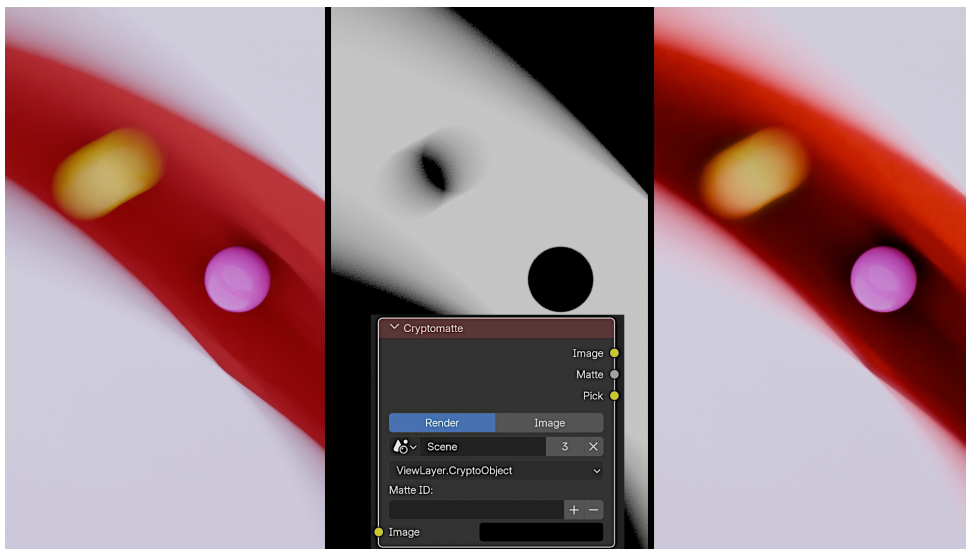


This PDF is available for free; feel free to share it!

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

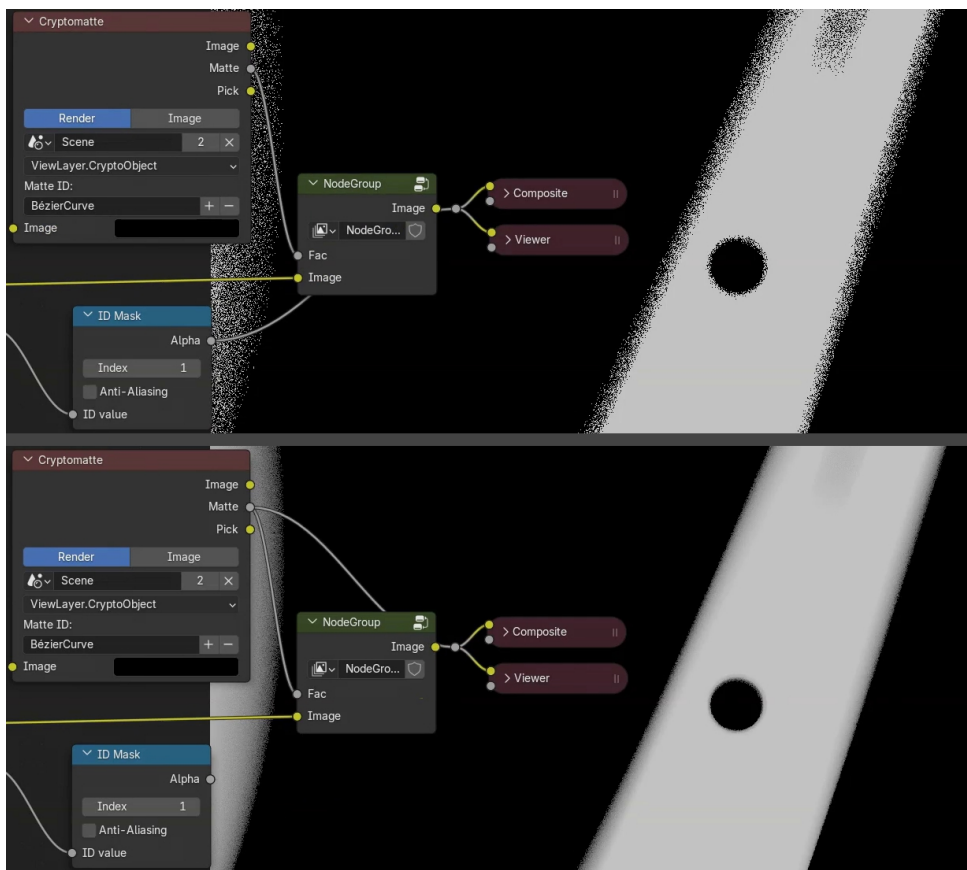


Hello everyone! This is a basic tutorial on how to use the Crypto Matte node in Blender 4.3 to extract object masks in Compositing, especially when your final render includes Motion Blur, like the example I'm showing on screen.

