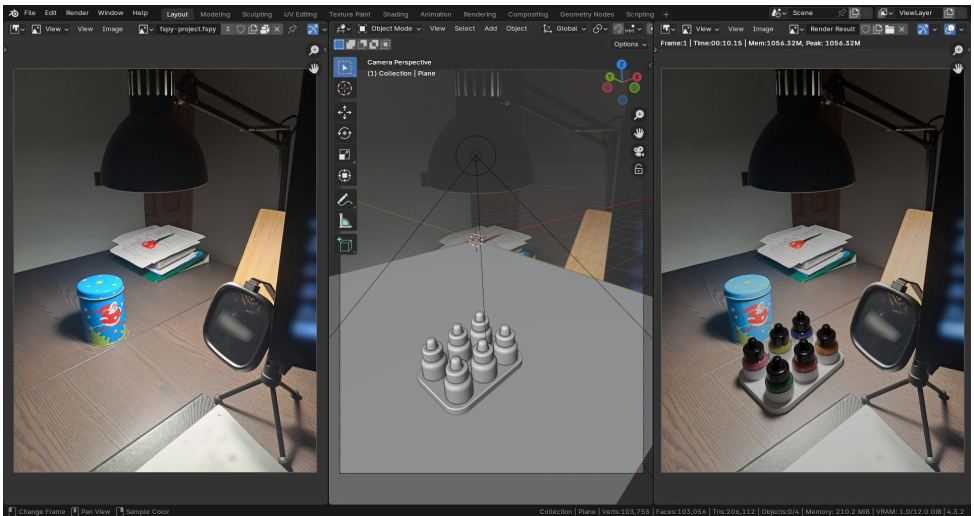


Photo Matching in Blender, Part 2: Shadow Catcher | Blender 4.x Tutorial
www.francescomilanese.com --- 2025 --- [GUMROAD \(PDF\)](#)
video version: <https://youtu.be/FL0t1F1QBWQ> (Scheduled for March 10, 2025)

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 second of three tutorials about a small photo compositing project in Blender.

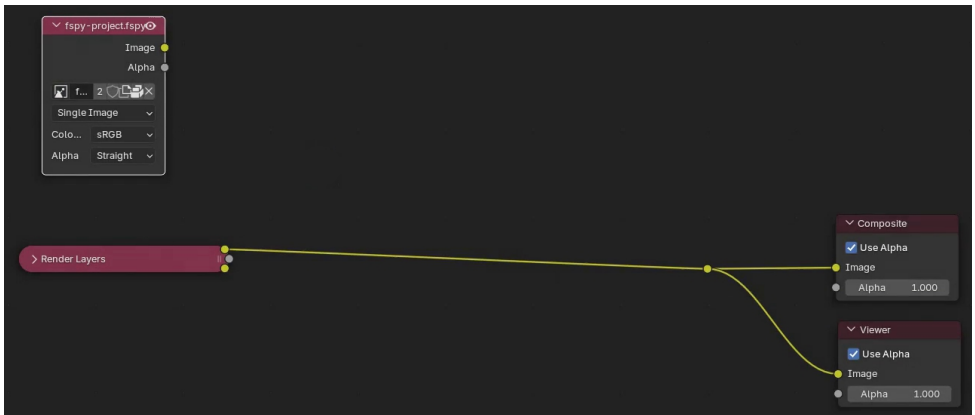
In the previous tutorial, we saw how to recreate the original photo's framing and how to position objects using the free fSpy add-on for Blender.

In this tutorial, we'll see how to use the Shadow Catcher mode to capture the shadows cast by the 3D model onto the Plane and add them to the final render in Compositing.

As with many of my other video tutorials, the PDF version of this tutorial can be downloaded for free from the link in the description. This is also a great time to subscribe to the channel and enable notifications so you won't miss the upcoming video tutorials!

First, in the Compositing editor, I insert an Input Image node with the original photo.

Photo Matching in Blender, Part 2: Shadow Catcher | Blender 4.x Tutorial
www.francescomilanese.com --- 2025 --- [GUMROAD \(PDF\)](#)
video version: <https://youtu.be/FL0t1F1QBWQ> (Scheduled for March 10, 2025)



At this point, the background image of the virtual camera is visible only within the virtual camera itself, not in the final render. So, if we start a render now, we would see the World background instead of the photo.

