V-Ray Denoiser

This page gives information on the V-Ray Denoiser Render Element.

Overview

The V-Ray Denoiser takes an existing render and applies a denoising operation to it after the image is completely rendered out via normal means. The denoising operation detects areas where noise is present and smooths them out.

Images can also be denoised using the Standalone Denoise Tool included with the installation of V-Ray for Houdini.

For animations, it is recommended to use the standalone denoiser tool. It performs frame blending and reduces flickering.

The V-Ray Denoiser operates on other render elements (like RGB_Color) rather than being part of the rendering process itself, therefore the denoising operation does not require re-rendering of the scene.

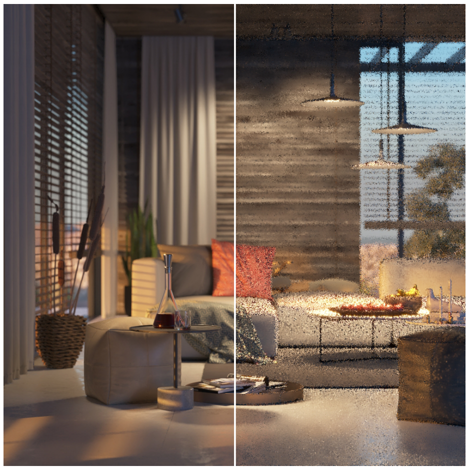
There are three denoising engines to choose from - the **Default V-Ray denoiser**, the **NVIDIA AI denoiser** (V-Ray's implementation of NVIDIA's AI-based denoising algorithm), and the **Intel Open Image Denoise**.

When rendering, the V-Ray Denoiser automatically adds a few render channels in the V-Ray Frame Buffer which are required to guide the denoising algorithm. The two denoising engines require different render elements. Some of them are standard render channels like the diffuse filter color, the reflection filter color, etc. A few special channels are also generated for the Default V-Ray Denoiser:

- The **effectsResult** channel holds the result of the denoising operations and the lens effects that are executed over that image. The RGB Channel button in the VFB will toggle between the effectsResult and original RGB color channels.
- The **noiseLevel** channel is the amount of noise for a pixel as estimated by the V-Ray image sampler.
- The **defocusAmount** channel is non-black when depth of field and motion blur are enabled and contains the estimated pixel blurring in screen space.
- The **Denoiserchannel** contains the result of the noise removal. This channel appears in the VFB only if **mode** is set to **Show denoiser result channel**.

Currently the scene can contain only one Denoiser Render Element. Future versions of V-Ray will support multiple Denoiser Render Elements with different settings.

VRayDenoiser can be applied to the Viewport IPR by enabling the **Use Denoiser** option from V-Ray Renderer > Export > IPR.



Denoising Engines

VRayDenoiser offers a choice between the **Default V-Ray denoiser**, the **NVIDIA AI denoiser**, and the **Intel Open Image Denoise**. Each offers a different denoising algorithm that comes with different benefits. Each offers a different denoising algorithm that comes with different benefits. *See the denoising engine examples below.*

Default V-Ray Denoiser - V-Ray's denoising algorithm. It can utilize the CPU or the GPU (AMD or NVIDIA GPUs) to perform the denoising. It is consistent when denoising render elements, as it applies the same denoising operator to all render channels, which means that it is recommended for denoising the render elements to be used for compositing back the beauty image. In addition, it comes with a standalone version, which is recommended for denoising animation by using frame blending.

NVIDIA AI Denoiser - V-Ray's integration of NVIDIA's AI-based denoising algorithm. The NVIDIA AI Denoiser **requires an NVIDIA GPU** to work, regardless of whether the actual rendering was performed on the CPU or GPU. This means that rendering on the CPU still requires an NVIDIA GPU for denoising with the NVIDIA AI Denoiser and has some advantages and drawbacks compared to the Default V-Ray Denoiser. For example, the NVIDIA AI Denoiser performs the denoising faster, but is not consistent when denoising render elements. This means that there will be differences between the original RGB image and the one reconstructed from render elements that are denoised with the NVIDIA AI Denoiser. It also doesn't support cross-frame denoising and will likely produce flickering when used in animation.

