Autodesk VRED Tuning Guide for NVIDIA RTX Server

May 22, 2020
v1.0

Table Of Contents

EXECUTIVE SUMMARY

This document provides insight and best practices using NVIDIA RTX Server with Autodesk VRED for automotive batch and interactive rendering. Recommendations are based on actual customer deployments and collaborative internal testing.

- For an accurate RTX Server configuration, what information should I provide for a POC?
- What should I know to get the best results using NVIDIA RTX GPUs for VRED rendering?
- What are VRED recommended settings for an accurate GPU rendering benchmark?

AUTODESK VRED

Autodesk VRED™ is a 3D visualization application used to evaluate CAD data interactively and to create off-line photo-realistic images and animation in batch mode. VRED is the dominant visualization app used throughout the automotive design and engineering industry to evaluate the entire process from design to perceived quality. VRED is used for advanced, concept, and production interior and exterior design as well engineer confirmation for fit and finish. The VRED tools are optimal for both in-progress design review as well as final presentation using interactive ray tracing and analytic render modes.

NVIDIA RTX SERVER

NVIDIA RTX Server is a validated reference design for multiple workloads that are accelerated by Quadro RTX 6000 or RTX 8000 GPUs. When deployed for high performance virtual workstations, the RTX Server solution delivers a native physical workstation experience from the data center, enabling creative professionals to do their best work from anywhere, using any device. RTX Server can also bring GPU-acceleration and performance to deliver the most efficient end-to-end rendering solution, from interactive sessions in the desktop to final batch rendering in the data center. Content production is undergoing massive growth as render complexity and quality demands increase. Designers and artists across industries continually strive to produce more visually rich content faster than ever before, yet find their creativity and productivity bound by inefficient CPU-based render solutions. NVIDIA RTX Server delivers the performance that all artists need, by allowing them to take advantage of key GPU enhancements to increase interactivity and visual quality, while centralizing GPU resources.

ESC4000 DHD G4 is a high-density 1U server that delivers powerful performance for a wide variety of applications. It is powered by dual 2nd Gen Intel Xeon Scalable family processors and supports up to 16 DIMMs and up to four double-deck active or passive-cooled graphics cards in a compact 1U chassis. Its unique single-root design connected directly between the CPU and GPUs allows maximum graphics performance for AI and ML applications.

The Thunder HX GA88-B5631 is a 1U 4GPU server platform based on the 2nd Generation Intel Xeon Scalable Processor Family. It builds upon the success of previous generations by adding additional features that our customers have asked for over the years. Tyan engineers have compressed the available space

within the Thunder HX GA88-B5631 to include hot swappable SATA drives, additional x16 PCIex slot for NIC support up to 100GB/s such as Mellanox solutions and built in onboard 10GBase-T LAN ports.

NVIDIA QUADRO RTX

The NVIDIA Quadro RTX 6000 and RTX 8000, both powered by the NVIDIA Turing™ architecture and the NVIDIA RTX platform, bring the most significant advancement in computer graphics in over a decade to professional workflows. Designers and artists can now wield the power of hardware-accelerated ray tracing, deep learning, and advanced shading to dramatically boost productivity and create amazing content faster than ever before. The Quadro RTX 6000 has 24GB of GPU memory, whereas the RTX 8000 has 48GB to handle larger animations or visualizations. The artistic workflows covered within our testing for this reference architecture used RTX 6000 GPUs.

SOLUTION CONFIGURATION & DETAILS

Table 1 outlines the system configuration utilized to complete the rigorous NVIDIA NVQual verification along with the Autodesk VRED, Autodesk Arnold, and Teradici software packages all in line with the NVIDIA RTX Server validation process.

Table 1: Solution Components

COMPONENT	VENDOR & MODEL & QUANTITY	DETAILS
RTX Server Workstation System	Tyan Thunder HX GA88-B5631 1x	• CPU: Intel Xeon Gold 6126 x1 • Memory: 384GB DDR4-2933 • Storage: 1TB Kingston DC500M SSD
RTX Server Render Node(s)	ASUS ESC4000 DHD G4 4x Find full list of RTX Server validated systems here: https://www.nvidia.com/en-us/design-visualization/quadro-servers/rtx/	• CPU: Intel Xeon Gold 6126 x2 • Memory: 384GB DDR4-2933 • Storage: 1TB Kingston DC500M SSD
Graphics	2x (Tyan) / 4x (ASUS) Quadro RTX 8000 (Passive) Quadro Driver Release: R440 U6 (442.50) or Later	• GPU Memory: 48GB • CUDA Cores: 4,608 • Tensor Cores: 576 • RT Cores: 72
Application / Software	Autodesk VRED 2021	

POC QUESTIONNAIRE

NVIDIA continually performs VRED GPU rendering tests on RTX Server. Based on your render workload feedback we can provide an RTX Server configuration that would best fit current and future rendering requirements.