In most of my Blender Compositing tutorials, I usually rely on the ID Mask node, using information from Object Index or Material Index to create object masks and edit them separately in Compositing.

However, this method requires setting up Object Index or Material Index for the elements you want to isolate, which can become quite impractical if there are many elements involved.

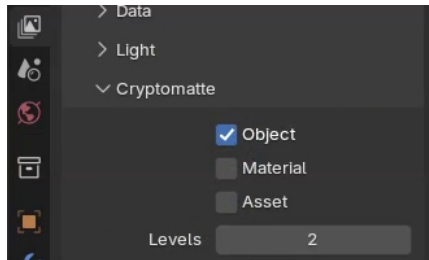
Furthermore, the mask generated by the ID Mask node is not very accurate when effects like Depth of Field or Motion Blur are present in your frames, as shown in this example. The quality of the mask produced with ID Mask is lower, which obviously affects the overall compositing quality. So, let's see how to use the Crypto Matte node instead.



As with many of my other video tutorials, the PDF version of this tutorial is available for free at the link provided in the description. Also, this is a great time to subscribe to the channel and turn on notifications so you won't miss upcoming video tutorials!

To use the Crypto Matte node in Compositing, the first thing we need to do is activate at least one of the three options in the Crypto Matte section of the **View Layer** tab in the **Properties Editor**.

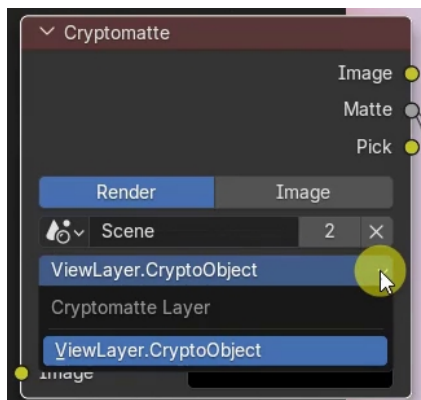
These options allow us to extract Crypto Matte masks for different objects, materials, or assets in Compositing. The Object and Material options have intuitive meanings. In my example, I'm using the Object option because I want to isolate the marble track, not sub-materials or objects that share the same material. The Assets option is used to isolate groups of objects that share the same Parent, but it's not needed in this example.



When you select at least one of these options, the **Levels** parameter becomes available. However, for this basic tutorial, I won't cover it and will leave it at its default value.

For this tutorial, I kept the **Levels** parameter at its default value, then selected **Object** in the Crypto Matte section. I also enabled **Object Index** further up in the tab to extract the Index Object information and show you the differences between the two methods. Once everything was set, I started the rendering process.

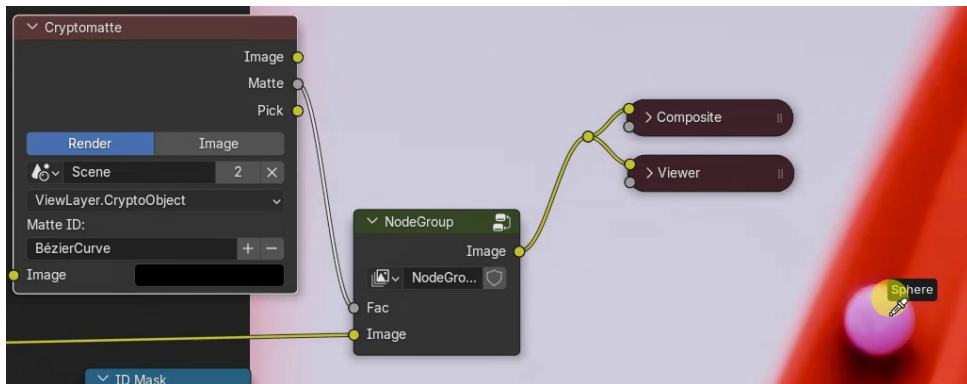
In the Compositor, I added a **Crypto Matte** node. As you can see, there is a **Layer** menu in this node, where I selected **Crypto Object** because that's the type of information I chose in the **View Layer** tab.



