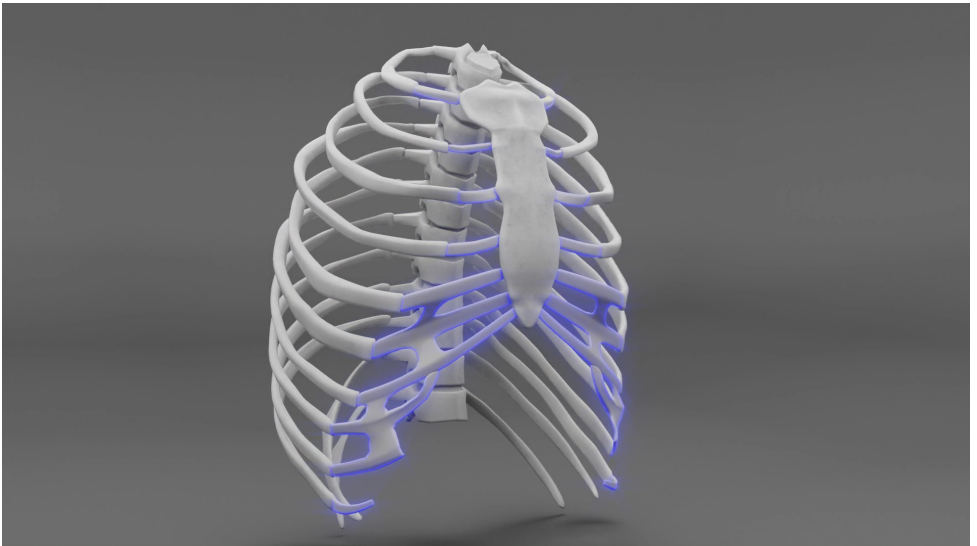


This PDF is available for free; feel free to share it!

Subscribe to my YT channel & turn on notifications to get updates on new videos!



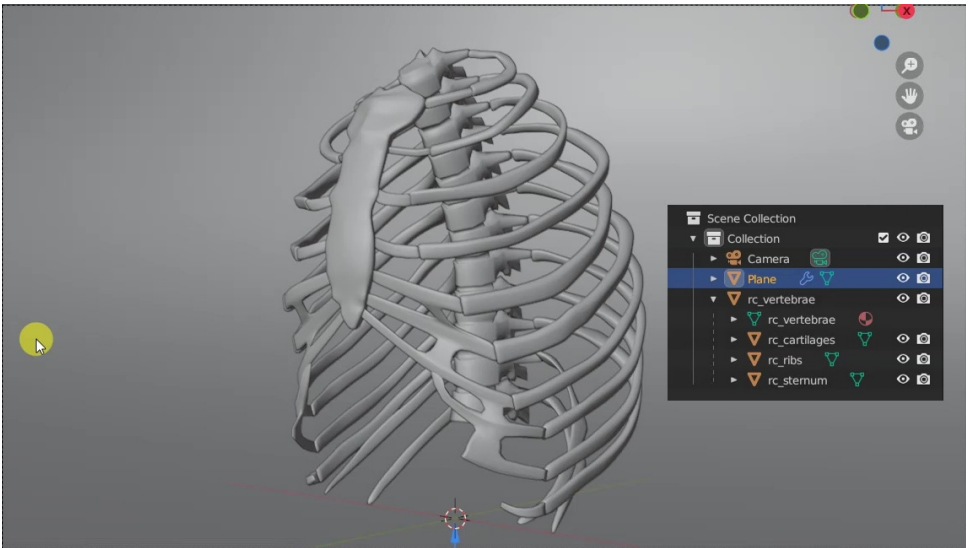
Hello everyone!

In this tutorial, we will see how to **emphasize the edges of an object** and add a **Glare** type glow through **post-production** of our renderings in Blender.

This tutorial was created using **Blender 3.6**, but the Compositing nodes used have been available since at least version 2.49, and it's likely they will remain available in future versions.

To emphasize the edges of an object placed within a Blender scene, we first need to **isolate that object from the render**. This can be achieved using **Cryptomatte** or, in my case, the **ID Mask** node, which will provide us with a white mask corresponding to the object we wish to isolate.

