

Global DMC

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

Overview

Monte Carlo (MC) sampling is a method for evaluating "blurry" values (antialiasing, depth of field, indirect illumination, area lights, glossy reflections/refractions, translucency, motion blur, etc). V-Ray uses a variant of Monte Carlo sampling called Deterministic Monte Carlo (DMC).

The difference between pure Monte Carlo sampling and Deterministic Monte Carlo is that the first uses pseudo-random numbers which are different for each and every evaluation (and so re-rendering a single image will always produce slightly different results in the noise), while Deterministic Monte Carlo uses a pre-defined set of samples (possibly optimized to reduce the noise), which allows re-rendering an image to always produce the exact same result. By default, the Deterministic Monte Carlo method used by V-Ray is a modification of Schlick sampling, introduced by Christophe Schlick in [1] (see the [References](#) section below for more information).

Instead of having separate sampling methods for each of the blurry values, V-Ray has a single unified framework that determines how many and which exact samples are to be taken for a particular value, depending on the context in which that value is required. This framework is called the *DMC sampler*.

UI Path

||Properties editor|| > **Render** > **Sampler** tab > **Global DMC** rollout

Parameters

Adaptive Amount – Controls the extent to which the number of samples depends on the importance of a blurry value. It also controls the minimum number of samples that are taken. A value of 1.0 means full adaptation; a value of 0.0 means no adaptation.

Min samples – Determines the minimum number of samples that must be made before the early termination algorithm is used. Higher values slow things down but make the early termination algorithm more reliable.

Noise Thresh – Controls V-Ray's judgement of when a blurry value is "good enough" to be used. This directly translates to noise in the result. Smaller values mean less noise, more samples, and higher quality. A value of 0.0 means that no adaptation is performed.

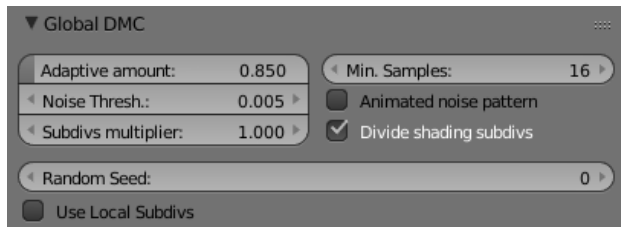
Subdivs multiplier – Multiplies all subdiv values during rendering. You can use this to quickly increase/decrease sampling quality everywhere. This affects everything, except for the light cache, caustics and AA subdivs. Everything else (dof, motion blur, irradiance map, brute-force GI, area lights, area shadows, glossy reflections/refractions) is affected by this parameter.

Animated noise pattern – When this option is Off, the sampling pattern is the same from frame to frame in an animation. Since this may be undesirable in some cases, you can turn this option Off to make the sampling pattern change with time. Note that re-rendering the same frame produces the same result in both cases.

Divide Shading Subdivs – When enabled V-Ray automatically adjusts the number of samples in lights, materials etc., when you change the antialiasing settings in order to achieve the same quality.

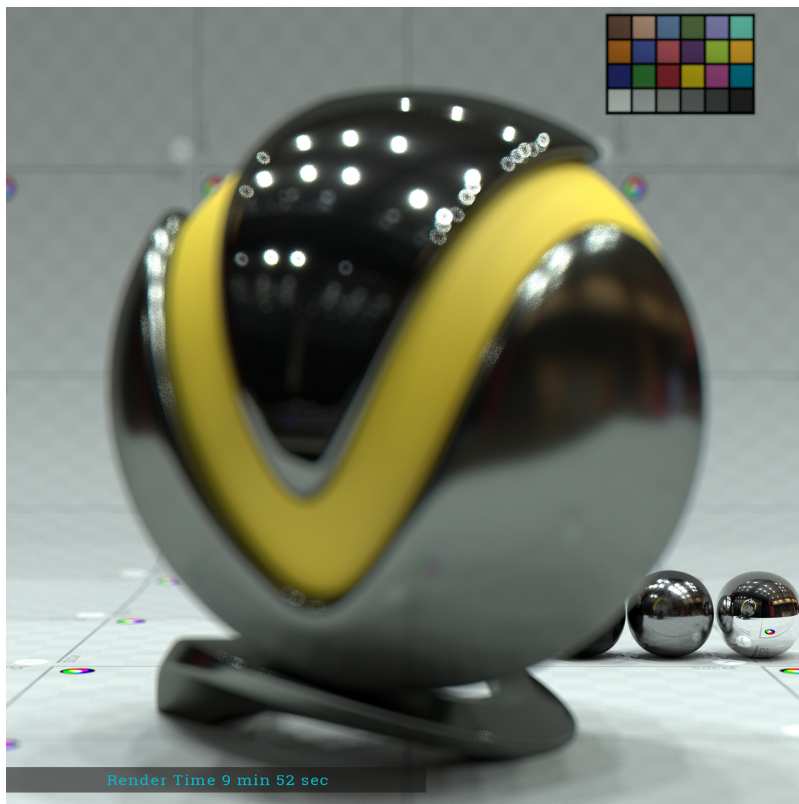
Random Seed – Changes the random seed for the noise pattern.