- What would you like to learn from your POC?
- What metrics would you like to confirm from your RTX benchmark testing?
- Are your CPU render nodes used primarily for interior or exterior renderings? Percentage breakdown?
- Are CPU render nodes supporting interactive as well as batch rendering? Percentage breakdown?
- Do the CPU render nodes adequately support interactive VRED workflow during work hours and batch rendering during the evening and weekends?
- Number of interior and exterior render jobs per week?
- Number of frames per render job?
- What resolution do you render content?
- How often do you experience render jobs scheduled at the same time?
- Do you anticipate an increase in VRED rendering for interactive and/or batch rendering?

GPU RENDERING 101: USE CASES AND COMMONLY USED SETTINGS

https://help.autodesk.com/videos/13438b90-6ed3-11ea-815e-b5e476fe406d/video.webm

VRED GPU Cluster Mode and Workload

VRED cluster rendering distributes workloads evenly on available GPUs in a cluster. It does so by dividing the frame into equally sized tiles and then assigns bundles of tiles to individual GPUs in the cluster using a sophisticated load balancing algorithm. This way, all GPUs in a cluster contribute to the same render frame at a time. Each individual tile potentially generates a different workload depending on the scene section it is mapped to. Some tiles might only cover the scene background or environment with relatively low shading complexity while other tiles cover materials with very high shading complexity. The load balancing takes care of gathering different tiles with different shading complexity into tile bundles in a way that each tile bundle of a render frame has approximately the same shading complexity.

In a multi-GPU setup, like a GPU cluster, the scaling across the GPUs is optimal (close to linear) if the GPUs stay busy, that is, if the workload for each individual GPU is constantly at 100%. Underutilizing GPUs leads to suboptimal scaling as the overhead added by distributing the workload to the cluster nodes (GPUs) and sending back the rendered tile bundles to the cluster master for final compositing and post-processing might become more visible.

Scaling & Performance

Ray tracing is essentially separated into geometry processing (building acceleration structures and traversing these acceleration structures (tracing)) and shading (calculating the final color of a pixel). With our latest GPU generation (Turing), we introduced new hardware units, called RT cores, to massively accelerate the geometry processing part. Compared to our previous generation GPU (Pascal), the geometry processing of ray tracing has been accelerated by a factor of 10. The shading part is performed on the streaming multi-processors (SMs).

Note that render performance and scaling across GPUs are two different things. Complex scenes, both in terms of geometry complexity and shading complexity, render at lower frames per second (FPS) than less complex scenes. However, complex scenes show a more efficient scaling across GPUs than less complex scenes. Less complex scenes might not even benefit from more GPUs at all as the overhead introduced by the cluster processing and resource handling might become predominant as too much of the GPU resources (RT cores and streaming multiprocessors) left behind sitting idle.

Underutilization of GPUs and, with this, poor scaling across GPUs and GPU nodes can be countered by increasing the quality of renderings. Without modifying the content in a costly preparation there are options to increase the display resolution or to increase the number of image samples used with rendering, or both.

Resolution

The display resolution has a bigger impact on scaling as higher resolutions generate a higher workload.
While FullHD (1920x1080) is widely used today, 4K resolution (3840x2160) will become the next standard, in particular with automotive design workflows. 4K (~8.3M pixels) is 4-times the amount of pixels than FullHD.

Higher display resolutions, like 4K, 5K, 8K, generate more shading workload accordingly.

Anti-Aliasing

Anti-Aliasing is a technique in computer graphics intended to reduce the aliasing effect inherent to digital image synthesis. One approach used in particular is multisampling. Translated to ray tracing this means that multiple primary rays (according to the AA setting) are traced per pixel and can result in a smoother surface edge or visual separation between surfaces with different materials.

A higher number of image samples (AA settings in VRED) will also generate a higher workload per GPU and therefore helps better scaling across GPUs. As a side effect, a higher number of image samples will also lower the data traffic between render nodes and the master. Rendered tiles will be sent back to the master when all image samples have been processed. So, as an example, if you render with just one image sample per pixel, the tiles will be sent back after each iteration. With 16 image samples per pixel, tiles will only be sent back after the 16 iterations have been completely processed.