The Nvidia AI Denoiser only works on Nvidia Maxwell and newer GPU architectures.

Intel Open Image Denoise – V-Ray's integration of Intel Open Image Denoise. The Intel Open Image denoiser works with your CPU device and does not use hardware acceleration.

Parameters

Enabled – Enables V-Ray Denoiser.

Name – The text added to the end of the rendered file, when saved as a separate file (e.g. myrender.Denoiser.vrimg).

Engine – Allows choosing between the **Default V-Ray denoiser**, the **NVIDIA AI denoiser**, or **Intel Open Image Denoise**. Note that, the NVIDIA AI Denoiser requires an NVIDIA GPU. *See the denoising engine examples below.*

Hardware Acceleration – Uses the GPU device(s) to accelerate the denoising calculations. In case there is no compatible GPU device, denoising automatically falls back to use the CPU, even if the option is enabled. When the NVIDIA AI denoiser is used, this option is not available, as it requires an NVIDIA GPU.

Preset – When using the **Default V-Ray Denoiser**, use the presets to automatically set the **Strength** and **Radius** values.

> **Default** – Applies a mid-level denoising.
> **Mild** – Applies a more subtle level of denoising than the Default preset.
> **Strong** – Applies a stronger level of denoising than the Default preset.
> **Custom** – Allows the **Strength** and **Radius** parameters to be set to custom values.

Mode – Specifies how the results of the Denoiser are saved.

> **Only Generate Render Elements** – All render elements required for denoising are generated so that denoising can be done with the Standalone Denoise Tool. The information calculated within them is not applied to other render elements, and no V-Ray Denoiser Render Element is generated.
> **Hide Denoiser Element** – The V-Ray Denoiser channel is not present separately in the VFB. The effectsResult channel is generated with the denoised image.
> **Show Denoiser Element** – The V-Ray Denoiser Render Element is generated to contain a denoised version of the RGB Color Render Element using the specified settings. The original render elements, including the RGB Color Render Element, are not changed.

Radius – Specifies the area around each pixel to be sampled for determining how to denoise a given pixel. Larger values produce smoother results, but slow down the denoiser.

Strength – Determines how strong the denoising operation is. Larger values remove noise more aggressively, but may blur the image too much.

NVIDIA AI Upscale – When enabled, the rendered image is internally calculated with half the resolution set in the render setup (in each dimension), and the denoised image is upscaled to the full resolution using NVIDIA's AI upscaling technology. NVIDIA's AI upscaling technology represents an advanced method based on machine learning which provides better and more detailed results when compared to a standard image interpolation. For example, a 3840x2160 resolution renders the image internally in 1920x1080 and the AI upscaling denoiser produces the final 3840x2160 image from the lower resolution render.

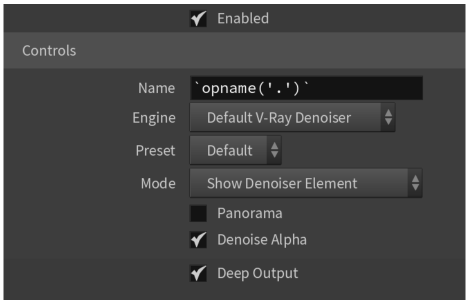
Render elements that don't have the **Denoise** option enabled are upscaled using simple interpolation.
It is recommended to use NVidia AI upscale option with the latest recommended NVIDIA driver.

Temporal Mode – Only available with the **NVIDIA AI denoiser**. When enabled, the **Denoiser** uses information from previous frames to create a smoother transition. Useful for rendering animation.

Panorama – Specifies that the denoised image can be wrapped around the left/right border. The denoiser can then use this information to avoid generating artifacts over the seamed area.