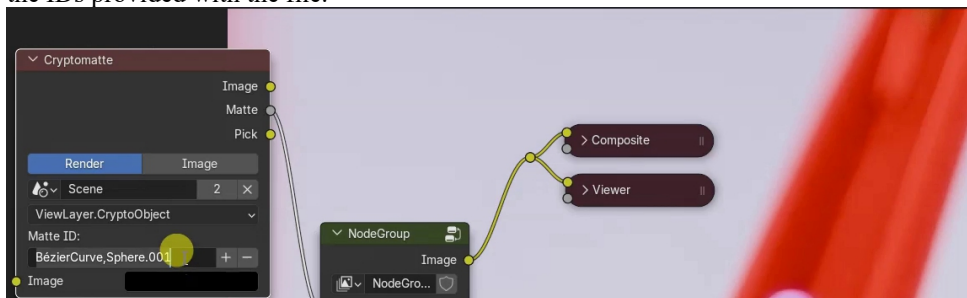
In the **Crypto Matte** node, you can choose to use information from the scene by selecting **Render** or retrieve data from an OpenEXR image by selecting **Image**. In my case, I'm obviously using the default **Render** option.

A unique feature of the **Crypto Matte** node is that the field where you specify the objects you want to mask is a list, allowing you to add or remove objects in two different ways.

The first method is to click the buttons next to the list to add or remove objects or materials by using the selector to pick them directly from the rendered image.



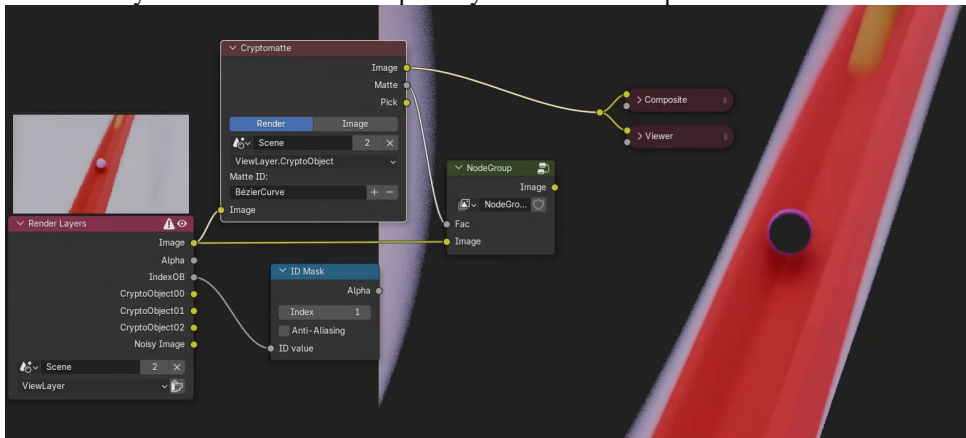
The second method is to type or delete the names of elements directly in the **Matte ID** field, separating them with commas, as I'm demonstrating with the object **Sphere 001**, which is somewhat difficult to select with the cursor due to significant Motion Blur. This method is also useful when your source is an EXR image because you can directly input the IDs provided with the file.