Batch vs Interactive Rendering

Batch rendering is offline rendering without immediate visual feedback. This technique is used with high quality final frame renderings or movies with 256 samples per pixel or more. Batch rendering jobs usually take from minutes to hours. Output is written to a file (image or movie) rather than presented on a display. Batch rendering, as compared to interactive rendering on a cluster, does not generate as much data traffic between the render nodes and the master and therefore shows less fluctuations with frame times (time to render a frame) and scaling across GPUs.

Interactive rendering uses way lower sample rates to maintain interactivity while navigating the scene. Usual sample rates used are between 1 spp (sample per pixel) and 16 spp. Each frame rendered on the cluster will be transferred to the master for compositing, post-processing (e.g. optional denoiser, tone mapping), and presented on a display. Interactive rendering generates more data traffic between master, render nodes and vice versa. Therefore, it shows way more fluctuations in frame times and scaling across GPUs.

Due to the much lower sample rates used while interacting with the rendered scene, the rendered frame shows noise, that is, the frame is not fully converged. The lower the sample rate (e.g. 1spp vs 4spp vs 16spp) the more noise is visible. As soon as the interaction with the scene stops (mouse up), the renderer renders more iteration for the image to converge if the VRED still-frame antialiasing option is enabled. There also exists a delay option on how much time the still-frame AA kicks in. Immediate is not by default..

VRED comes with an optional AI-based denoiser. AI is good at filling in missing information, and that is what noise represents. However, the noise still visible at 1spp is too much missing information for the denoiser to generate

decent visual quality. At 16spp the results generated by the denoiser tend to be much better but still there might be undesirable visual artifacts. The denoiser, though, is subject to being developed further to generate more satisfactory results in near future.

VRED SPECIFICS FOR GPU RENDERING

With the first version of GPU accelerated rendering in VRED 2021, Photon Mapping is not supported. GPU ray tracing will always use path tracing. Please contact Autodesk about when Photon Mapping will be supported for GPU ray tracing.

Any forms of Sampling Quality Overrides should be avoided with the GPU ray tracing. Light sampling works completely differently on the GPU compared to the CPU. Sampling Override will therefore be completely ignored and would yield different visual results. For the GPU ray tracing it is recommended to adjust the image samples (AA settings) to control the amount of noise.

Reflection Overrides are not implemented for the GPU ray tracing. This is because the GPU ray tracing algorithm works iteratively, not recursively, like with the CPU implementation. Increasing the number of image samples is recommended to reduce the amount of noise for GPU ray tracing.

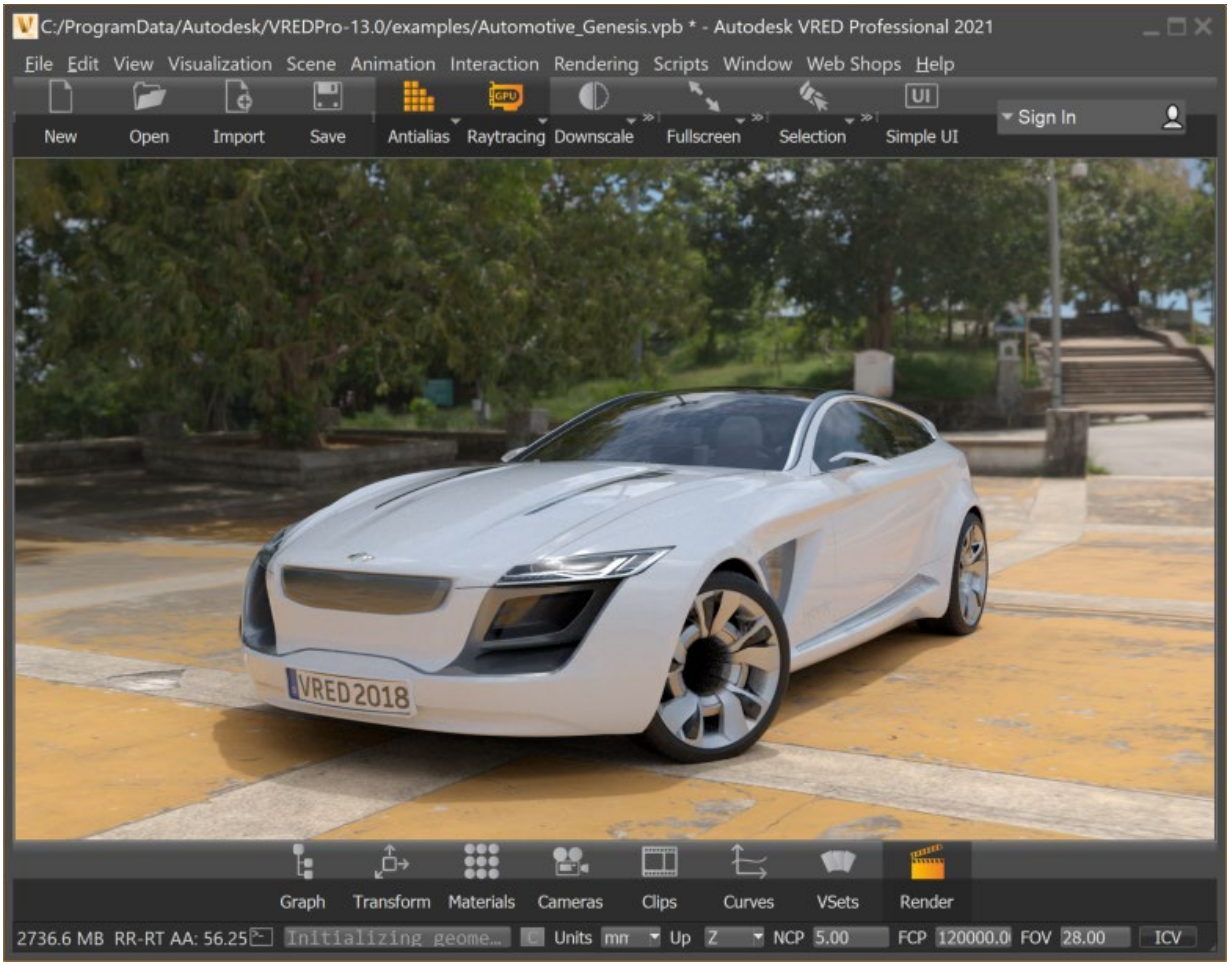
The "Use-Two-Sample MIS" setting should be turned off for better interactivity (>30%) at the cost of more noise due to fewer light samples being used. The noise is due to a threshold setting in VRED 2021 and will be addressed with the next VRED update.