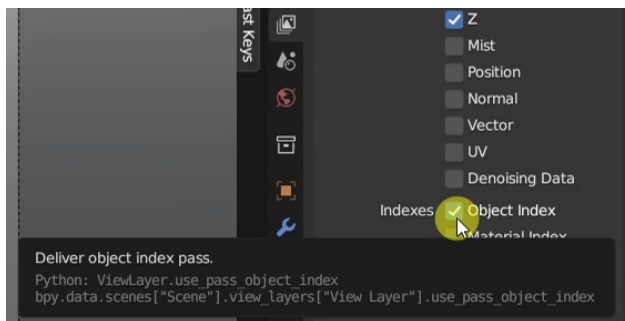
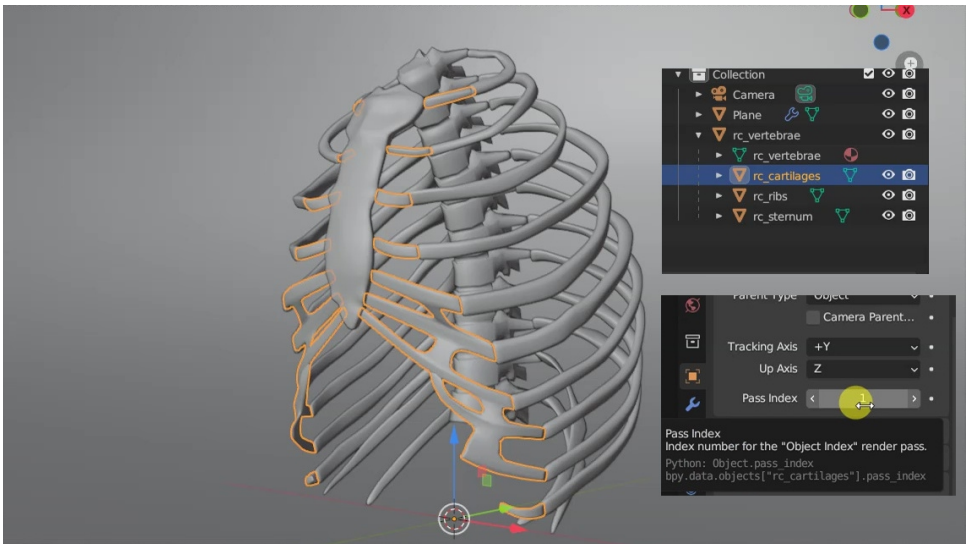
In the 3D model I'm using for this demonstration, there are four separate objects related to the ribcage: the vertebrae, cartilage, ribs, and sternum.



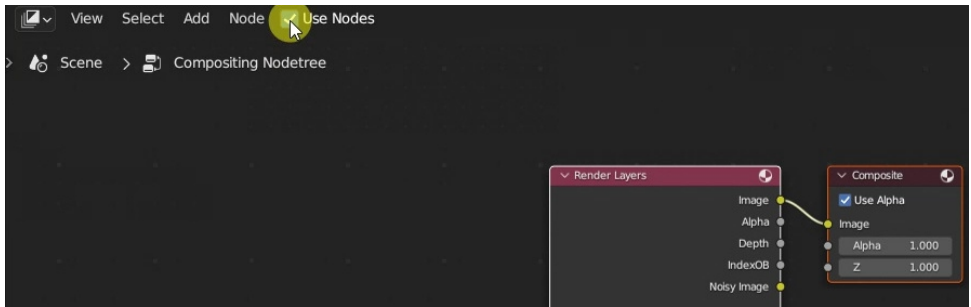
For our example, let's choose to highlight the edges of the **cartilage** parts.

Before starting the render, assign a unique numerical identifier to the object we want to isolate by specifying a value different from 0 in the **Pass Index field**, found within the **Relations** section of the **Object Properties** tab. I'm therefore setting this field to **1**.

To retrieve this information during post-production, we need to open the **View Layer** tab and check the **Object Index** box in the **Passes** section.



