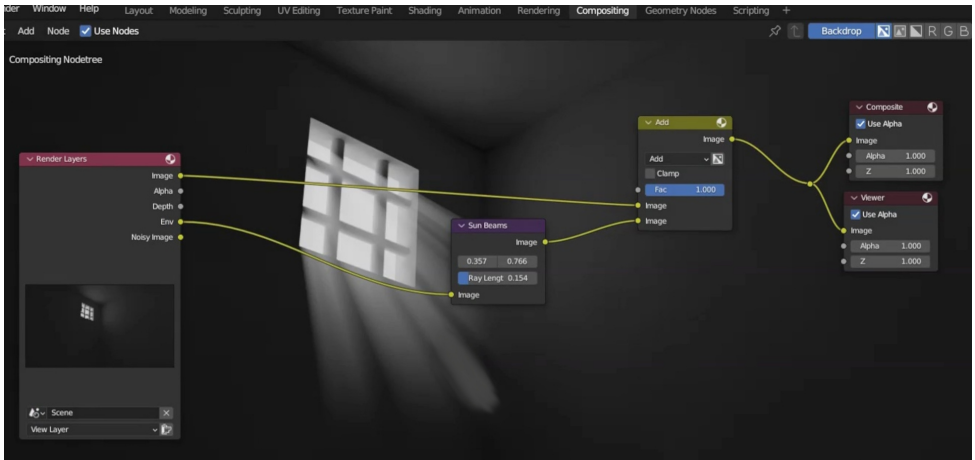


This PDF is available for free; feel free to share it!

[Subscribe to my YT channel](#) & turn on notifications to get updates on new videos!



Hello everyone!

This is the first of two tutorials, created using Blender 3.6, in which we will examine the Sun Beams node to simulate the **dispersion of light in the air in Compositing**, that is: after rendering, without resorting to Volume shaders or other elements in the scene.

This effect can be called in various ways: Crepuscular Rays, God Rays, or, because of the Compositing node we will be using, **Sun Beams**; in this particular case, I would avoid calling it Volumetric Light, because it is a post-production effect, not true volumetric lighting.

In this first tutorial, we will only look at the Sun Beams node with a very simple scene; so, if you already know this node and its parameters, you can stop watching this tutorial and wait for the next one!

In the next tutorial, indeed, we will examine a slightly more complex scene, in which we will need to use masks and other compositing techniques.

In the scene I'm showing right now on the screen, we have the virtual camera placed inside a very stylized room. The light enters only from the opening at the top, which is the World lighting, which is called **Environment** in Compositing.

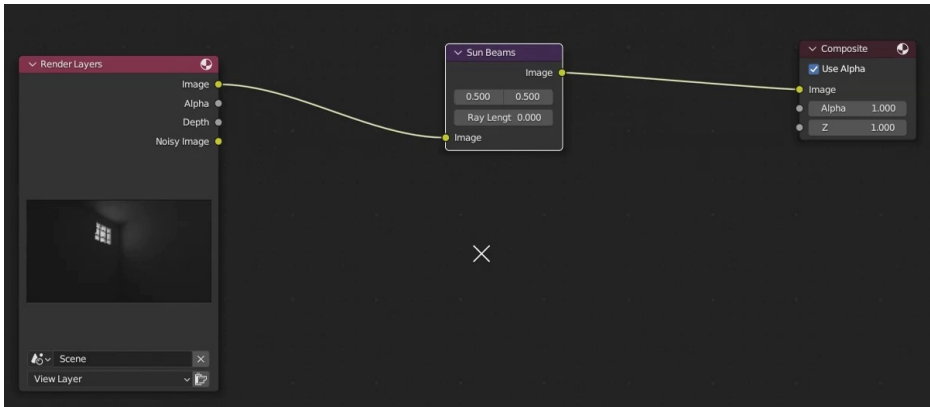


When we start rendering, we get a very simple result: the background visible through the opening is well lit, while the rest of the environment is lit by bounced light.

