artstation.com

Parallax Mapping in Blender - Part 1

Oliver Loftus

5-6 minutes

I decided to write this tutorial to help people get started with parallax mapping in Blender. I have come across some amazing examples of parallax mapping online, however, the node setups are often insanely complicated, or something that you need to pay to use.

While parallax mapping can certainly get complicated, it does not need to be. It is something you can easily do yourself using standard nodes. My goal here is to provide an entry point, so that you can start building your own parallax maps in Blender right away. How complicated you want to get afterwards is entirely up to you, but for the sake of clarity, I am going to keep this as rudimentary as I can.

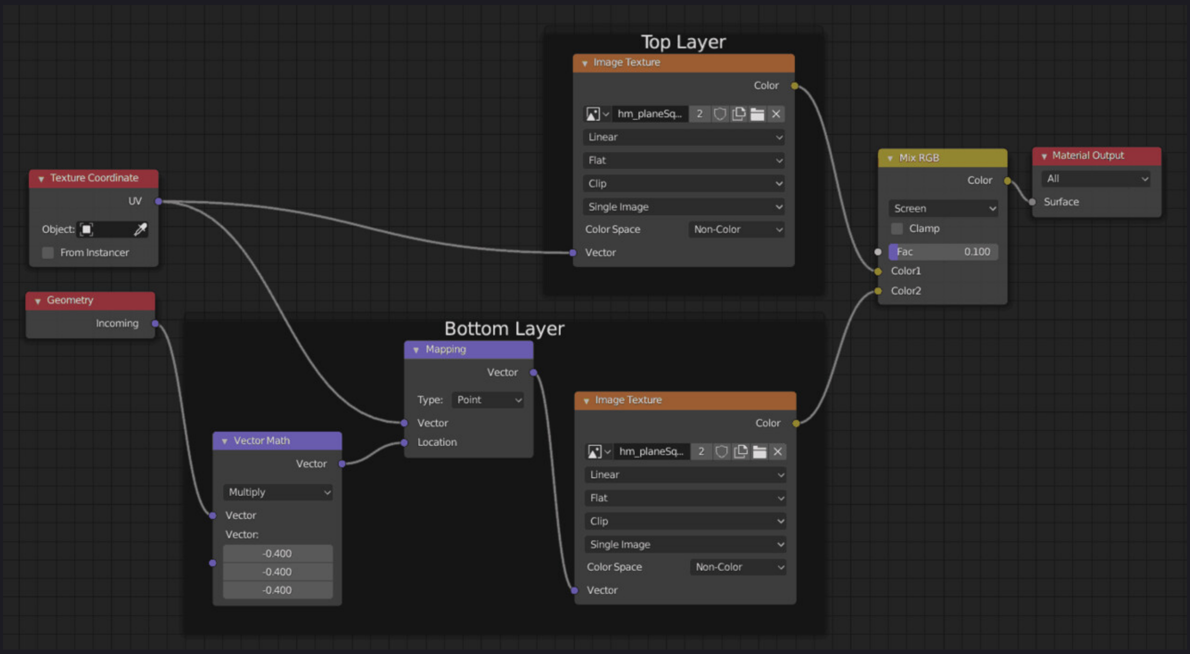


Image Texture: hm_planeSq... 2, Linear, Flat, Clip, Single Image, Color Space: Non-Color

