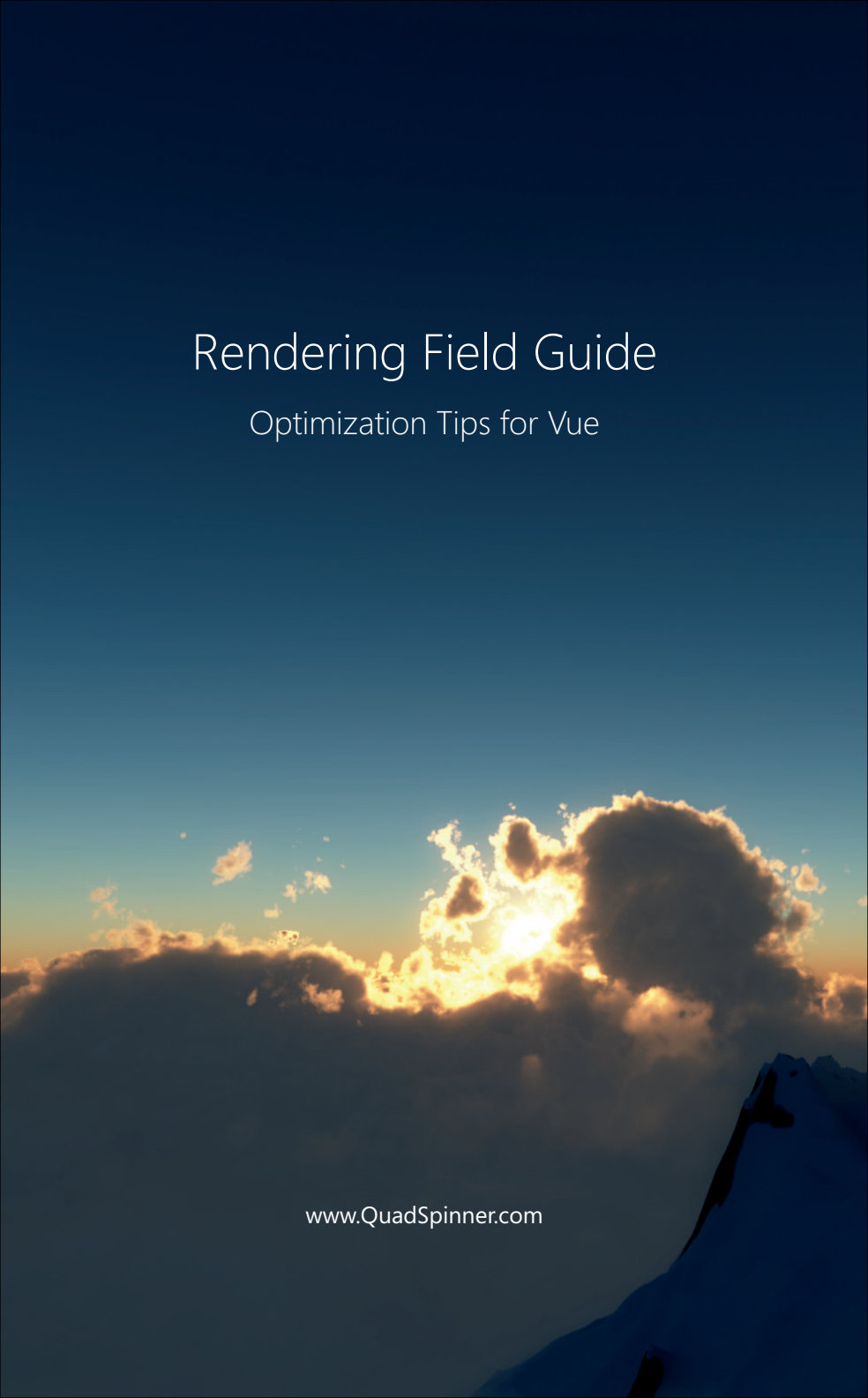
Rendering Field Guide

Optimization Tips for Vue



www.QuadSpinner.com

Contents

"It is theoretically impossible to have both speed and quality in renders. But these techniques will get you as close to that balance as possible."

The simplicity of the Vue interface, and the ease-of-use with which the user can create sophisticated worlds, belie the staggering complexity of the software. Most quality and speed bottlenecks encountered can be overcome by understanding Vue's principles which are quite different from other 3D packages. This book presents a quick reference guide to the most common scenarios and practices from the Master Class, for constructing and troubleshooting a render.