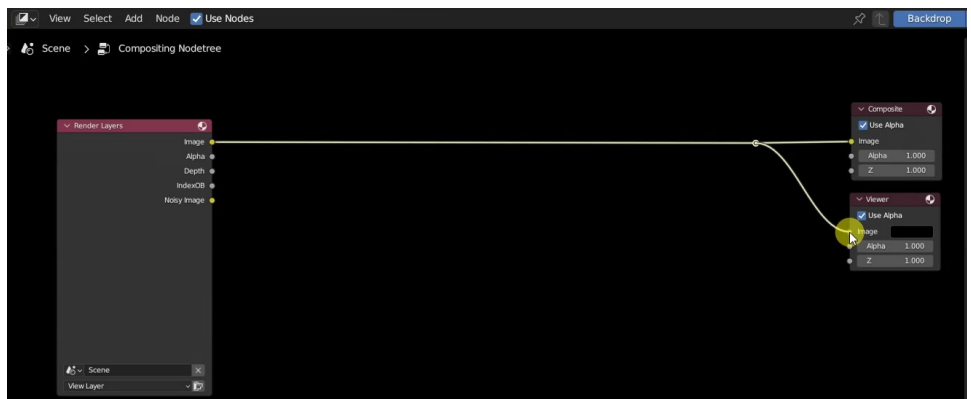
Now, we can open a **Compositor Editor** and activate node usage by selecting the **Use Nodes** checkbox.



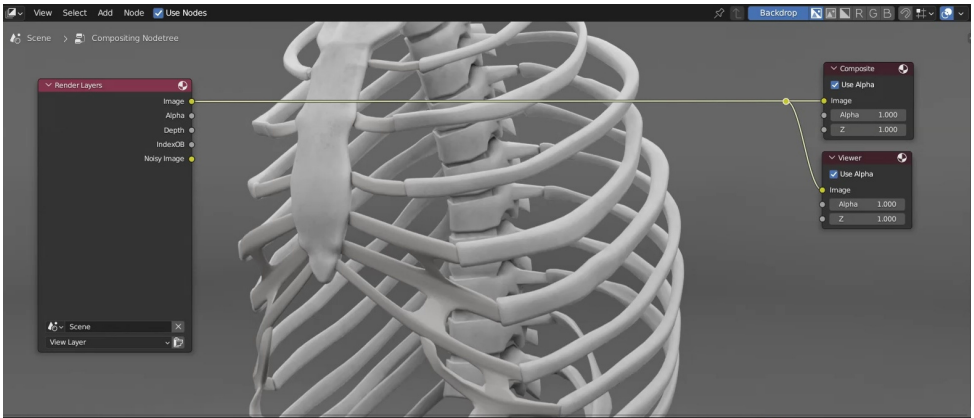
To preview the Compositing effects, also enable the **Backdrop** checkbox.

Then, insert a **Layout Reroute** node and an **Output Viewer** node.

This allows us to send the Compositing result to Reroute and from there, simultaneously to both **Composite** and **Viewer**.



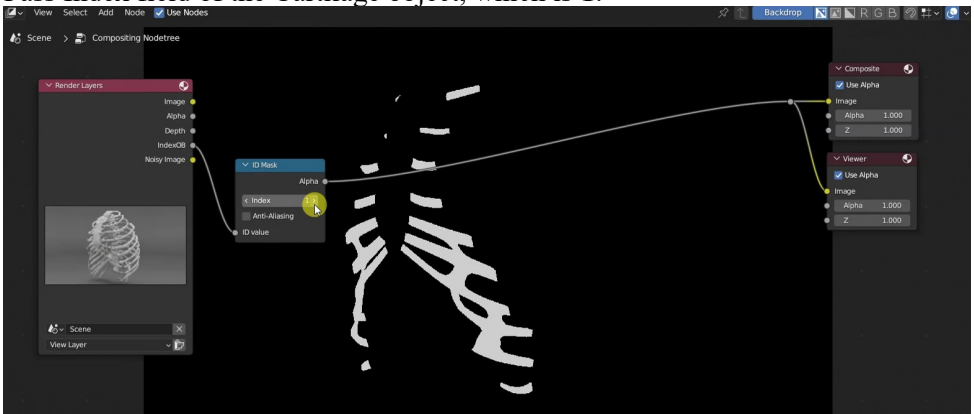
Now, we should perform an **initial render**, even with just a few samples, to obtain the first **Compositing data**. Among these, we will especially have the **mask** of the object, which can be retrieved from the **IndexOB** output of the **Render Layers** node.



If the render preview appears too large, you can shrink it by pressing the **V** key; if it's too small, you can enlarge it using the **ALT + V** key combination.