Denoise Alpha – Enabled by default. When disabled, the lpha channel remains undenoised.

Deep Output – Specifies whether to include this render element in deep images.



☑ Enabled

Controls

Name `opname('.')`

Engine Default V-Ray Denoiser

Preset Default

Mode Show Denoiser Element

□ Panorama

☑ Denoise Alpha

☑ Deep Output

Suggested Render Settings

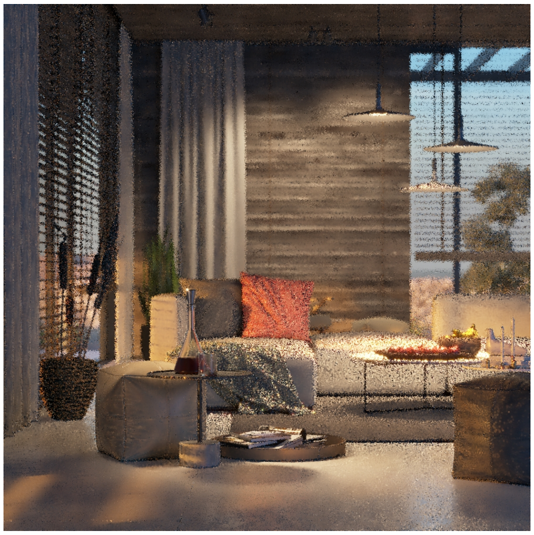
While the denoiser can be quite effective at removing noise, it may produce artifacts and loss of detail if the image is very noisy. For most scenes, use **Bucket** or **Progressive** image sampler with the **Noise threshold** set to 0.05 or lower. Additionally, the denoiser works best when the noise levels across the image are similar (the **noiseLevel** render channel is uniform grey), so using very low sampling is not recommended.

When rendering animations, disabling the **Animated Noise Pattern** option in the DMC Sampler rollout of the Sampler tab generally improves the results.

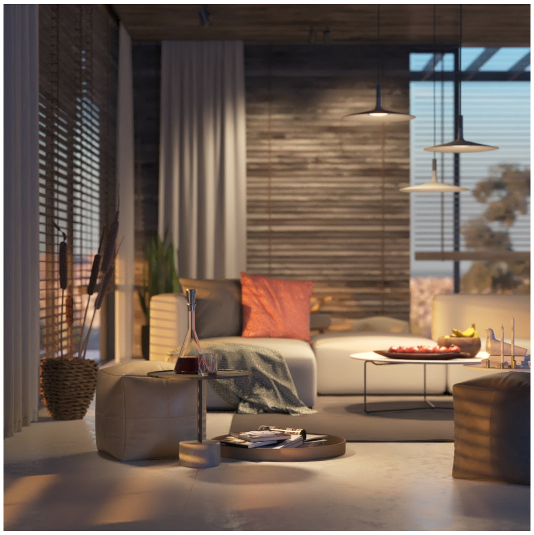
Using the Standalone vdenoise tool on the rendered frames can additionally improve the quality of the animation.

Example: Default V-Ray Denoiser

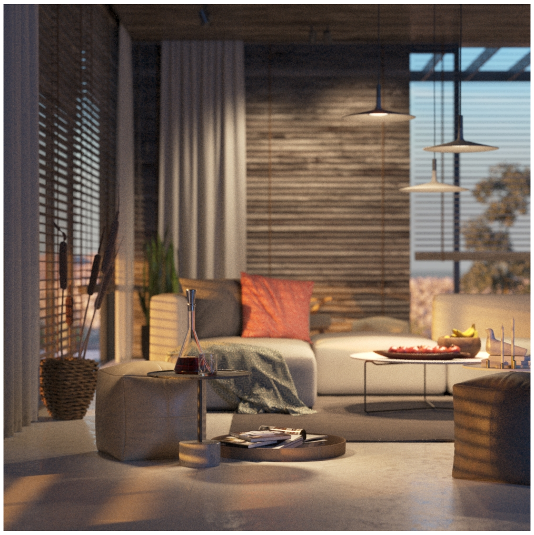
The example below illustrates how the Default V-Ray Denoiser works after more samples are made with the Progressive image sampler. When the samples are too few, there is not enough information for the denoising to produce a smooth result. You can compare the results between an image with applied denoising and without.