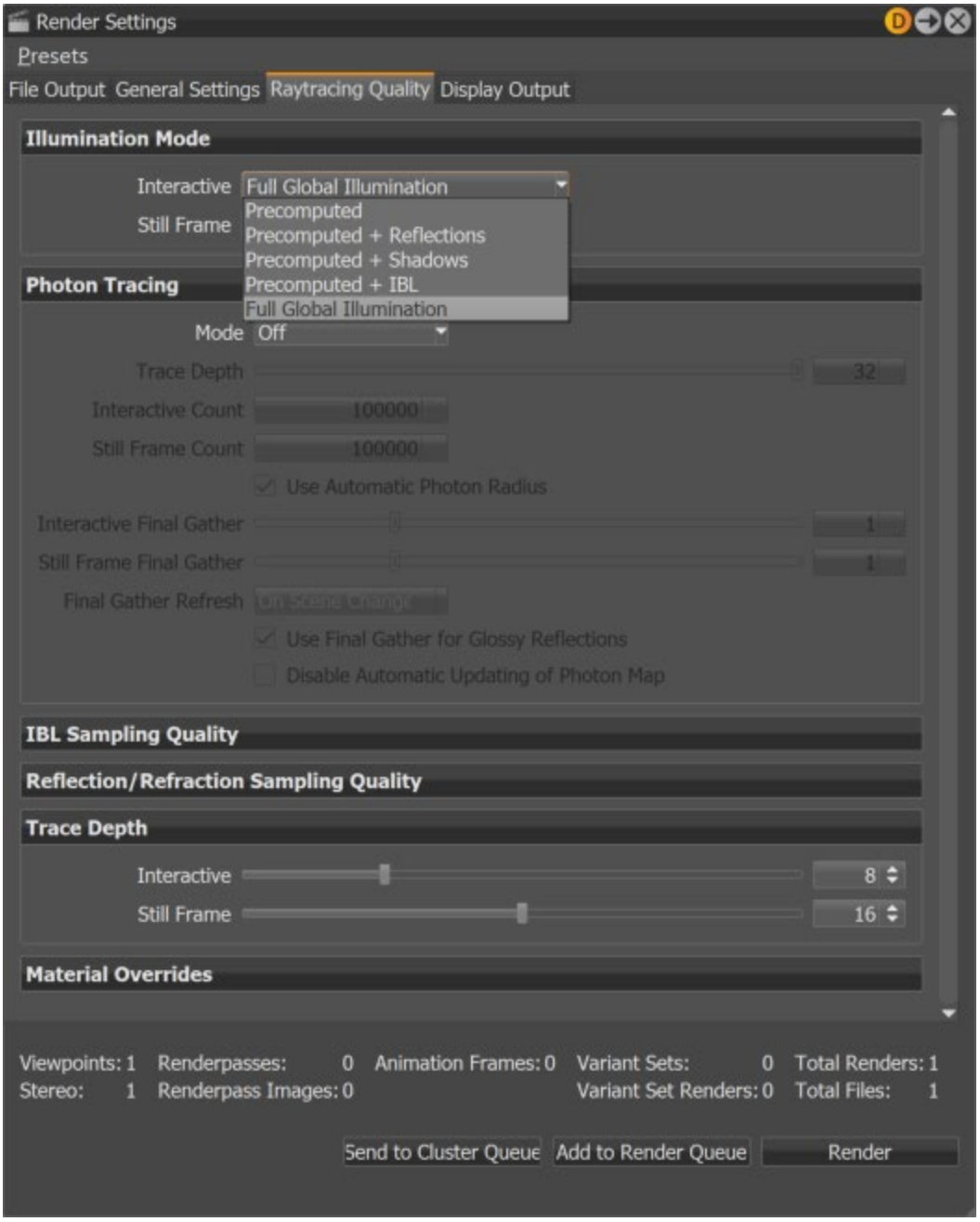
Use Ray Splitting should also be turned on.