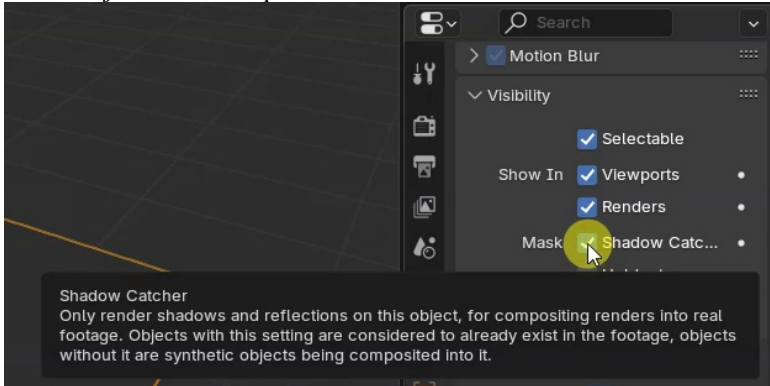
I also add an Alpha Over node, connecting the original photo to the first Image input of Alpha Over and the render coming from Render Layers to the second Image input. I then connect the Alpha Over output to both Composite and Viewer.

Even in this case, though, the final render will show the World background. Removing that background is one of the two necessary steps to use Shadow Catcher.



To render only the object to be inserted into the photo while keeping its shadows cast on the Plane, we need to perform two operations.

The first step is to select the Plane and any other objects you want to hide, and enable the *Shadow Catcher* option for each of these objects in the *Visibility* section of the *Object Data Properties* tab.



The second step is to activate the *Transparent* mode in the *Film* section of the *Render* tab. This mode will hide both the virtual universe background and the objects marked as *Shadow Catcher* in the final render.

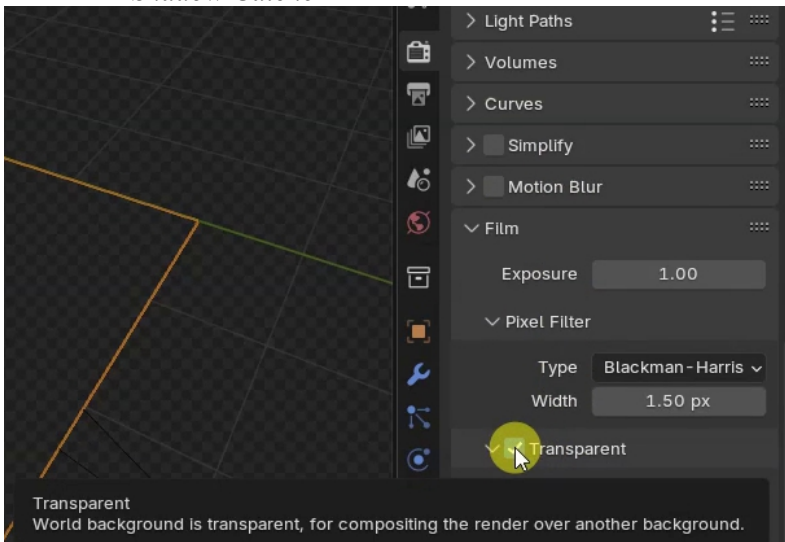


Photo Matching in Blender, Part 2: Shadow Catcher | Blender 4.x Tutorial
www.francescomilanese.com --- 2025 --- [GUMROAD \(PDF\)](#)
video version: <https://youtu.be/FL0t1F1QBWQ> (Scheduled for March 10, 2025)

If we start a render now, with the nodes set up as I showed you earlier, we'll get a basic photo compositing of the 3D model onto the original photo.



I'd like to point out something in *Rendered* mode within the 3D View: enabling *Shadow Catcher* is different from simply hiding objects because with *Shadow Catcher*, indirect lighting is still preserved.

At the end of the previous tutorial, I occasionally hid the Plane to assess the positioning of the objects. In that case, the visible objects appeared darker because the Plane's bounce lighting was no longer present.