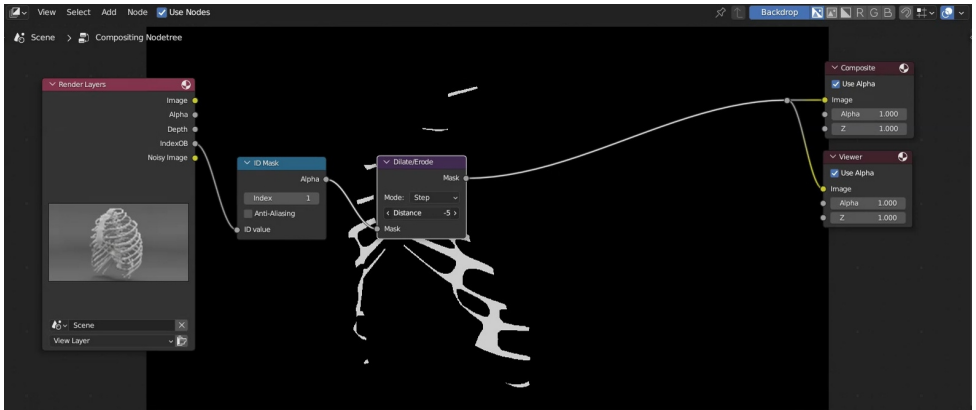
Now, let's examine the mask that isolates only the Cartilage object as follows: connect a **Convertor ID Mask** node between the **Render Layer** and **Reroute** nodes, but take the **Index OB** output instead of the Image output.

In the **Index** field of the **ID Mask** node, specify the **value** associated with the Pass Index field of the Cartilage object, which is **1**.



As you can see, we have a **white mask** corresponding to the Cartilage object. If we wanted to highlight or modify just those parts in some way, this mask would be an excellent starting point. However, we want to emphasize **only the edges** of this mask, so we need to find a way to isolate them.

In this case, the **Dilate/Erode Filter node** comes to our rescue: it allows us to **enlarge or shrink a mask**. You can see its effects by placing it between the **ID Mask** and **Reroute** nodes and then adjusting the **Distance** field value.



Specifically, by specifying **negative values**, we'll get a **shrunk** version of the original **mask**.

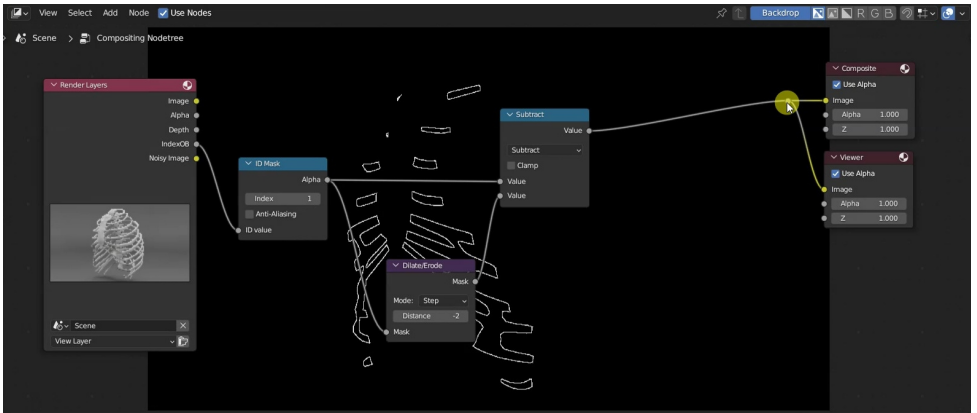
This suggests a solution to our problem, because **we can subtract the shrunk mask from the original one**, thus obtaining only the edges!

The node we'll use is the **Convertor Math** set to **Subtract** mode because this node works on numerical values, which are coming from the Alpha outputs of ID Mask and the Mask output of Dilate/Erode, as you can see by examining the color of these input/output sockets.

So, connect the Alpha output of the ID Mask to both the first Value input of the Math Subtract node and the Mask input of the Dilate/Erode node.