Final objective is to have the same visual results for both CPU and GPU ray tracing. There are still some known issues, though, with the current implementation of the GPU raytracing in VRED plus some missing features as mentioned. These issues will be addressed with the next VRED update. However, if the visual results of the GPU ray tracing are too much off the results from the CPU ray tracing, it is considered a defect, and therefore it is recommended to report these back to Autodesk.

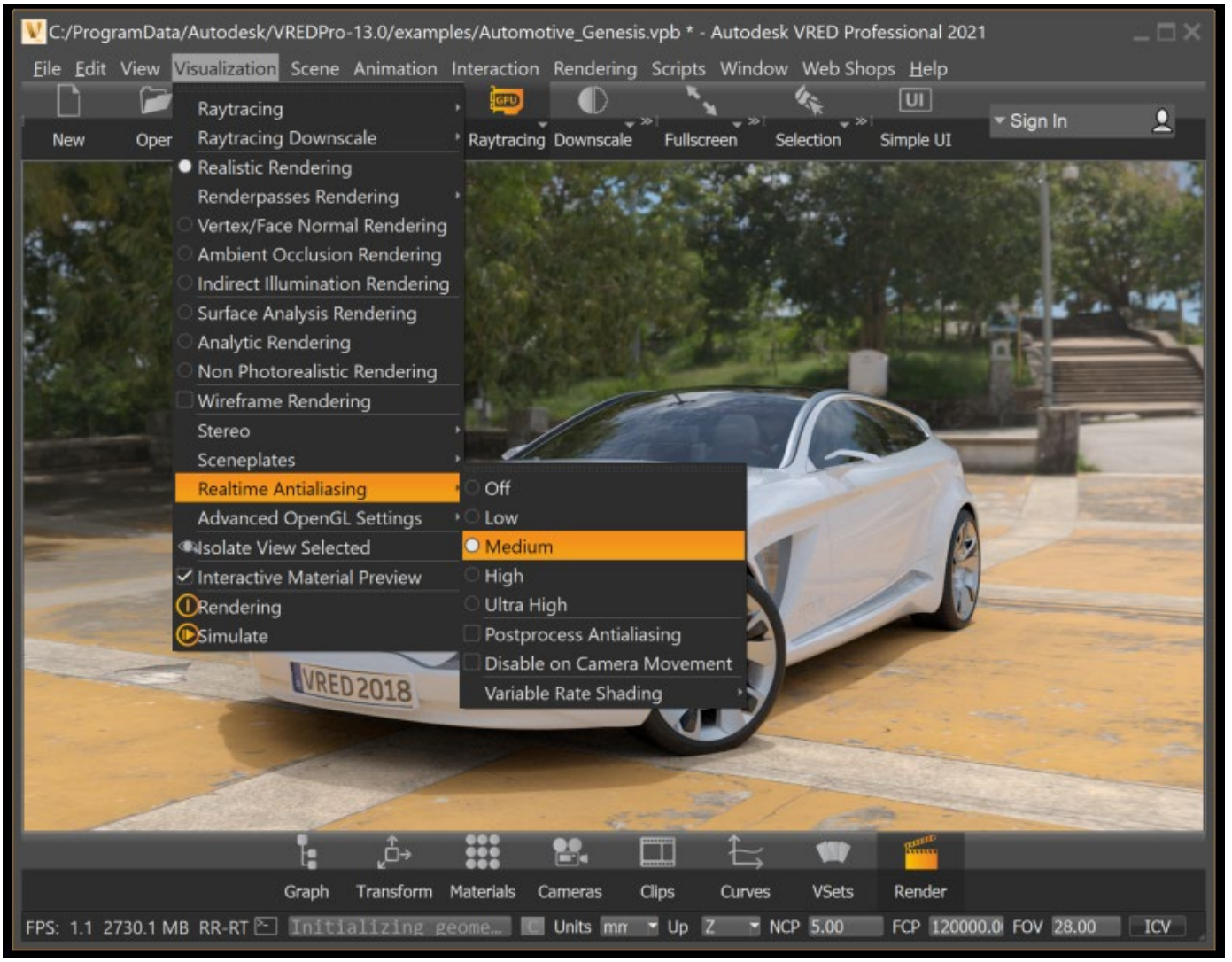
SETTING UP GPU ACCELERATED RENDERING IN VRED 2021