Use Local Subdivs – When disabled, V-Ray automatically determines subdiv values for sampling of materials, lights and other shading effects based on the **Min shading rate** parameter of the [Image Sampler](#). When enabled, V-Ray uses the Subdivs Mult parameter and the Subdiv values of individual V-Ray lights and materials.

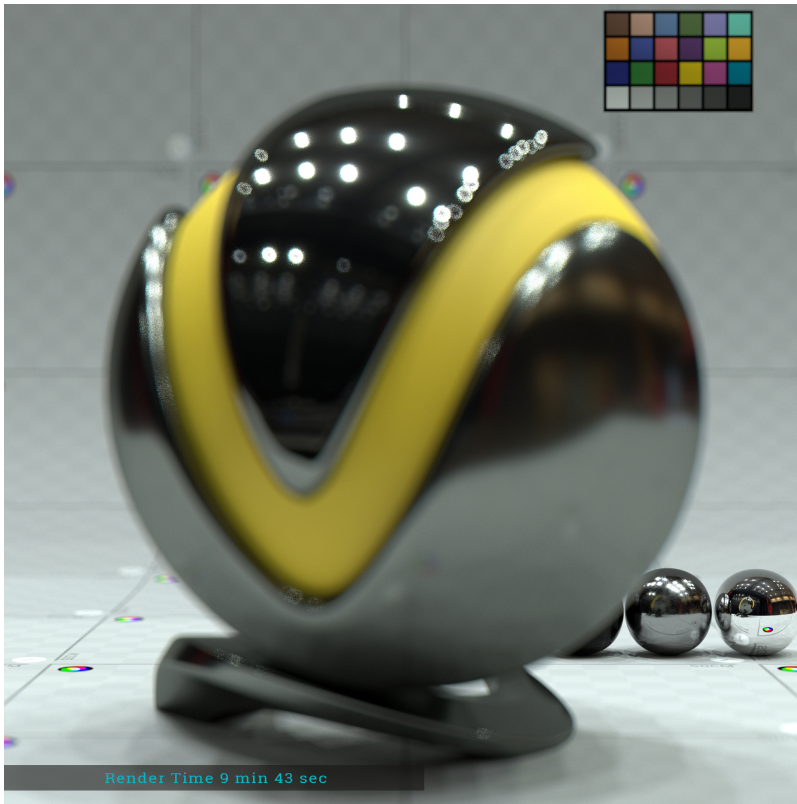


Example: Noise vs Speed

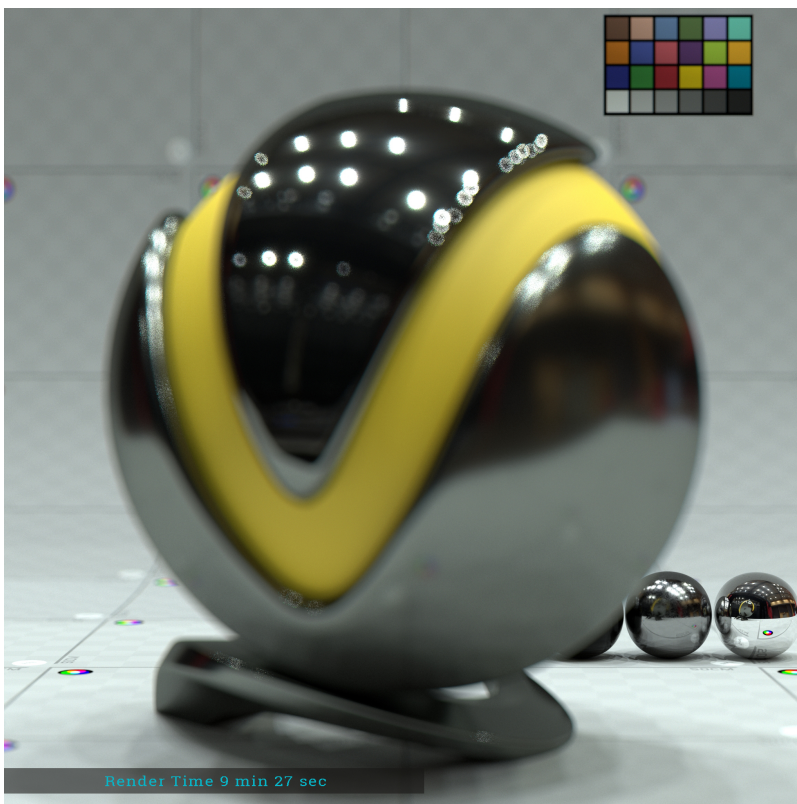
The results in the render quality and render time are negligibly small. That is why, we highly recommend the default settings that work for a wide variety of scenes.