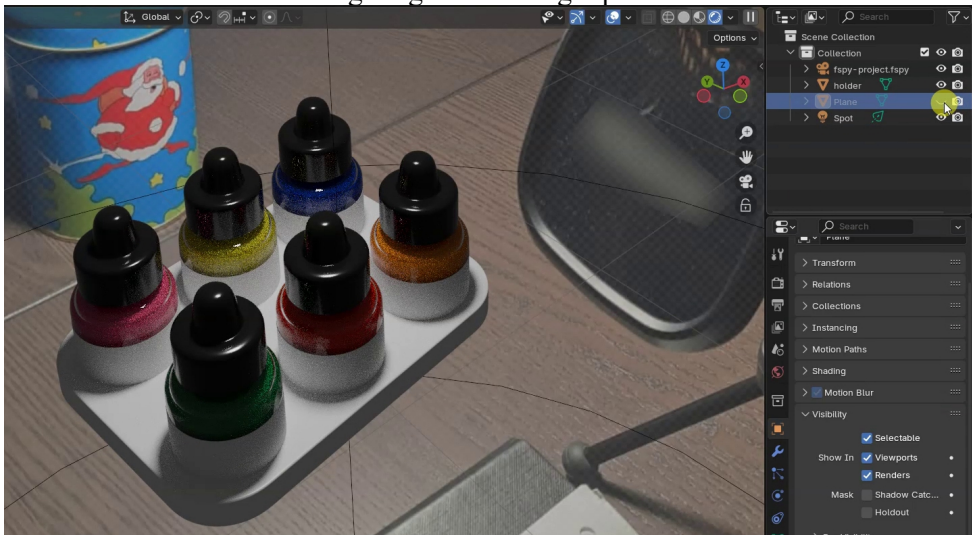


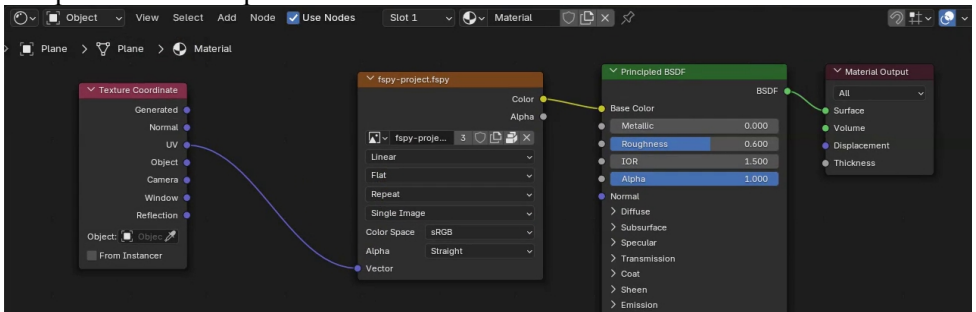
Photo Matching in Blender, Part 2: Shadow Catcher | Blender 4.x Tutorial
www.francescomilanesi.com --- 2025 --- [GUMROAD \(PDF\)](#)
video version: <https://youtu.be/FL0t1F1QBWQ> (Scheduled for March 10, 2025)

With *Shadow Catcher* and *Transparent* enabled, however, bounce lighting is maintained. To add more realism to your photo compositing, you can use colored objects or ones with *Textures*, as their colors will influence the appearance of the 3D models.