Dax Pandhi

Antialiasing

The following antialiasing scenarios are very context sensitive. Their reaction may vary based on the scene's complexity. The best practice is to try them out one at a time on single frames, and choose the best result. The time spent on this process will result in large time savings for the final animation render.

Optimized vs. Systematic Antialiasing

The main difference between the two types of antialiasing for objects is: **Optimized AA** will try to intelligently decide what portions of the render require supersampling, and ignore those that do not; **Systematic AA** will force all portions of the image to be supersampled.

For most situations Optimized AA is recommended. Systematic AA can greatly increase render time in scenes with complex objects, EcoSystems, and Clouds.

Systematic AA can be useful for simpler scenes or when sharp physical details may need to be preserved. To preserve texture details, use higher **Texture Antialiasing** instead.

Object Antialiasing (OAA)

OAA subrays can often be misleading. While they do work in a systematic manner in theory, actual contents of the scene will shift their range of usefulness.

Vue will fire the **Minimum** amount of subrays into every supersampled pixel. If the engine judges that the pixel needs more antialiasing, it will fire more rays. If the **Maximum** amount of rays are fired, then the engine will not fire any new subrays.

For best performance, test key frames of your animation with different AA settings ranging from low to high, but in steps. For example, use 1/1, 1/2, 1/4, 1/8, 2/6, 2/12, 3/3, 3/9, 6/12, etc. For strong AA, use high values such as 8/8, 12/32, 32/48, etc.

The point being, if a lower values works, don't go higher. It is better to raise the Minimum, than to keep the Minimum low and the Maximum high. A low or medium range Maximum paired with a strong Minimum will often result in better antialiasing. For example: 4/8 may get better results than 2/12.

While the engine can judge things from a logic-fulfillment point of view, your precision will be based on visual impact. This fine tuning process can save hundreds of hours for animations.

Texture Antialiasing (TAA)

For many objects, especially plants, if you see flickering or aliasing of any form, increasing Texture Antialiasing is recommended over Object Antialiasing. (See Texture Filtering for critical information)

The principles behind the TAA texels is the same as OAA subrays. Start with low pairs and increase as needed. Often using a higher TAA can allow you to lower your OAA in situations where the flickering comes as a result of texture maps used on trees or highly detailed infinite procedural textures. By compensating the OAA in

respect to increases in TAA, you can achieve faster render times. In other cases where TAA adds overhead, reducing OAA can result in keeping the previous render speed more or less intact.

TAA is a very memory intensive process and can significantly reduce render times. The amount of time spent finding the minimum working texel range will greatly affect how much speed can be achieved for the final animation render.

Texture Filtering

When increasing TAA settings, keep the **Texture Filtering** value in mind. To reduce flickering, in many cases high Texture Filtering values are better than high TAA texel ranges. TAA texels per pixel settings are for preserving texture quality, while Texture Filtering provides a way of blurring texture maps which results in less detail and subsequently - less flickering.

For large scale EcoSystems, higher Texture Filtering and less texels may result in a good balance of visual quality and render speed.

Antialiasing Strategy

The rule of thumb is: the higher the frame size, the less antialiasing you need. This does not mean that a 4000px frame should use minimum antialiasing. It means that the amount of sharpness as decided by Vue's **Anti-aliasing Stategy** setting gets lower. AA Strategy ranges from **Crisp** to **Blurred.** In most cases, the default is **Automatic**, but by choosing the right kind of AA Strategy, your render speed and visual quality may be impacted quite strongly.

The smaller the frame size, the higher the quality (sharper AA) you might need. Useful animations tend to be 480p or higher. If your frame size is above 1000px in width, **Soft** is your best bet. This strategy requires less AA samples resulting in a slightly softer visual result and significantly faster render speed. The advantage of this particular strategy is that it will also make it easier for the animation to be integrated with live action.

This may seem untrue when thinking of 2000px or 4000px frames. As the size of the frame increases, it becomes more likely that CG objects will appear too pristine and sharp. For normal VFX this may work, but for almost all scenarios where Vue is used, a softer AA strategy results in better integration.

If film grain is added in compositing, a **Sharp** or Crisp AA strategy may result in too much grain or bad grain/edge interaction, while Soft AA will smoothly integrate the grain.