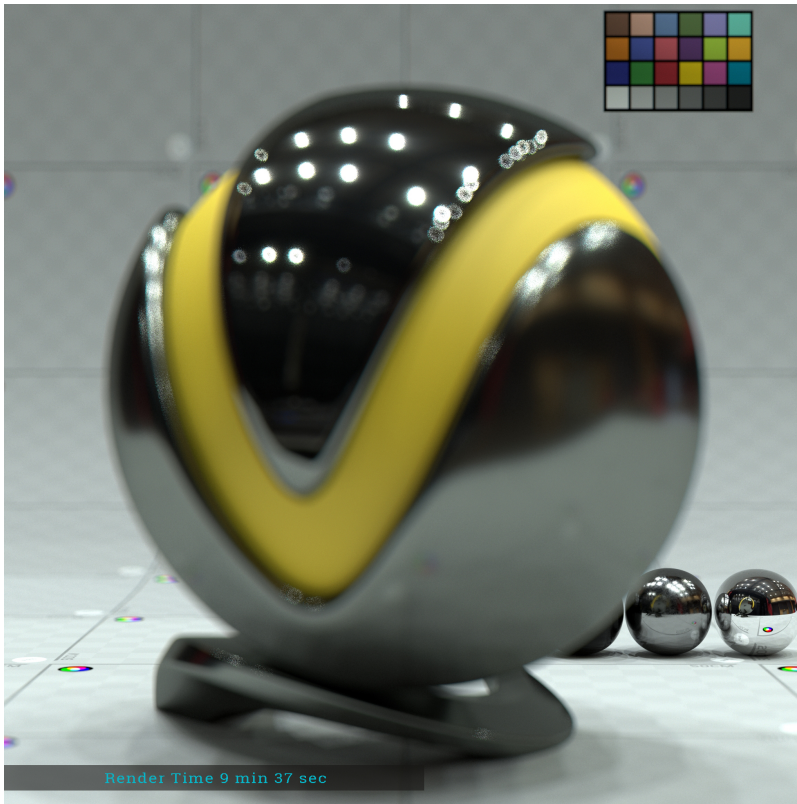
Adaptive amount 0,85; Noise threshold 0,005; Min samples 16



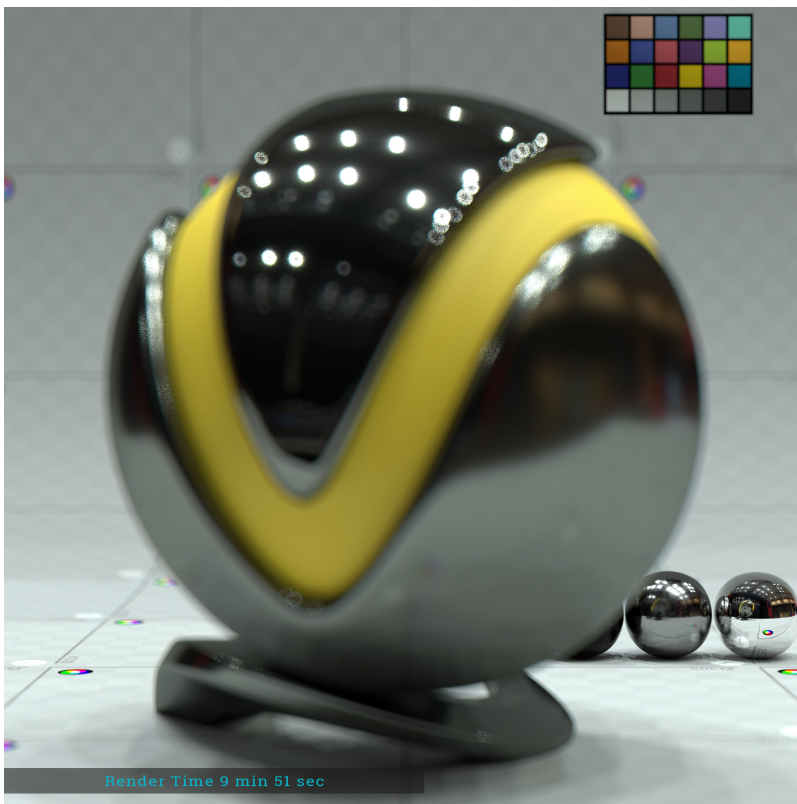
Adaptive amount 0,95 Noise threshold 0,01 Min samples 5