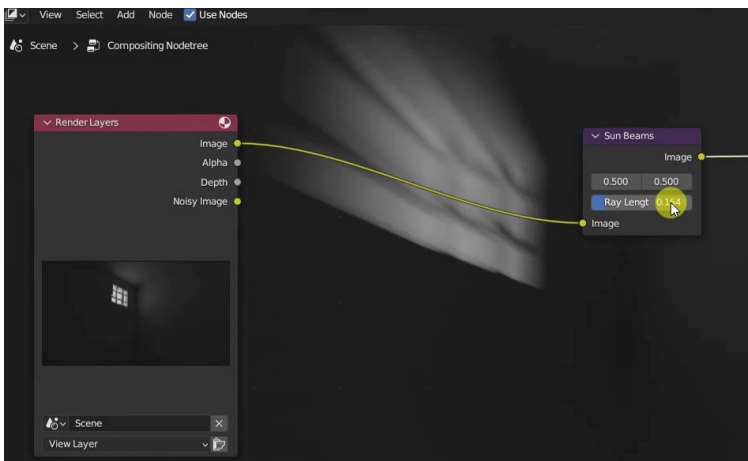


Let's open the Compositing editor and **add a Sun Beams node** to the setup, placing it right between the Render Layers' Image output and the Reroute node, so we can have a preview of the effect in the Backdrop.

Immediately, we notice two things: the node has only three parameters, and... **there's nothing visible**: not only is there no effect visible, but the entire rendering has **disappeared**!

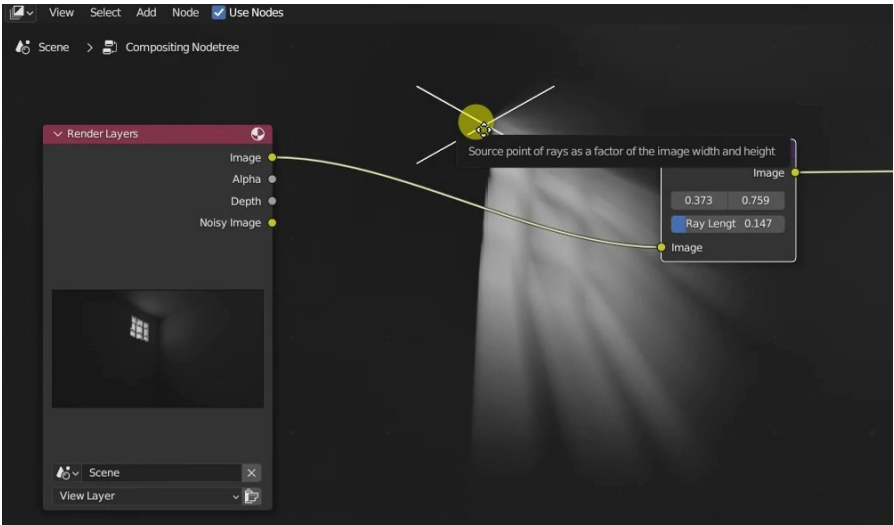


The Sun Beams node is applying itself to the entire rendering, but since the length of the rays, set in the **Ray Length** parameter, is **0**, there is no effect. So, let's **increase the value** of the Ray Length parameter and notice three new things.



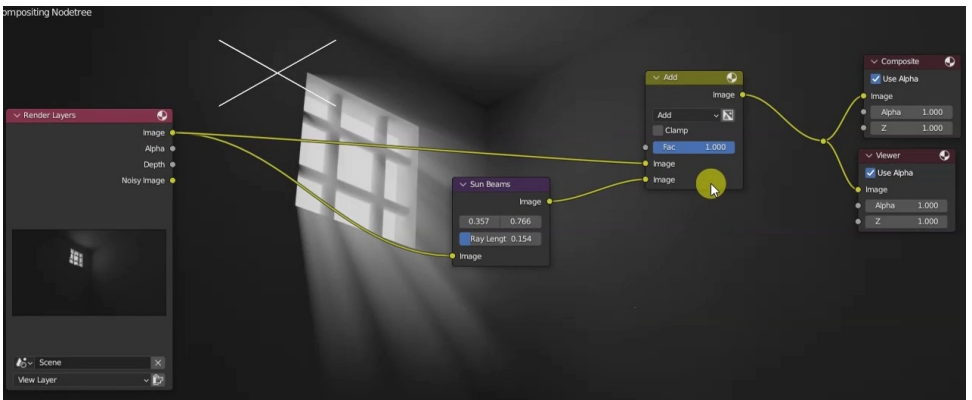
The first thing we notice is that the rays have their own **direction**, which, however, does not coincide with the direction of the incoming light rays into the scene. This direction is actually given by the position of the **X-shaped element** visible in the editor when we select the Sun Beams node.