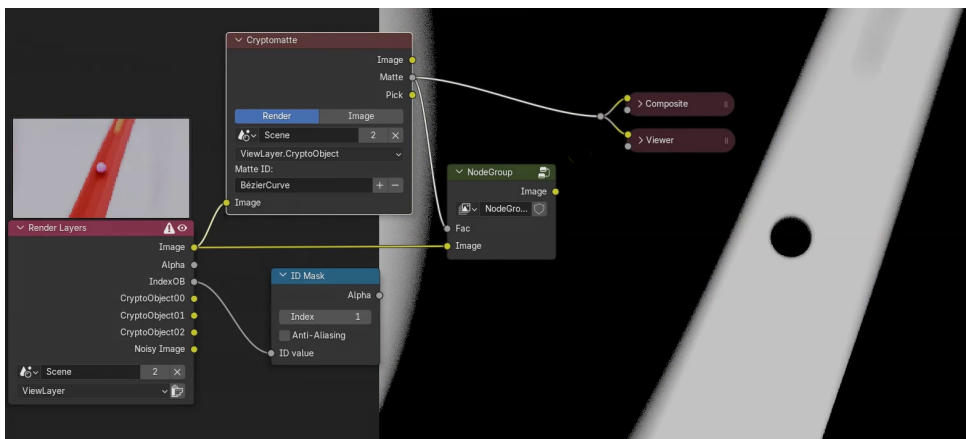
A major difference from using the **ID Mask** node is that you don't need to assign numeric identifiers to objects or materials before rendering. Instead, you can simply enable the **Crypto Matte Render Pass** before rendering and select the objects or materials you want afterward.

As you may have noticed, I didn't connect the **Crypto Object** outputs from the **Render Layers** node to the **Crypto Matte** node. However, it's still necessary to enable the **Crypto Matte Object Render Pass** in the **View Layer** tab; otherwise, the required information won't be available for the **Crypto Matte** node after rendering.

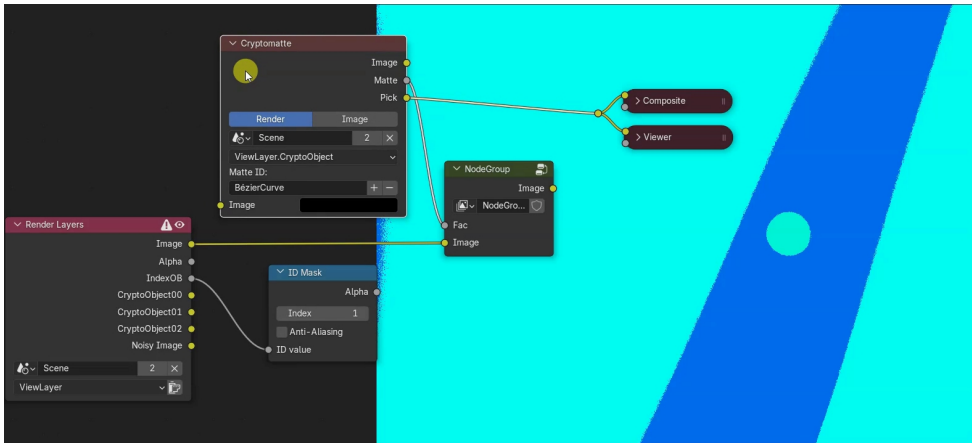
The **Crypto Matte** node has three output sockets. The first socket, called **Image**, returns a color image containing only the objects listed in the **Matte ID** field, provided that the original color image is connected to the **Image** input of the node. Essentially, it multiplies the render by the mask and sets transparency for all the black parts of the mask.



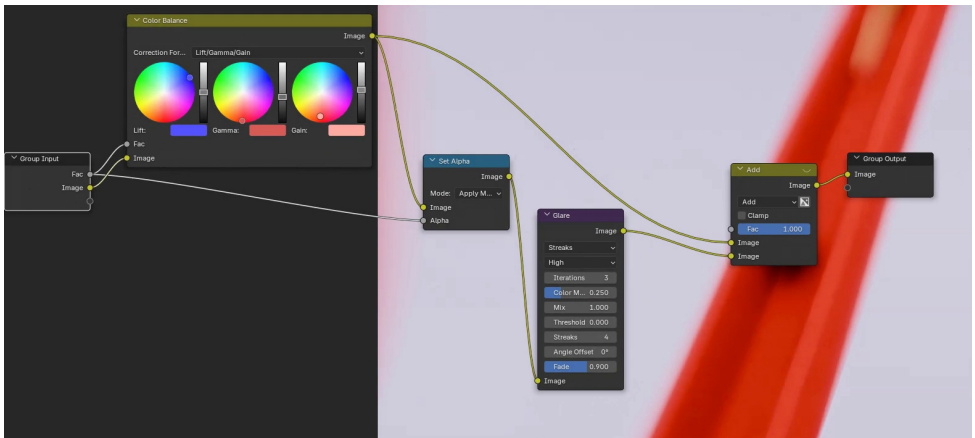
The second output socket, **Matte**, returns only the black and white mask without transparency. This is the socket I'm using for this example. It's not necessary to connect the **Image** input of the node because the information is taken from the scene and the **Matte ID** list within the node.