Adaptive amount 0,95 Noise threshold 0,05 Min samples 5



Adaptive amount 0,99 Noise threshold 0,05 Min samples 5



Adaptive amount 1 Noise threshold 0,1 Min samples 5

Determining Values for DMC Sampler

The actual number of samples for any blurry value is determined based on three factors:

- The subdivs value supplied by the user for a particular blurry effect. This is multiplied by the **Subdivs multiplier** parameter.
- The importance of the value (for example, dark glossy reflections can do with fewer samples than bright ones, since the effect of the reflection on the final result is smaller; distant area lights require fewer samples than closer ones. Basing the number of samples allocated for a value on importance is called *importance sampling*. The variance (think "noise") of the samples taken for a particular value - if the samples are not very different from each other, then the value can do with fewer samples; if the samples are very different, then a larger number of them are necessary to get a good result. This works by looking at the samples as they are computed one by one and deciding, after each new sample, if more samples are required. This technique is called *early termination* or *adaptive sampling*.

References

More information on deterministic Monte Carlo sampling for computer graphics can be found from the sources listed below.

- [1] C. Schlick, **An Adaptive Sampling Technique for Multidimensional Integration by Ray Tracing**, in Second Eurographics Workshop on Rendering (Spain), 1991, pp. 48-56
Describes deterministic MC sampling for antialiasing, motion blur, depth of field, area light sampling and glossy reflections.
- [2] K. Chiu, P. Shirley and C. Wang, **Multi-Jittered Sampling**, in Graphics Gems IV, 1994
Describes a combination of jittered and N-rooks sampling for the purposes of computer graphics.
- [3] Masaki Aono and Ryutarou Ohbuchi, **Quasi-Monte Carlo Rendering with Adaptive Sampling**, IBM Tokyo Research Laboratory Technical Report RT0167, November 25, 1996, pp.1-5
An online version can be found at http://www.kki.yamanashi.ac.jp/~ohbuchi/online_pubs/eg96_html/eg96.htm
Describes an application of low discrepancy sequences to area light sampling and the global illumination problem.
- [4] M. Fajardo, **Monte Carlo Raytracing in Action**, in **State of the Art in Monte Carlo Ray Tracing for Realistic Image Synthesis**, SIGGRAPH 2001 Course 21, pp. 151-162;
An online version can be found at <http://cseweb.ucsd.edu/~viscomp/classes/cse274/wi18/readings/course29sig01.pdf>
Describes the ARNOLD renderer employing randomized quasi-Monte Carlo sampling using low discrepancy sequences for pixel sampling, global illumination, area light sampling, motion blur, depth of field, etc.
- [5] E. Veach, December, **Robust Monte Carlo Methods for Light Transport Simulation**, Ph. D. dissertation for Stanford University, 1997, pp. 58-65
An online version can be found at http://graphics.stanford.edu/papers/veach_thesis/
Includes a description of low discrepancy sequences, quasi-Monte Carlo sampling and its application to solving the global illumination problem.
- [6] L. Szirmay-Kalos, **Importance Driven Quasi-Monte Carlo Walk Solution of the Rendering Equation**, Winter School of Computer Graphics Conf., 1998
An online version can be purchased at <https://www.sciencedirect.com/science/article/abs/pii/S0097849399000308>
Describes a two-pass method for solving the global illumination problem employing quasi-Monte Carlo sampling, as well as importance sampling using low discrepancy sequences.