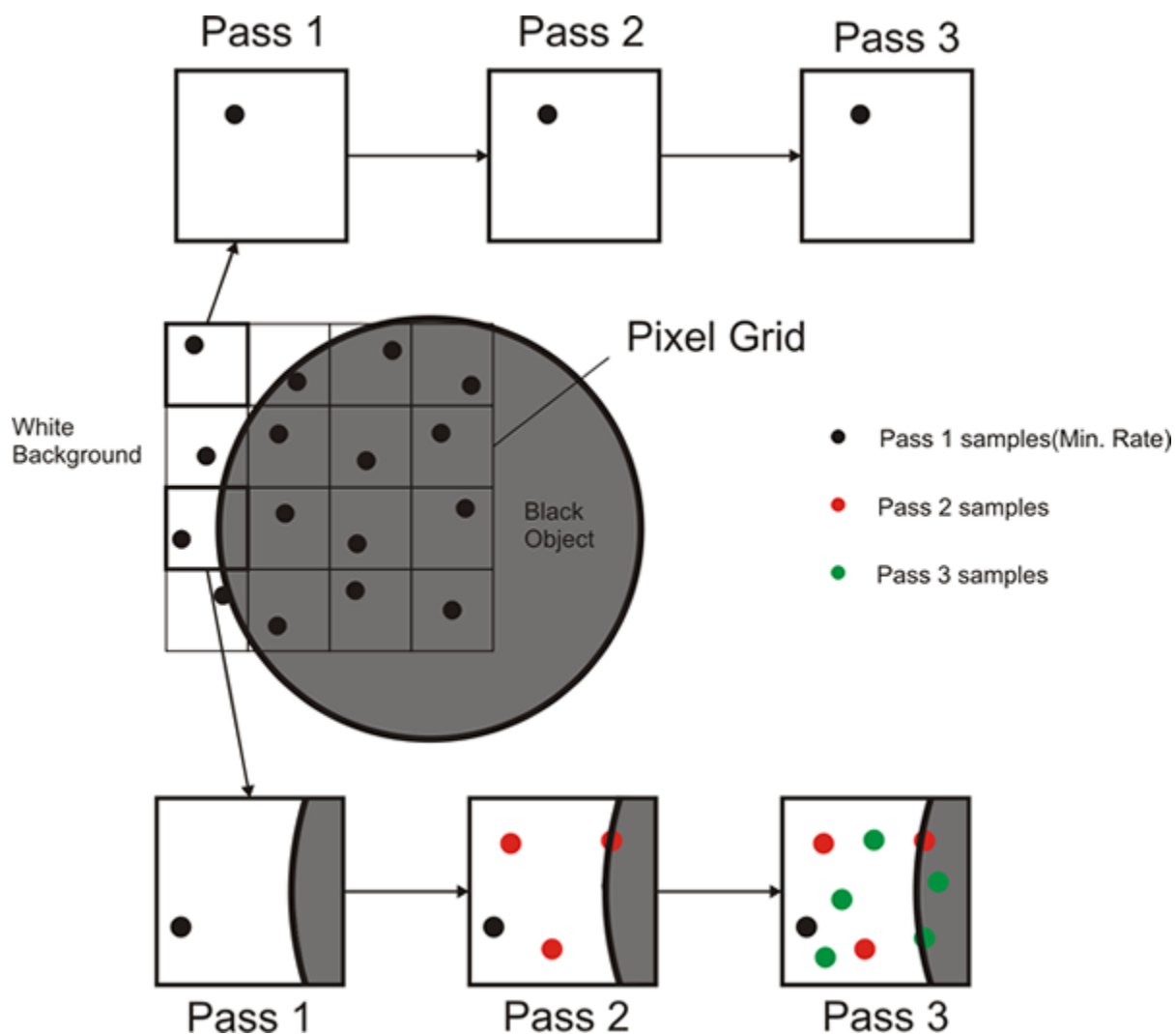
Gaussian Filter, size 1.5



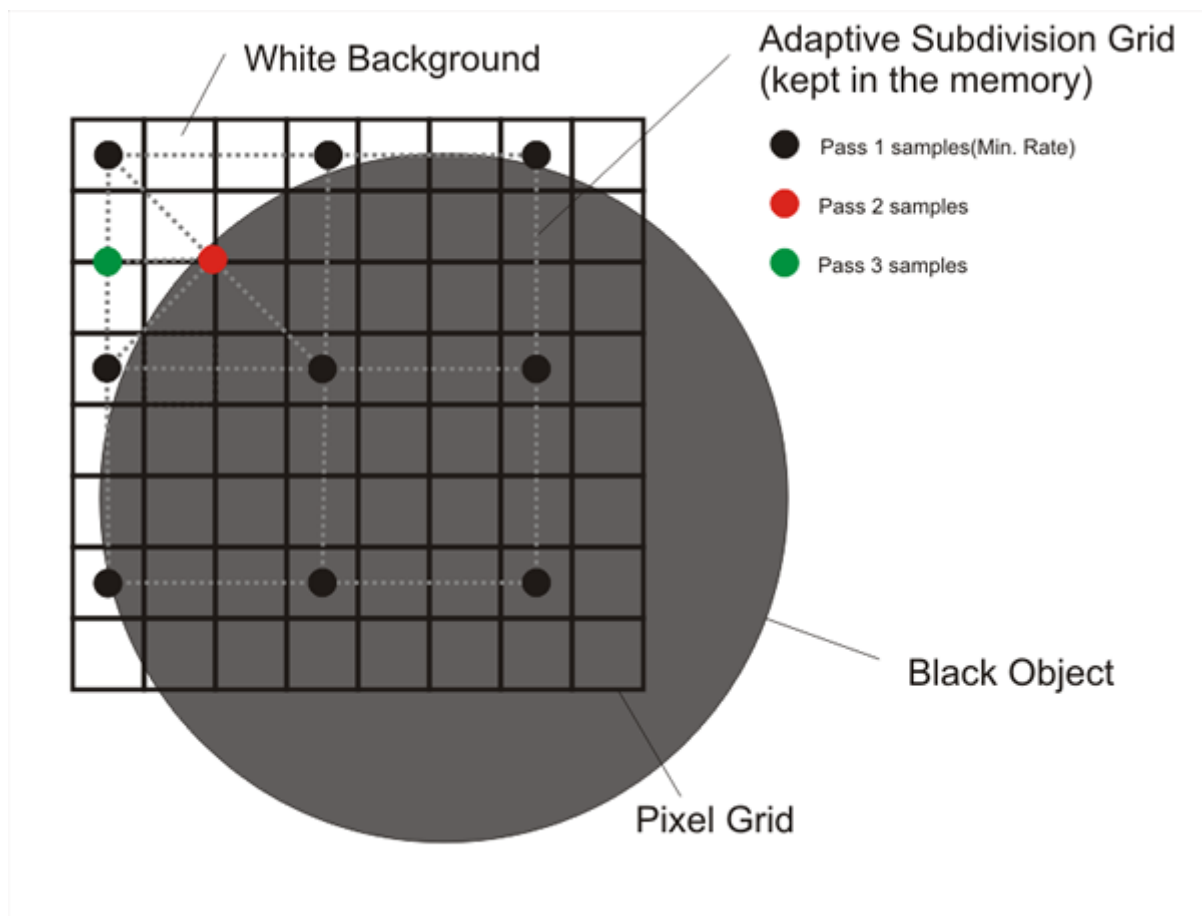
Gaussian filter, size 6.0

Drag the slider to change the filter type.

The following diagram shows visually the way V-Ray is placing samples when using the *Adaptive DMC* sampler. The black squares represent the pixels of the image while the dots represent the individual samples. In the first pass, V-Ray always places the minimum number of samples determined by the **Min subdivs** parameter. Then, the color of samples is compared and more are added where needed in the following passes.



The following diagram shows visually the way V-Ray works when using the *Adaptive Subdivision* image sampler. With this mode V-Ray creates a secondary grid on top of the pixel grid and uses this grid to position the samples. This allows it to use less than a sample per pixel. After the first pass, the samples are compared and, if the difference between two samples is bigger than the values in the thresholds, the grid is subdivided and more samples are added. During the whole time, V-Ray has to keep the whole grid in the memory, which makes this method less memory efficient compared to the other two methods - see the [Notes](#) below.