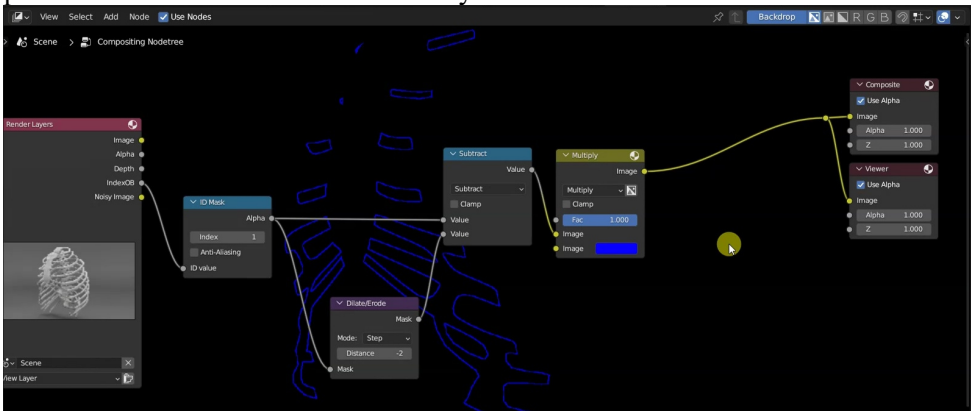
In the **Dilate/Erode node**, set a **negative value** for the **Distance** field, such as **-1 or -2**.

Next, connect the Mask output to the second Value input of the Math Subtract node. Then connect the output of Math Subtract to the Reroute node to preview the result we've achieved so far.

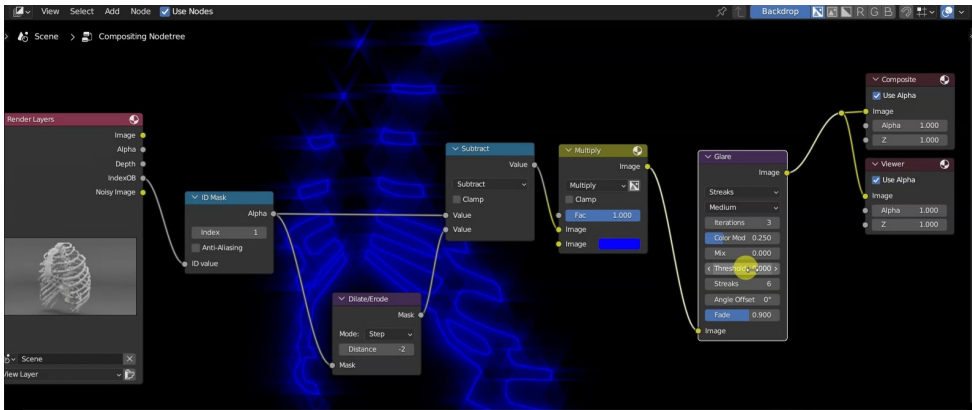


We can already **perform some operations on this mask**, such as applying a colored **Glare** filter.

First, **let's color the mask** by simply **multiplying** the output of the Math Subtract node with blue using a **Color Mix node**, in which we'll set the mode to **Multiply**. The result of multiplying by white is indeed the color by which we multiply, while multiplying by black always yields black as the result since it's equivalent to multiplying by 0. Since our starting image is white only at the edges of the cartilage object and black elsewhere, multiplying it by blue will give us blue in place of the white areas and black everywhere else.



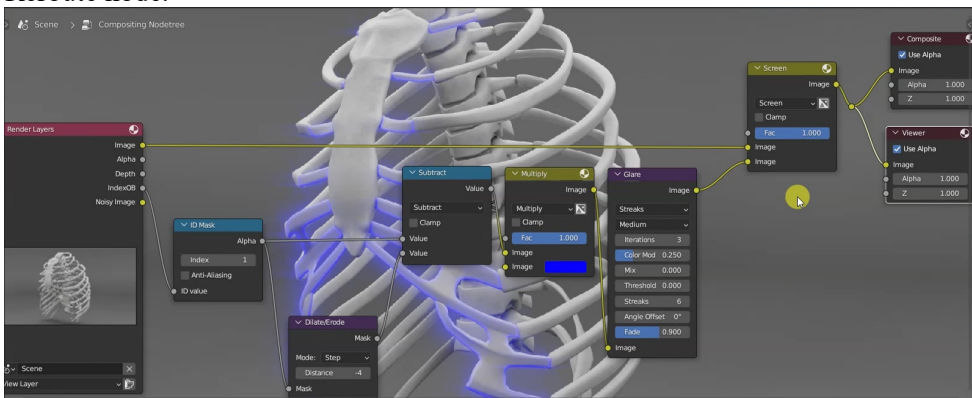
Next, we can connect a **Filter Glare** node to the output of the Color Mix Multiply node. In my case, I'm using the **Streaks mode with 6 light streaks**. Note that I'm setting the **Threshold value to 0**; otherwise, the node will have no effect.



