

V-Ray: Render optimization



INTRODUCTION

This brief tutorial's aim is to help you optimize a rendering in 3ds Max using V-Ray. I can't guarantee these methods will work every time, but in general they do the job. This should help you cut off hundreds of megabytes used for your renderings, as well as time. I won't go into detail and bore you with technical stuff that you don't need to know, I'll focus on keeping this as brief and helpful as possible for everyone.

I am a long time V-Ray user, nearing 5 years, and still, doing some reading and experimenting I still learn a lot of new things. Please don't expect me to cover everything there is to know about V-Ray and optimization because... I won't... ;)

Let's get to it!

OPTIMIZATION

Start up by opening the V-Ray Render Setup.

Under **Common Parameters** set a low resolution, you won't use this anyway.

Scroll down to the bottom and uncheck the box **Rendered Frame Window**.

Next up is the **V-Ray:: Frame buffer**, uncheck the box **Get resolution from MAX**, and type the desired resolution.

This creates a savable preview using **VFB**.

Optionally you can also turn off the **VFB** to save even more memory.

Now go to the **System** rollout.

In this section, tweaking is crucial. The settings here vary from project to project, just keep that in mind.

The first thing we'll look at is the option **Max. Tree Depth**, default value: 60.

This option affects memory in a simple manner. Higher values consume more memory and lower values consume less. Not only that, high values will also speed up your rendering and vice versa. Just keep track of this setting and don't overdo the values.

The next option, **Face/level Coef.**, default value: 2.0.

There are some controversies around the coefficient, I would say this value more than the others is scene dependent.

In most cases though this option follows the same pattern as the **Max. Tree Depth**, Higher values consume more memory and lower values consume less. Same thing with the speed.

An important setting is the **Dynamic Memory Limit**, default value: 400 megabytes.

Taking myself as an example, I use a Quad Core combined with 4 gigabytes of ram and a 64-bit operating system. This means I can use up to 3000 megabytes, but I found that what works best for me was 2500 megabytes.

I advise you to tweak this value, to find what works best for you.

Now, **Render region division**, default value: X64 x Y64.

This section affects the rendering buckets you see e.g. when prepasses are being done with **Irradiance map** or rendering with **Brute force**.

These buckets are best kept in binary sequence; 64, 32, 16...

In most cases 64 works best.

The smaller the buckets the longer it will take, and sure enough less memory is being used, and the other way around.

V-Ray: Render optimization

This method takes a bit longer, but is a great way to save memory on low end machines. It is essentially a two part process, and is dependent on the type of GI being used.

For this example I'll use *Irradiance map* as primary and *Light cache* as secondary.

Go to **V-Ray:: Global switches** and check the box **Don't render final image**.

Next go to the **Indirect illumination** tab and check the boxes **Auto save** and **Switch to saved cache/map**.

Now render. Once the rendering is done you might be asked to open the files, do so. Go back to **V-Ray:: Global switches** and now uncheck **Don't render final image**. Now you won't see it doing anything besides rendering. Optionally, you can combine this method with others, like turning of the **VFB**, again to save even more memory.

If you have a scene with many similar objects or just one heavy duty, **VrayProxy** is a great way to save memory.

Select the object you want to use, right click and choose **V-Ray mesh export**. depending on what you want, change some of the settings. Remember to check the box **Automatically create proxies**. This makes sure an instance is placed in your scene.

VrayProxy can be combined with an amazing plugin called **VrayScatter** which automatically places as many instances as you'd like in your scene.

One last tip, which is kind of annoying. 3ds Max, seems to hold on to memory, this memory is the taken out of the equation when you want to render. A good thing to do is to restart Max, this frees up the memory. The amount can be considerably high at times.

As I said, this would be a brief tutorial. These small tweaks can unload hundreds of megabytes of memory from your rendering.