- Open the **Render Setting** window with the **Render** button on the bottom ribbon

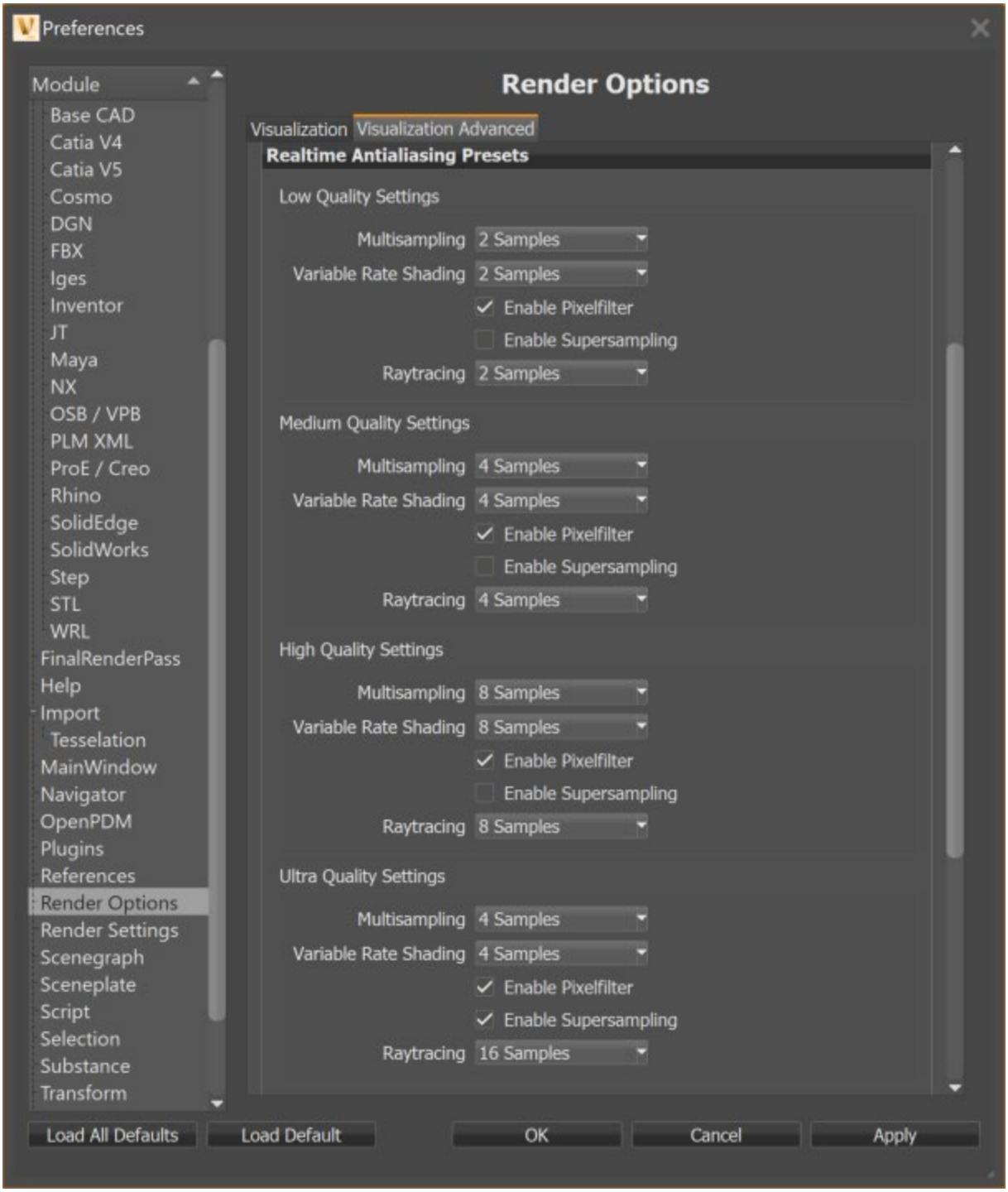

C:/ProgramData/Autodesk/VREDPro-13.0/examples/Automotive_Genesis.vpb * - Autodesk VRED Professional 2021
File Edit View Visualization Scene Animation Interaction Rendering Scripts Window Web Shops Help
New
Open
Import
Save
Antialias
Raytracing
Downscale
Fullscreen
Selection
Simple UI
Sign In
VRED 2018
Graph
Transform
Materials
Cameras
Clips
Curves
VSets
Render
2736.6 MB RR-RT AA: 56.25
Initializing geome...