For atmospheric renders, where the main element of focus is the sky and clouds, the Blurred AA Strategy can often be very helpful. Clouds do not need to be rendered with too much sharpness, and in fact, Blurred AA may result in the clouds looking softer without having to increase the atmosphere's **Global Quality Boost**. This can result in very high rendering speeds.

Selective TAA

See next section for selectively setting Texture Antialiasing on a per-object basis using **TAA Boost**.

Matte Paintings + Compositing

All settings presented in this section were specifically optimized for animation rendering.

When rendering still frames for compositing or matte painting, please see the Antialiasing Appendix in the book **Realism in Vue**. (www.RealismInVue.com)

Render Boosts

There is no single way to maintain a balance between fast render speed and good visual output. However, with the techniques presented in this section, you can fine tune the balance to your needs by supplementing Vue's render engine with precise rules, rather than relying on automatic calculations which may not always provide the most intelligent results.

Disable Shadows

This applies to both individual objects as well as entire EcoSystems. Often certain objects will not require shadows and turning them off in the **Advanced Material Editor** will result in a dramatic speed increase for even the most complex scenes.

Limit EcoSystems to Visible Frame

When populating EcoSystems manually, place large cubes or other primitive objects on the area of the terrain that is outside of the visible frame. Use **Decay near foreign objects** in the EcoSystem to prevent instances from being populated in the areas that will never be in the shot. Hide the primitives from rendering to avoid lighting problems.

Clear OpenGL for Renders

Even though the majority of the load for displaying items in the viewports is handled by the GPU, complex objects such as HyperTerrains or imported objects may put some stress on the computer's memory. Check the render option **Clear OpenGL data before render** to ensure more memory is available.

Reuse Indirect Lighting

You can cut your render time in half when re-rendering the same scene with minor changes or rendering an animation, when you reuse the indirect lighting solution.

This topic is covered in greater detail in the next section.

Disable Unnecessary Render Settings

If your scene or a particular test of a scene is not specifically focused on reflections or transparency, you can turn off **Trace Reflections** and **Trace Transparency**, and in effect, **Blurred Reflections** and **Blurred Transparency**.

Use TAA Boost

Use the **Texture Antialiasing Boost** setting to either increase or decrease the amount of TAA for individual objects instead of unnecessarily increasing the overall TAA of the scene. With this you can manage to have important objects be antialiased to a high degree without suffering from a uniform TAA increase.

Use Subray Drop

If your object uses transparency in any form, Vue will take longer to process that object because subrays will take longer to interact beyond the transparent surface. If refracted visuals are not a priority, use **Subray Drop** in the object's material to speed up the render process.

Near-Zero Decay Limit for EcoSystems

If you are turning off **Decay Near Foreign Objects** to get greater coverage near overlapping objects, you may be weighing the scene down with unnecessary and invisible instances. Use a small value such as 0.5 or 1.0 for **Influence** to get close to the overlap edge without creating too many instances.

Projected Shadows instead of Volumetric

If you simply require shadows on clouds instead of **Godrays**, or **Volumetric Lighting** for the entire scene, switch to **Projected Shadows on Clouds**. The non-volumetric calculation speed is several magnitudes faster.

Avoid Infinite Detail

When creating procedural textures or objects, avoid using **Smallest Feature** values under 0.25. Anything below 0.1 can be theoretically something that Vue's engine may decide to process, even though it will not add anything special to the visual result. By cutting off the minute detail level, the renders will become faster and flickering may be removed without resorting to TAA.

Delete Hidden Objects

Before running the final render of an animation, especially when sending to a render node network, delete all hidden objects in the scene. The overall load of the file will decrease and memory usage will go down, freeing up more memory for the renderer.

Decimate Meshes

If your baked or imported object is far away, try decimating it as much as possible. The lower polygon count will result in less ray bounces and increase the overall render performance. You can hide low polygon 'defects' by creatively using fractal materials.

Disable Indirect Lighting for Plants

If your EcoSystem is very far away, try using the render option for **Disable Indirect Lighting on Plants**. If the result is acceptable, you may gain a significant increase in render speed, as indirect lighting calculation will ignore the plants as they are often the most memory intensive objects to render.

Bake Displacement

Live Displacement can often use a lot of memory to render. You can bake the displacement into the mesh to conserve memory. This is highly recommended for distant objects. This will increase the size of the Vue file, but the memory usage will go down considerably.