If the effect seems too subtle for you, **try decreasing the Distance** value in the Dilate/Erode node. This adjustment can also be made later on when the effect is added to the original render.

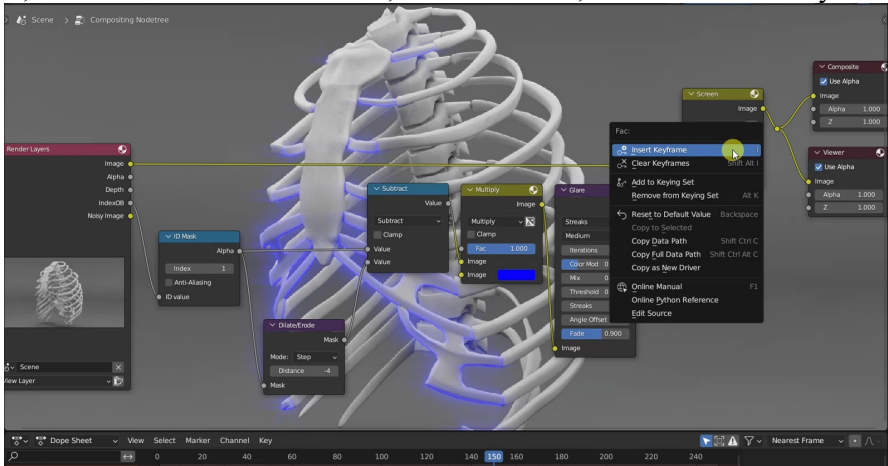
All that remains is to **overlay this result onto the original render**.

There are various ways to do this, but perhaps the simplest is to insert a **Color Mix** node set to **Screen** mode, with a value of 1 for the **Factor** field, before the Reroute node.

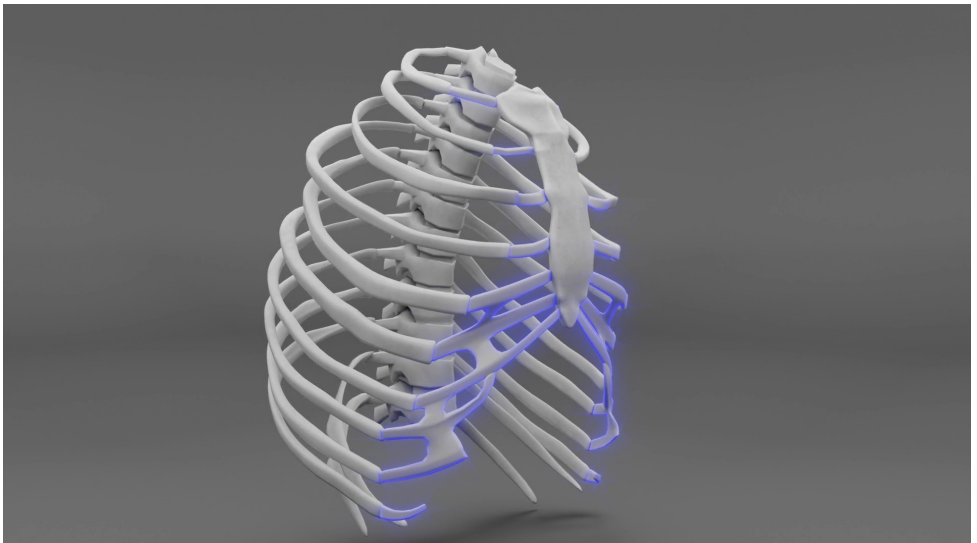


Be careful about the order of the connections on the Screen node: **the image** coming from Render Layers should be connected to the **first Image input**, while the one coming from the **Glow** node or any other mask effect nodes should be connected to the **second Image input**.

If the **Screen node's Factor value is set to 0**, the output will only show the original render. To achieve the **animation** I showed at the beginning of the video, where the **glow** isn't visible at the start but **increases** over time, simply set the Factor to 0 in the first frame and insert an **animation keyframe** for that field. Then, move forward in the Timeline, set **Factor** to 1, and insert a new keyframe.



So here's the final result, with the edges of the object we selected clearly highlighted, colored, and with a subtle glow.



I hope you found this tutorial helpful! See you soon!