Units
mm
Up
Z
NCP 5.00
FCP 120000.0
FOV 28.00
ICV



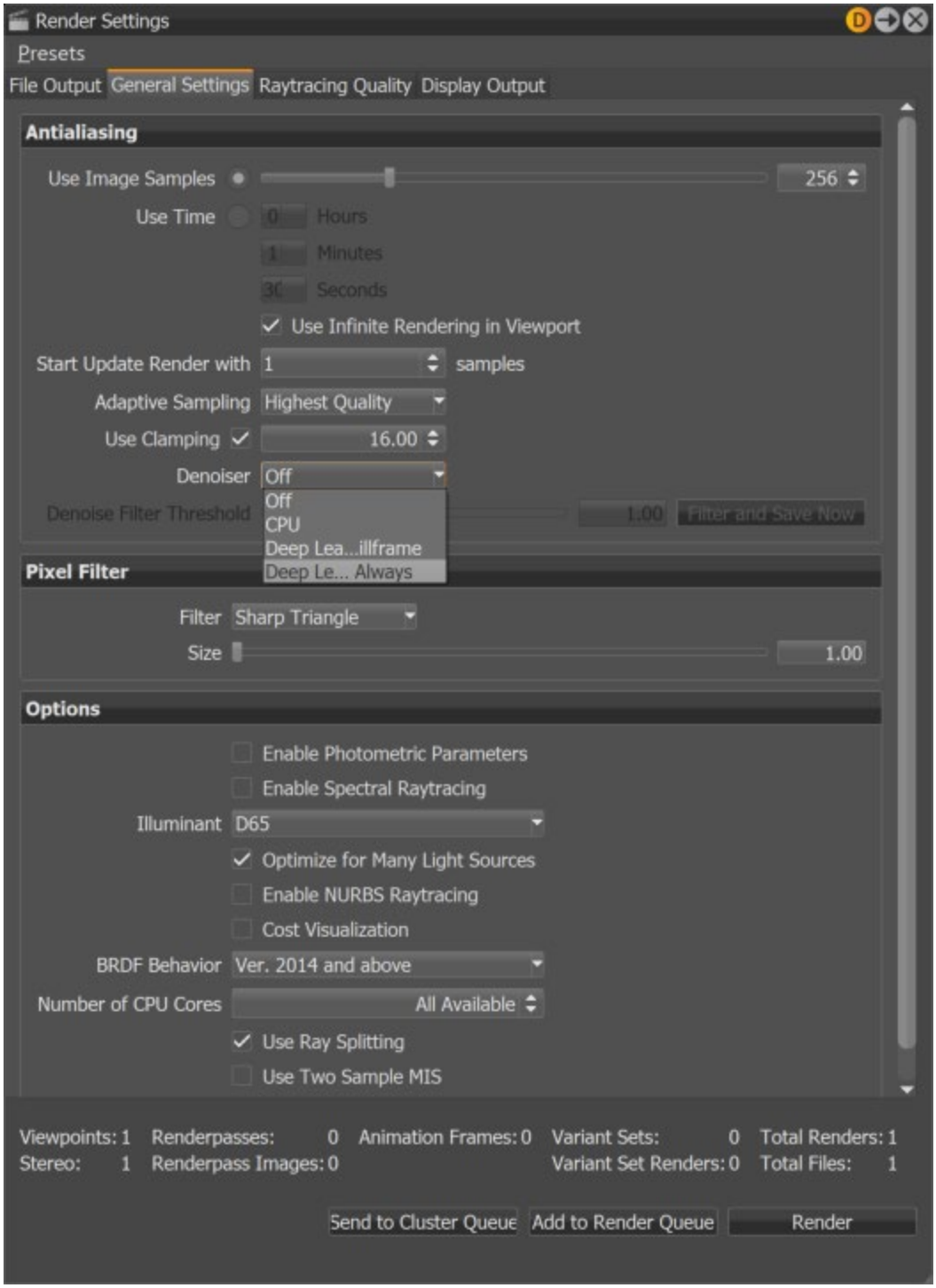
- On the **Raytracing Quality** Tab set the **Illumination Mode -> Interactive** to **Full Global Illumination**
- To enable GPU Raytracing left click and hold the **Raytracing** button on the Top ribbon and select **GPU Raytracing** (or with **Visualization -> Raytracing -> GPU Raytracing**)
- Select **Antialias** on the Top ribbon with the button right next to **Raytracing** to enable progressive refinement of the raytraced image