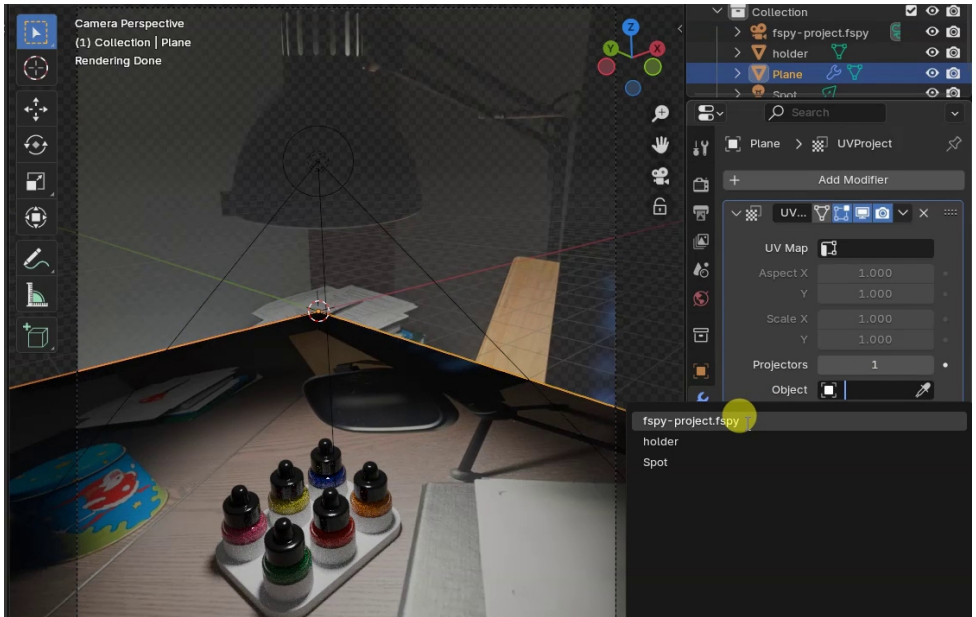
To show you a practical example, I'll temporarily disable *Shadow Catcher* so we can observe the appearance of the Plane in *Rendered* mode.

I'll then apply a *Principled* material to the Plane, setting the *Roughness* to a uniform value of 0.6 and using the original photo as a *Texture* for the *Base Color* channel. This isn't a proper PBR workflow since I should have provided a *Base Color* texture without lighting effects for the Plane. I'm proceeding this way only to highlight the differences in the indirect lighting on the 3D model being composited into the photo.



To perform UV mapping of the original photo onto the Plane, I'm using the *UV Project* modifier, which I covered in a previous video tutorial on my channel.

First, I set the *fSpy camera* as the *Object* in the modifier's field. At this point, to improve the mapping quality, I need to perform two operations.



The first step is to increase the number of subdivisions on the Plane. Pay attention to the lines of the objects to determine a suitable number of subdivisions. In my case, I'm using the *Loop Cut and Slide* tool to create the cuts.

The second step is to set the *Aspect Ratio* in the modifier to match that of the original image. By opening the *Image* tab of the original photo in an *Image Editor*, I see that the resolution of the original image is 1200 by 1600 pixels.

Therefore, I set 1.2 and 1.6 in the *Aspect X* and *Aspect Y* fields of the *UV Project* modifier, respectively.

Photo Matching in Blender, Part 2: Shadow Catcher | Blender 4.x Tutorial
www.francescomilanese.com --- 2025 --- [GUMROAD \(PDF\)](#)
video version: <https://youtu.be/FL0t1F1QBWQ> (Scheduled for March 10, 2025)