Example: Stages of Rendering with the Progressive Sampler

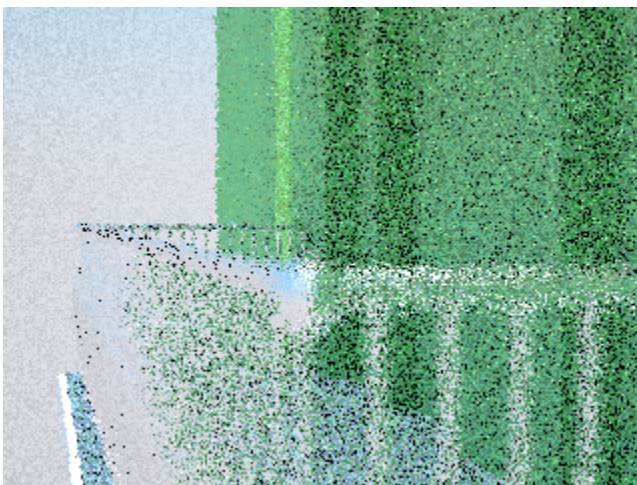


Image after 1 pass

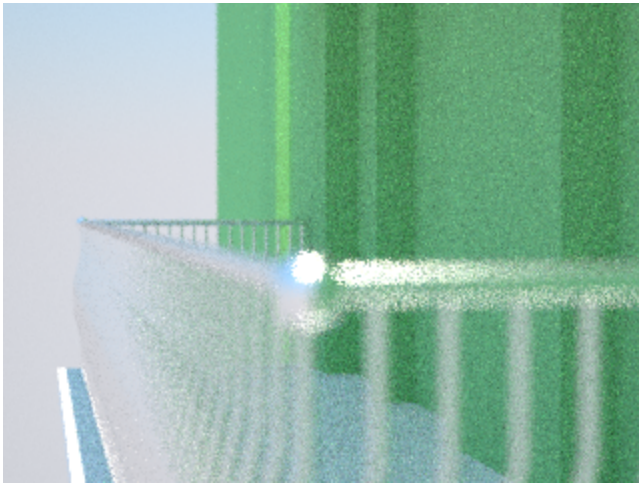


Image after 16 passes

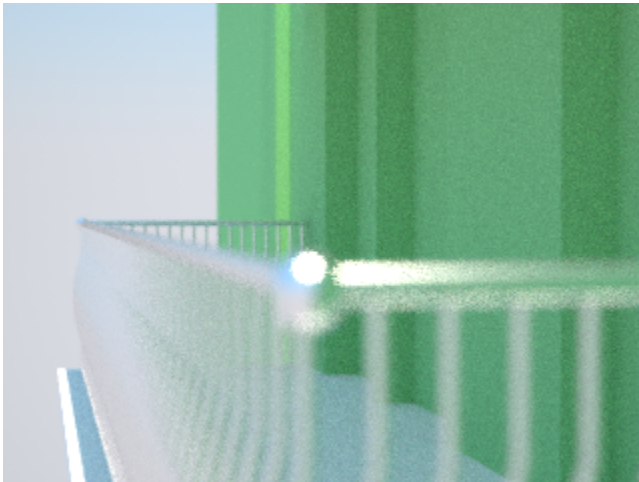


Image after 64 passes

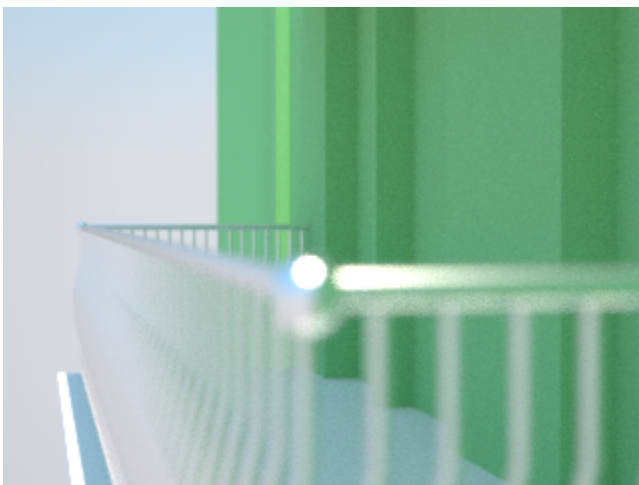
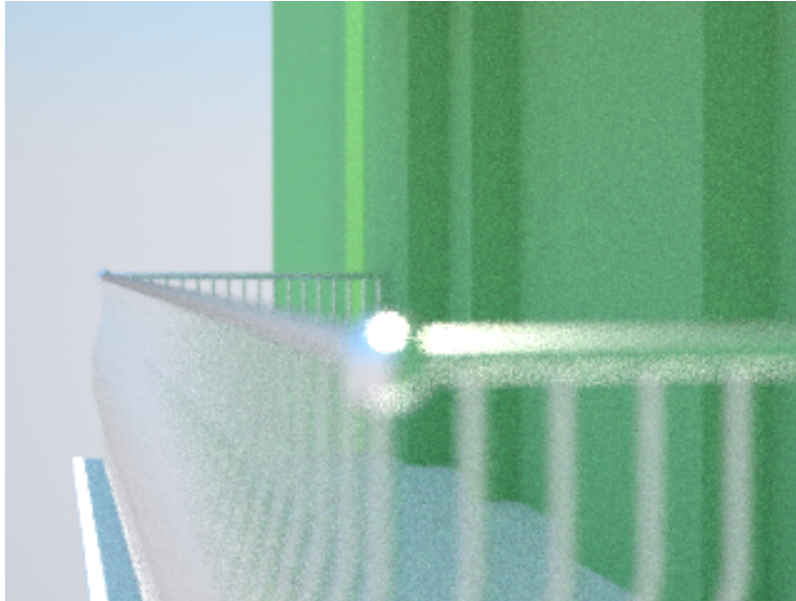


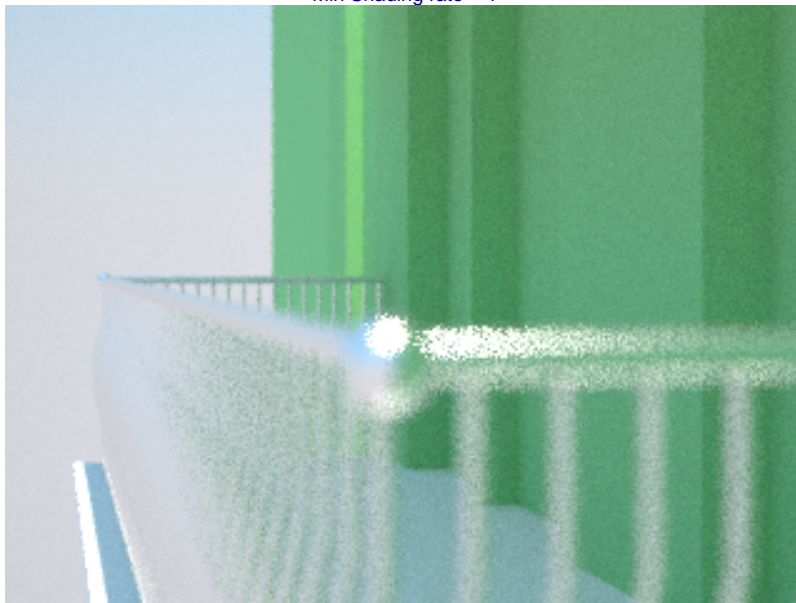
Image after 256 passes

1 pass
256 passes

Example: Min Shading Rate



Min Shading rate = 1



Min Shading rate = 16

Both of the images above rendered with the *Progressive* image sampler for the same amount of time. With Min Shading rate set to 16, at least 16 GI rays are shot for each AA ray, which produces smoother GI, but overall leads to less samples left for the DOF effect.

Notes

- Which sampler to use for a given scene? The answer is best found with experiments, but here are some tips:

- For smooth scenes with only a few blurry effects and smooth textures, the *Adaptive Subdivision* sampler with its ability to undersample the image is unbeatable.
- For images with detailed textures or lots of geometry detail and only a few blurry effects, the *Adaptive DMC* sampler performs best. Also in the case of animations involving detailed textures, the *Adaptive Subdivision* sampler might produce jittering which the *Adaptive DMC* sampler avoids.
- For complex scenes with lots of blurry effects and/or detailed textures, the *Fixed* sampler performs best and is very predictable with regards to the quality and render time.
- A note on RAM usage: image samplers require substantial amount of RAM to store information about each bucket. Using large bucket sizes may take a lot of RAM. This is especially true for the *Adaptive Subdivision* sampler, which stores all individual sub-samples taken within a bucket. The *Adaptive DMC* sampler and the *Fixed* sampler on the other hand only store the summed result of all sub-samples for a pixel and so usually require less RAM.