Sampling pass 1



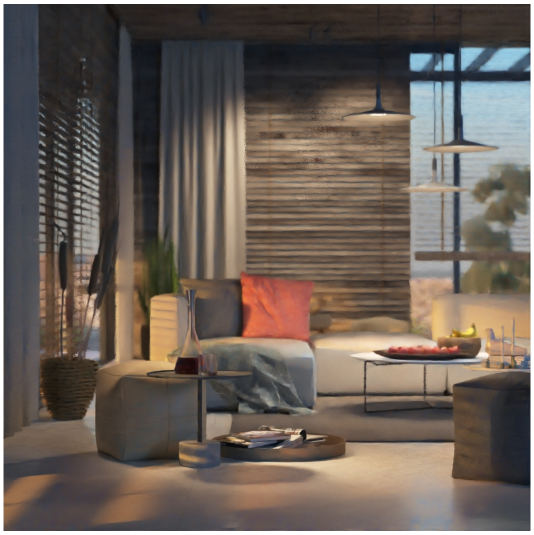
Sampling pass 64



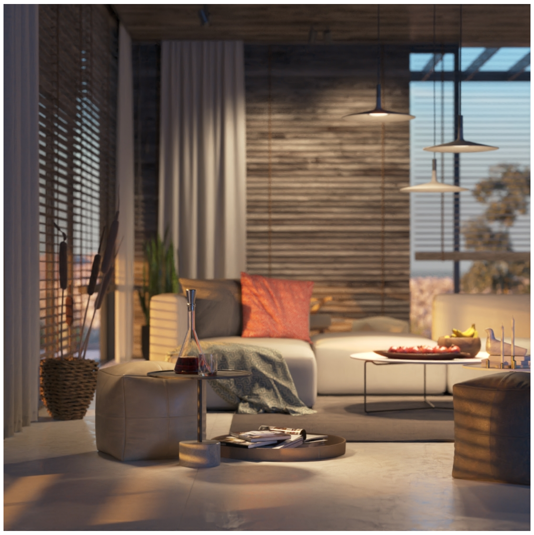
No Denoising

Example: NVIDIA AI Denoiser

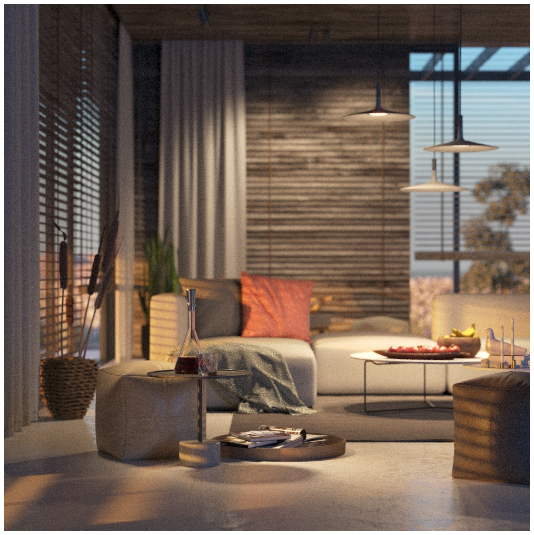
The example below illustrates how the NVIDIA AI Denoiser works after more samples are made with the Progressive image sampler. When the samples are too few, there's not enough information for the denoising to produce a smooth result. You can compare the results between a render with applied denoising and without.