The third output socket, **Pick**, provides a mask with flat, distinct colors, which is useful for further selections or operations. Here too, the information comes directly from the node itself, so there's no need to connect the **Image** input to other nodes.



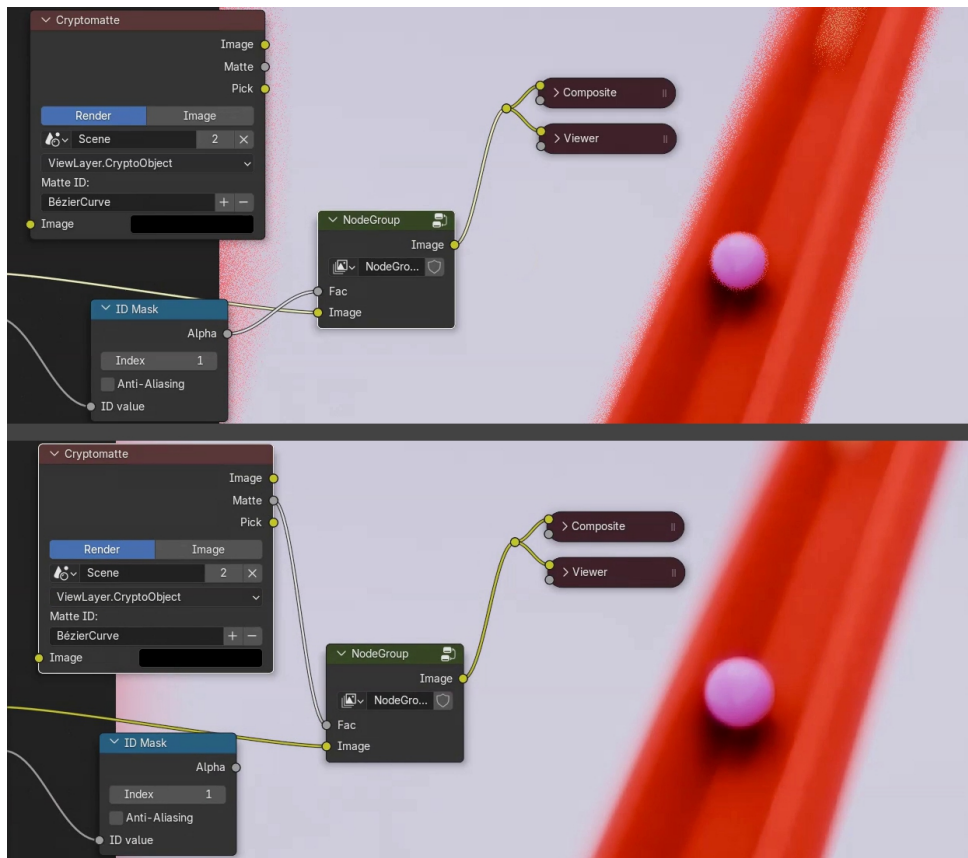
To modify the original render in the example I'm using, I took the **Matte** output from the **Crypto Matte** node and fed it into a group of nodes along with the original render to perform a simple color correction using the **Color Balance** node, limiting the corrections to the objects listed in the **Matte ID** field.



Using the **Set Alpha** node, I then isolated the colored image of the track and its Motion Blur on a transparent background, allowing me to process it separately.

With the **Mix** parameter of the **Glare** node set to **1**, I generated **Streaks** only for the track and its motion blur, then added this glow effect to the image obtained from the **Color Balance** node.

To compare the effects obtained with **Crypto Matte** and **ID Mask**, simply switch between the **ID Mask** and **Crypto Matte** masks as the **Factor** input in the compositing node group and observe the two results in the **Backdrop**.



Well, that's all for this basic tutorial on how to use the **Crypto Matte** node in Compositing! I hope you found it helpful!
See you next time!