Bake Procedural Terrains to Standard

If you have a non-infinite **Procedural Terrain**, increase the resolution to whatever may be best for your scene and then convert it to a **Standard Terrain**. In most cases, the loss of minute procedural detail is easily surpassed by the significant gain in render speed.

Lower Terrain Resolution

For distant terrains, try lowering the terrain resolution of Standard Terrains. If the visual result is acceptable you can get a boost in rendering speed. To maintain high visual quality you may want to use a richly detailed material.

Motion Blur instead of Texture Antialiasing

If you have peripheral plants that don't play a significant role in your scene, you may wish to use **Motion Blur** instead of Texture Antialiasing to achieve smooth renders. This is mostly applicable to a scene where peripheral plants will undergo motion blurring.

Reset EcoSystem Trunk Materials

Replace distant EcoSystem tree trunk materials with a simple solid-colored material. Such materials require very little antialiasing effort, and conserve a lot of memory compared to procedural or bitmap textures.

Hide EcoSystem Trunks

For very distant EcoSystems you can reset the material of tree trunks and branches (if available) to a simple material with 100% alpha to make them completely invisible. Turn off the **Cast** and **Receive Shadow** options for the material as well.

Create Custom Low Polygon Species

Edit existing **SolidGrowth** plant species and lower their polygon count and also downsample the texture maps used for the materials. Save these species and use them in your distant EcoSystems for significant gains in render speed.

Lower EcoSystem Density

In many scenarios, you can gain an increase of 10% to 20% in rendering speed by lowering the EcoSystem **Density** by a few percent. While the visual may remain very similar to the original, the memory usage will go down significantly. Also, use **Avoid Overlapping Instances** whenever possible.

Beware of Populate Below Foreign Object

If the **Populate Below Foreign Object** option is checked, which is the default, your EcoSystem may be creating instances which are partially or completely inside other overlapping objects. This will cause unnecessary load on memory.

Disable Backlighting on Distant Plants

Backlighting can often be a memory intensive process when used for too many objects. Disable Backlighting on EcoSystems that do not necessarily require it, or are too far away to contribute anything significant through Backlighting.

Godrays instead of Volumetric Lighting

If you do not require full Volumetric Lighting where terrains and objects throw volumetric shadows, then select **None** or **Projected Shadows on Clouds**, and check **Godrays** instead. This will process volumetric shadowing for the clouds, but will not affect anything outside of the atmosphere.

Downsample Distant or Small Maps

If a bitmap texture does not play a significant role, Vue provides in-built capabilities to downsample high resolution maps through the **Links Browser**. Alternatively, you can also downsample the map independently to not add extra overhead to the renderer.

Evaluate Key Points in Animation

Using any of the techniques in this section can potentially result in unexpected render artifacts or errors - especially if you are not used to manually altering complex aspects of a scene, or render settings. The best practice for avoiding such problems is to render key points in your animation with (and without) these boosts. Key points are usually where the frame contains something significantly different.

By testing these aspects, you can determine what techniques work and avoid the need to re-render entire animations, or portions thereof, to recover the malformed render.

Indirect Light

Indirect Lighting through **Global Radiosity** (GR) is the most sophisticated Lighting Model available in Vue, but also the most memory intensive. This section describes methods of gaining speed in GR renders without sacrificing important details.

Managing Quality

Global Radiosity has a feature called **Quality Boost** which can go from **-4** to **+4** to compensate for quality.

For most still renders or quick animation drafts, **-0.4** or **-0.2** can provide faster results without degrading the indirect lighting quality too much. In final quality animations where indirect lighting contributes less than 20% of the lighting, such low values may work well.

The majority of animation renders will require **0.0** to **+0.5** to produce good looking results. If your ambient lighting needs are high, increase this to get better results. Quality Boost values of **+2** and higher may heavily increase render time and should be used very sparingly.

Reusing Indirect Lighting

During the render prepass, Vue will calculate the indirect lighting solution for the frame. The solution will contain data for everything that is visible in the frame. You can see this indirect lighting data if you render on-screen. If any changes are made to antialiasing, render settings, material changes, etc. during the animation, the indirect lighting solution will become obsolete.

The indirect lighting solution can be reused for subsequent renders, but may not always work. Effective scenarios include:

- **Camera zooms or tracks in.** Since everything that will be visible is already available in the first frame, the subsequent frames can reuse the data. If the camera tracks in too much to reveal previously hidden perspectives of an object, there will be bad or incompatible data for that portion in the indirect lighting solution and bad samples will show up in that space.