Sampling pass 1



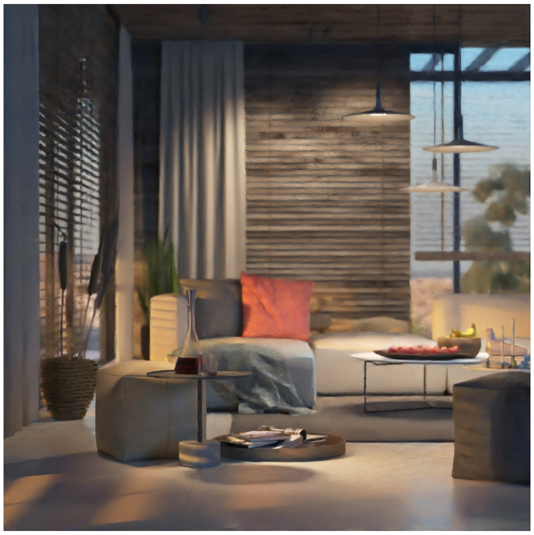
Sampling pass 64



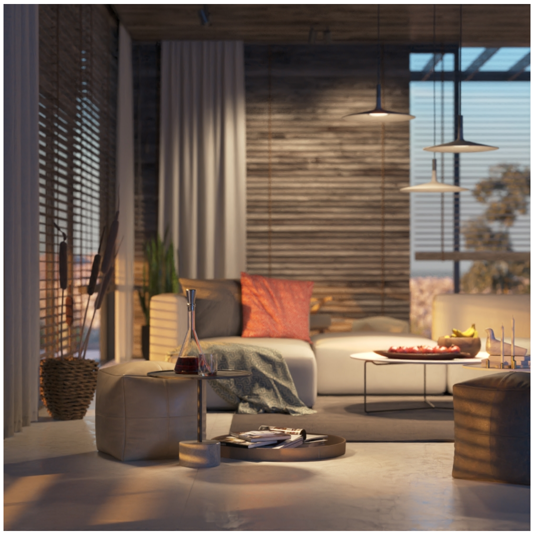
No Denoising

Example: Intel Open Image Denoise

The example below illustrates how the Intel Open Image Denoise works after more samples are made with the Progressive image sampler. When the samples are too few, there's not enough information for the denoising to produce a smooth result. You can compare the results between a render with applied denoising and without.



Sampling pass 1



Sampling pass 64

No Denoising

Denoising Animations

When denoising animations, it is recommended to use the Standalone Denoiser Tool. Unlike the denoiser integrated in the UI, the standalone tool can do frame blending for animations, which reduces flickering. The integrated denoiser only works on the rendered frame and does not take the next and previous frame(s) into account, like the standalone tool does.

To denoise an image sequence with **vdenoise** run the following command:

```
vdenoise -inputFile="path\to\sequence_????.ext
```

where the question mark (?) replaces the digits in the sequence's file names.

For example, if the images in the sequence are named *anim_0001.exr*, *anim_0002.exr*, etc. and are located in the folder *c:\renderoutput*, the full command will be:

```
vdenoise -inputFile="c:\renderoutput\anim_????.exr
```

When that command is run, the sequence is read and for each frame, the specified number of adjacent frames are also considered. A new output image is then written for each frame.

Recommended settings:

- **ode** set to **Only Generate Render Elements**.
- **denoising engine** set to **Default V-Ray Denoiser**.
- Render output set to **vrimg** or **multichannel exr**.

The NVIDIA AI Denoiser does not perform frame blending and will likely produce flickering when denoising animations.

Notes

- When bucket rendering, image denoising takes place after the frame has been rendered and does not show up until all rendering has finished.
- When progressive rendering, image denoising takes place during the rendering. How frequently the denoising is updated is controlled with the **Post effects rate** parameter found in Render Setup window > **Settings** tab > **System** rollout.

- Textures or materials such as **V-Ray Stochastic Flakes** material that could be considered to have a purposely noisy look, are not considered "noisy" by V-Ray Denoiser, and are not affected by the noise removal process.