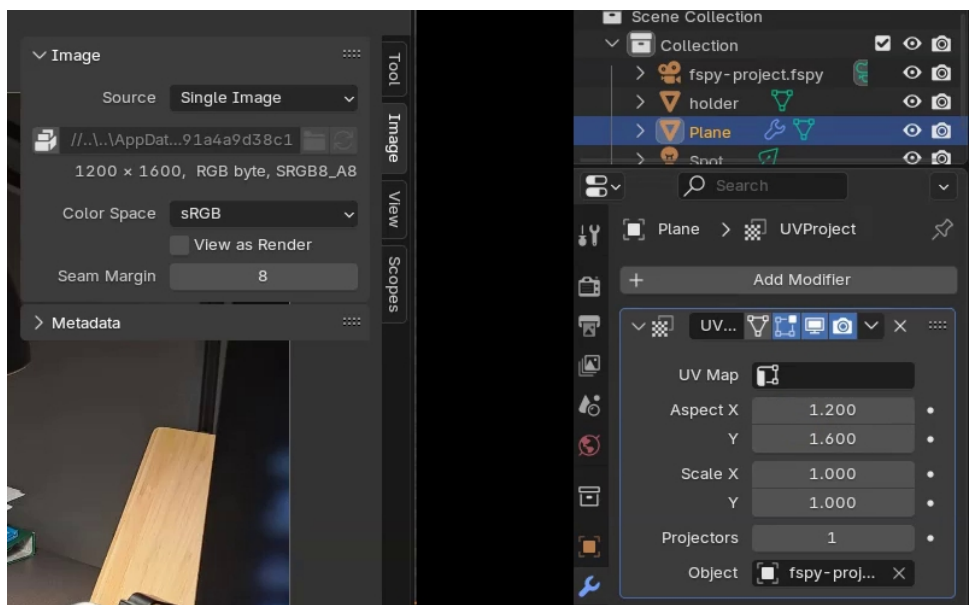
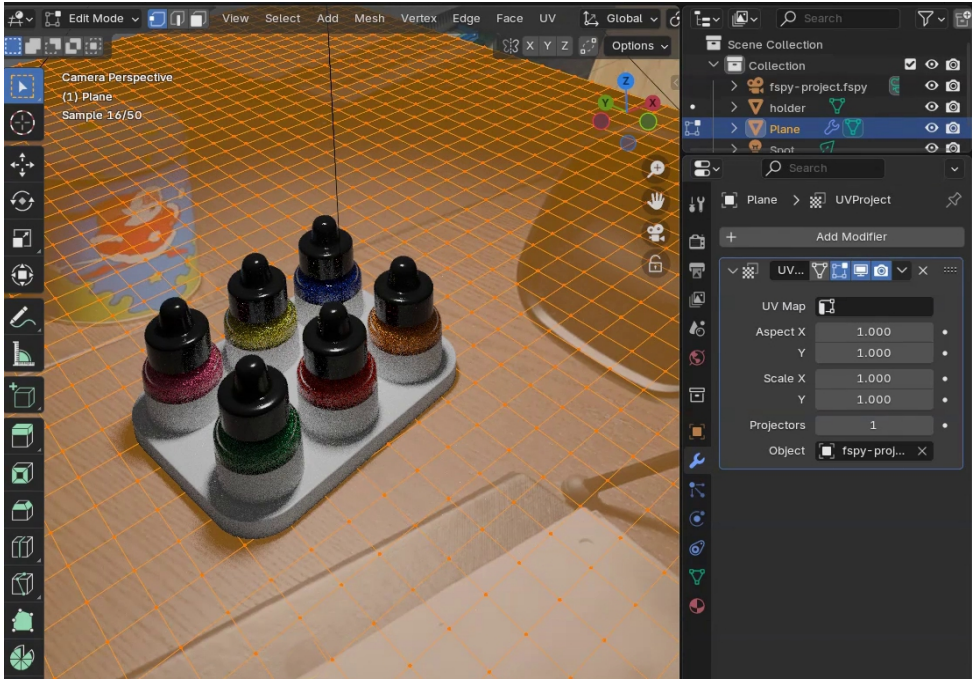
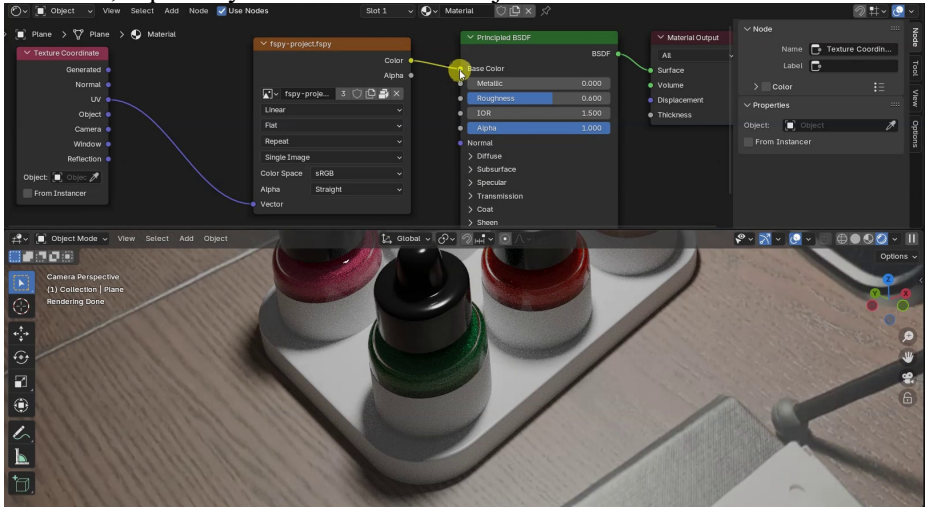