This symbol, representing **the origin of the effect**, can be repositioned by clicking and dragging or by entering the X and Y coordinates in the remaining two parameters of the node.



The second thing we notice is that the Sun Beams node returns only the effect, meaning that **the rest of the rendering is no longer visible**.

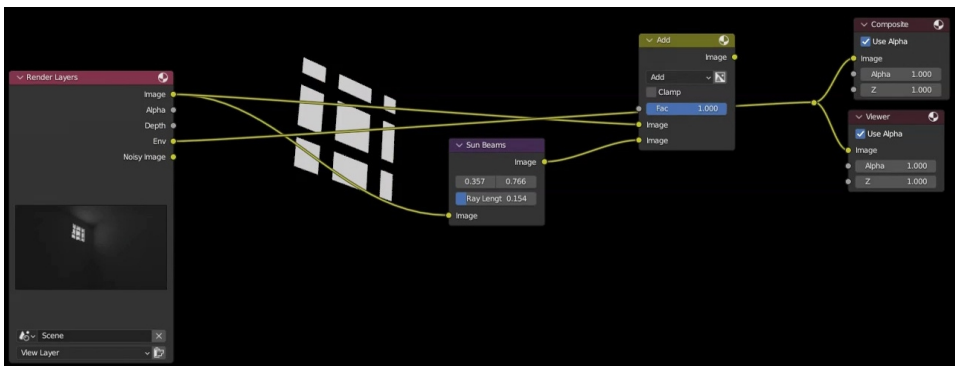
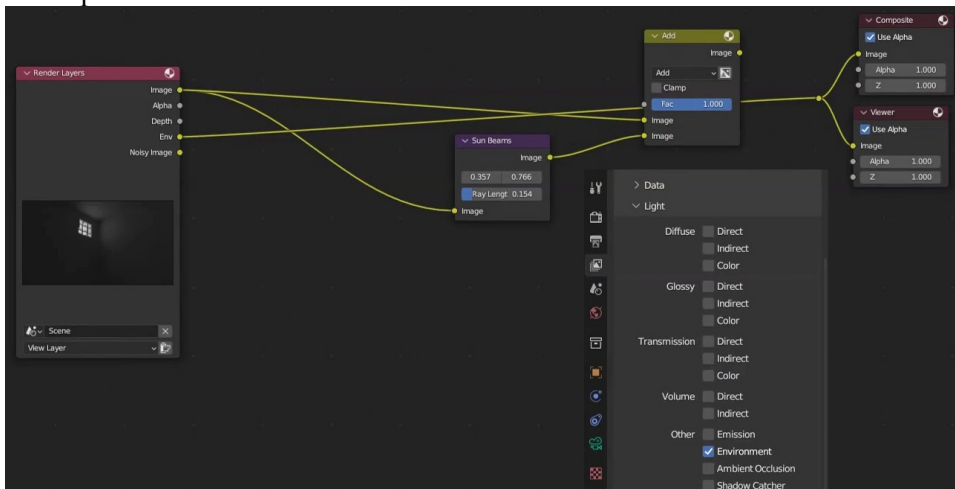
To add this effect to the original rendering, all we have to do is add a **Color Mix** node in **Add** mode to the scene, making sure to connect the Render Layer's Image output to the first Image input of the Color Mix node, and to connect **the Sun Beams' Image output** to the **second Image input** of the Color Mix node.



However, there is a third thing we noticed earlier, namely that the light rays are **projected omnidirectionally** from the point of origin of the effect.