- **No moving lights.** If your animation is focused on clouds, or works with the atmosphere but does not directly contribute anything to the indirect lighting solution (including Sunlight that does not move or change properties), then the lighting solution can be reused. Moving objects or moving clouds with **Indirect Atmospherics** can break the solution.

- **Baked indirect illumination.** If objects that fall outside of the above considerations have indirect illumination baked within, then the first-frame indirect lighting solution can be reused.

Reuse Indirect Lighting for Animations

Render the first frame as a still at full quality and size. Close the render window, go to Render Options, check **Reuse Indirect Lighting** and save the file. With the solution preserved, render the animation normally. The indirect lighting solution will be reused.

Indirect Lighting Cheat: Vue 11 or older

For some simple scenarios, especially where the camera will not go 'under' any objects, you can create a global indirect lighting solution. Create a separate camera that is set to capture, in a single frame, everything the normal camera will see from the first frame to the last. Render a single frame with that camera and preserve the indirect lighting solution. Proceed to render as usual with the normal camera, which will now reuse the indirect lighting solution from the previous camera.

Warning: While this technique may save you a lot of time, anything that was not directly visible to the indirect lighting camera but is visible in the final render camera, will have grainy/splotchy indirect lighting. Render several test frames at key intervals in the animation before locking in the final render.

Distributing Render into Clips

A stop-gap solution for reusing and refreshing indirect lighting data is to divide the animation into several clips for rendering.

For example, depending on the camera movement, you can render frames **0** to **100** using the indirect solution from frame **0**. The camera moves to a new location from frames **101** to **145,** so indirect lighting should be calculated for all of those frames. Then reuse the indirect lighting solution from frame **145** for frames **146** to **200** where the camera is static.

Often a **5-10** frame overlap/re-render between such clips is recommended, so any random seed calculations between two different indirect lighting solutions can smoothly transition in post-production.

Quality Control

Procedural Terrain Quality Boost

This setting affects how a procedural terrain is built during a render. **Lower values** create a fast terrain at loss of detail, while **higher values** will produce more accurate results. Normally, the default **0.0** value will suffice, but use higher values if you see moving artifacts in animations.

Atmosphere Global Boost

This setting affects the overall atmosphere. The main visible change is in volumetric lighting as well as cloud quality. **-2.0** to **0.0** values may be sufficient for most tests, as well as for simple final renders. For proper cloud shape, coloration, and volumetric shadowing, values of **+0.5** to **+10.0** may be required.

Soft Shadows Quality

This setting affects the quality of soft shadows when Sunlight softness is set above **0.0**. As usual, **sub-zero** values can be used for testing, but **0.0** is recommended for final renders. In most cases, except when dealing with very fine details, you may not need to go beyond **0.0.**

Shadow Smoothing

This setting affects how shadows are processed in indirect lighting solutions. **High values** tend to create faster results but shadows may be less accurate. **Low values** will produce accurate results at the expense of memory and render time.

Lighting Model Quality Boost

This setting affects all lighting models except **Standard,** and controls the quality of indirect lighting. **Sub-zero** values may be used for tests, but **0.0** or higher is recommended for final renders.

HyperTexture Quality Boost

This setting affects how accurately volume-filling operations are conducted when a HyperTexture or HyperBlob is rendered. **High values** will be extremely slow to render but will produce very accurate results. **Low values** may cause jittering artifacts.

Displacement Quality Boost

This setting affects how accurately an object is displaced. With bitmaps this will control interpolation and smoothing of the map to fit the mesh; with procedural textures, it will control the faithfulness with which the fractals are subdivided to match the mesh. **Lower values** will help speed up a render when the displaced object is far away, while **higher values** can be useful for closeup objects, or when baking the displacement.

Advanced Effects

This setting affects advanced visuals such as reflective/refractive caustics, indirect lighting, and radiosity. This slider provides a simple interface to the elaborate Advanced Effects editing dialog. **Middle range** values are the norm, but very **low values** can be used for pre-viz or motion tests. **Higher values** will contribute to better indirect illumination, as well as enhanced processing of the atmosphere in general.

This edition was printed for Master Class: Los Angeles participants. The Rendering Field Guide provides a glimpse into the powerful techniques demonstrated in the 7+ hour live training.

Order the downloadable version of the Master Class for the complete training on optimization techniques for Vue.

For more information, visit us at **www.QuadSpinner.com**