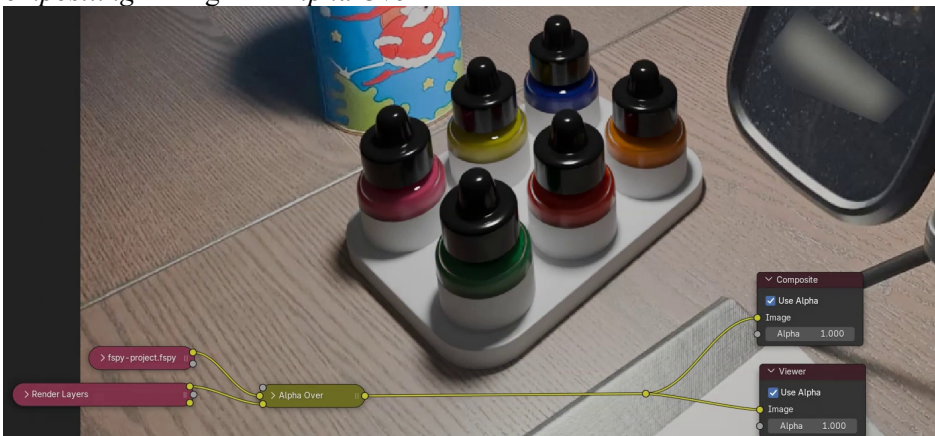
Photo Matching in Blender, Part 2: Shadow Catcher | Blender 4.x Tutorial
www.francescomilanese.com --- 2025 --- [GUMROAD \(PDF\)](#)
video version: <https://youtu.be/FL0t1F1QBWQ> (Scheduled for March 10, 2025)

If we compare the indirect lighting with a material without a texture to the lighting obtained with the textured material, we immediately notice some differences, especially in the areas of the object closest to the table.

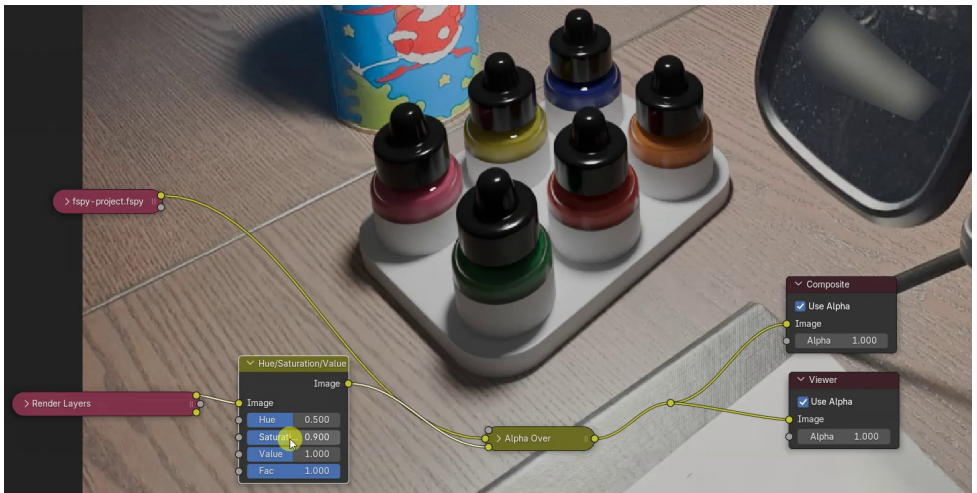


As I mentioned earlier, I should have used a physically accurate *Base Color* texture and let Blender calculate the bounce lighting with the *Spot Light* source. However, even with this example, you can observe the differences.

At this point, I can enable the *Shadow Catcher* option for the Plane again and start a render. This will account for the original image used as the background in *Compositing* through the *Alpha Over* node.



In the final image, we'll also have the shadows generated by the *Shadow Catcher*. Before wrapping up this second tutorial, I recommend adding a *Hue Saturation Value* node between the scene render and the *Alpha Over* node to slightly reduce the saturation of the render. This adjustment is particularly useful when working with objects that have very vibrant textures and colors, like the one I'm using in this project.



However, this small project isn't over yet because I haven't shown you how to process the *Shadow Catcher* information separately in *Compositing* or how to further modify the image of the 3D model, for example, by adding some distortion with *Lens Distortion*.

I'll cover these topics in the next tutorial! I hope you found this one helpful. See you soon!