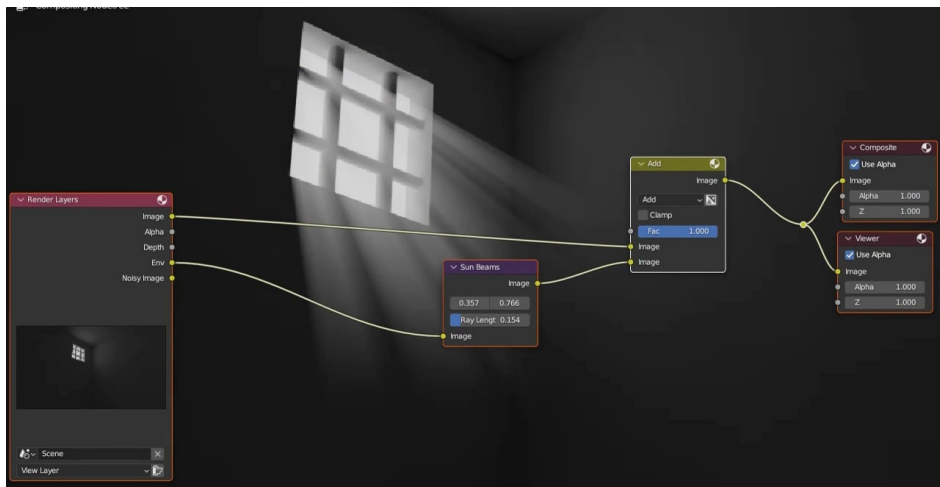
This is not suitable for our scene because we want to **limit the origin of the rays** to the virtual background visible through the **opening in the wall**.

Since that opening frames the background of the virtual universe, we can retrieve its **mask** by activating the **Environment Render Pass** in the **View Layers** tab of the Properties editor.



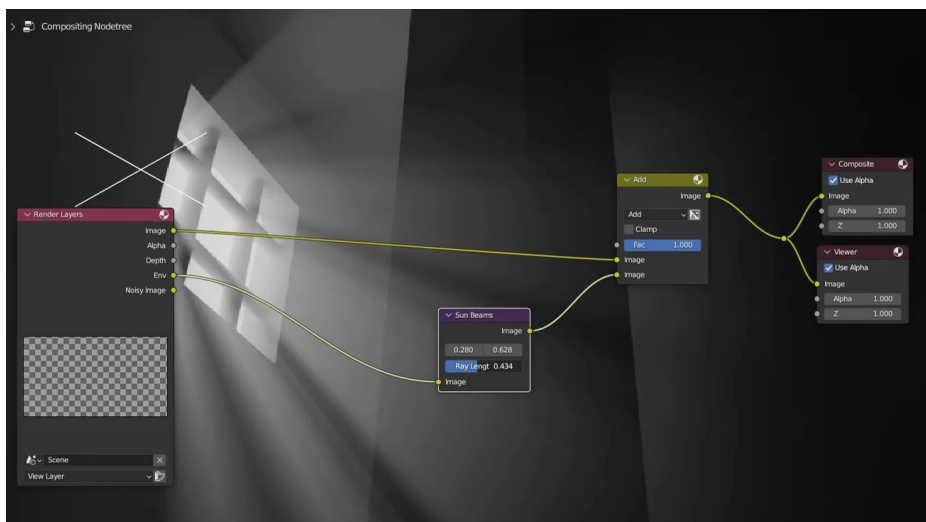
The inconvenience is that since this information has only been activated now, to visualize it and see the final effect obtained, **we will have to render again...** so, remember to activate it in time when you need to apply the effect to your scenes!

The Environment information can then be used as the source of the Sun Beams effect. The difference with using Render Image is evident.



Now, the **intensity** of the effect can be modified using the **Factor** parameter, which, like other node parameters, can have values that vary over time by inserting Keyframes, so you can even create some animations.

Before closing this first part, however, I must tell you that all that glitters is not gold: **these light rays are indeed post-production effects** applied to the image, so they have **no knowledge of the objects** that there are in the scene and therefore cannot be blocked by **obstacles**.



Therefore, this effect works well only in certain circumstances, but in other cases, it is necessary to resort to volumetric effects or other tricks.

For this tutorial, we stop here; in the next one, we will see how to apply the effect to light passing through windows (i.e., through a semi-transparent material), how to soften its edges, exclude some areas of the rendering that we are not interested in, and further modify it with Color Balance and other effects.

See you soon!