- Select **Visualization -> Realtime Antialiasing -> (Low, Medium, High, Ultra High)** to increase the number of raytraced samples per pixel



- The number of Raytraced samples that correspond to **Low, Medium, High and Ultra High** is under **Edit -> Preferences -> Render Options -> Visualization Advanced**. The default values are **Low = 2 Samples, Medium = 4 Samples, High = 8 Samples, and Ultra High = 16 Samples.**



- If you prefer a denoised image you can enable the NVIDIA AI Denoiser in the **Render Setting** window in **General Settings** Tab set **Antialiasing -> Denoiser** setting to **Deep Learning Always**.
- **Start Update Render with** can be set to **1 samples** to see the immediate effect of the Denoiser

FAQ

How does a low to medium complex model impact performance scaling across multiple GPUs and GPU render nodes?

Models with low or medium complexity generally tend to underutilize the GPU and hence leaving the participating hardware resources like RT cores and streaming multiprocessors sitting idle. This causes a disbalance between the pure rendering task (geometry and shader processing) and the cluster resource management. GPU utilization

ideally should be above 90% to yield an above 80% scaling across GPUs and GPU render nodes. If the utilization falls below 50% GPU load, you likely won't see any benefit from adding another GPU.
GPU load can be increased by rendering to a higher resolution (e.g. going from FullHD to 4K or even higher), increasing the number of image samples (AA settings), or increasing the bounce depth. Benefit doing so is a higher quality rendering and a faster noise elimination.
Purely indirect lighting conditions tend to be computationally way more intensive than direct lighting conditions. With automotive visualization, for example, you might notice that interiors challenge the GPUs more than exteriors, and hence, show better scaling across GPUs.

When is the Denoiser feature recommended and when not?
The NVIDIA denoiser that comes with VRED is AI based and uses the Tensor Cores on the GPU. While AI is good at filling in missing information it still requires some level of information to generate acceptable results. The level of noise is dependent on the samples per pixel (spp) used. At 1 spp you get highest interactivity but a very noisy image, like the image on the left below. If you apply the denoiser on such an image, the results likely are not acceptable as you end up with watercolor like smearing effects and loss of geometric details.

If you spend more samples per pixel rendering you will end up with less interactivity but also with less noise. The left image below is rendered at 16 spp and the denoiser applied to this image gives much better results as you see with the image on the right below.

The rendering algorithm is modeled after physical laws of how light interacts with surfaces and materials. The denoiser results, however, are an approximation. So, in the end it is a matter of taste whether the denoiser generates acceptable results for you or not. The denoiser is also subject to improvements. For the time being, this is a research topic where we experiment with different approaches, like overfitted training, and additional input like surface normals to better preserve geometric details, which seems essential for automotive design.

Does a higher-clocked CPU and a higher number of CPU cores benefit GPU render performance?

No. GPU render performance is completely independent of the CPU used in the individual render nodes. The only recommendation is for the CPU to have at least as many physical cores as the number of GPUs in the system. This is recommended because the data distribution to the individual GPUs in the system is performed multi-threaded, that is - one CPU core feeds one GPU. The distribution of data belongs to the load process and is decoupled from rendering.

Do my CPU render node licenses work with GPU render nodes?
Autodesk VRED CPU render node licenses do not support GPU render nodes. A GPU render node license is required. Please contact your Autodesk account representative for more details.

What is the performance and Total Cost of Ownership (TCO) difference comparing VRED CPU to GPU rendering?
GPU rendering performance takes many factors in to consideration: model size, # of polygons, image output resolution, and if interior or exterior rendering shot.

Optimal GPU rendering performance is 7X faster compared to a dual-CPU render node.
When comparing equal CPU render node to GPU render node performance, GPU render node TCO is 50%-67% less.

Comparison Specs:
GPU render node: Intel Xeon 2x Gold 6148 2.4 GHz 20C with 2 RTX 6000 GPUs
CPU render node: Intel Xeon 2x Gold 6126 2.6 GHz 12C

Who do I contact to learn more about RTX Server, set-up and support?
Please contact your local NVIDIA account manager for more information.

Notice