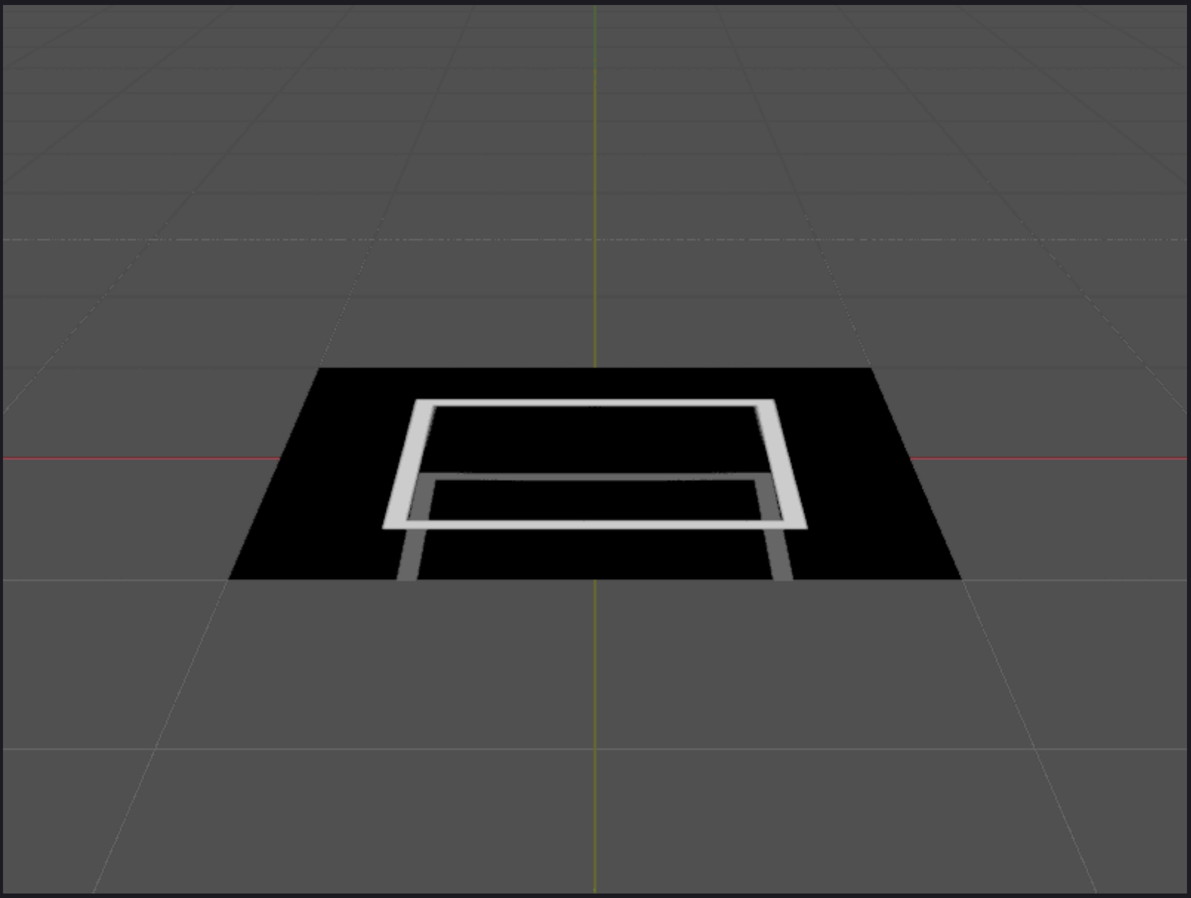
Mix RGB: Screen, Clamp □, Fac 0.100

Material Output: All

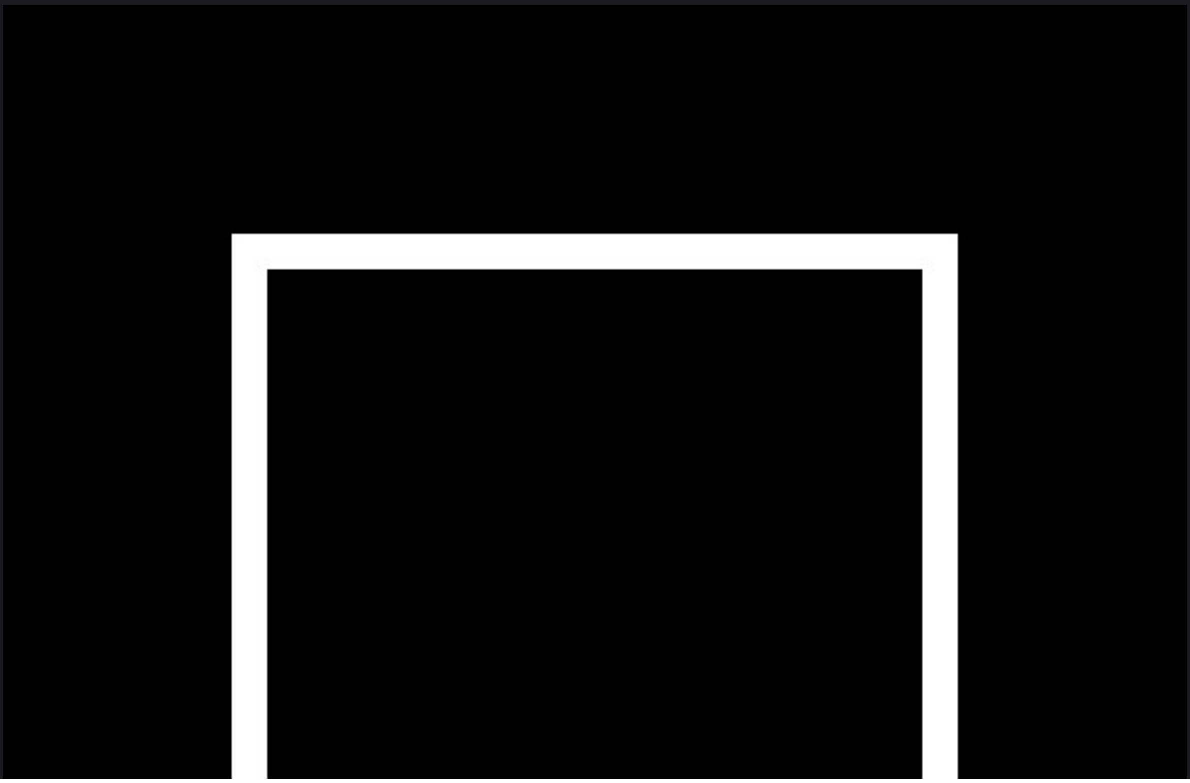
Mapping: Type: Point

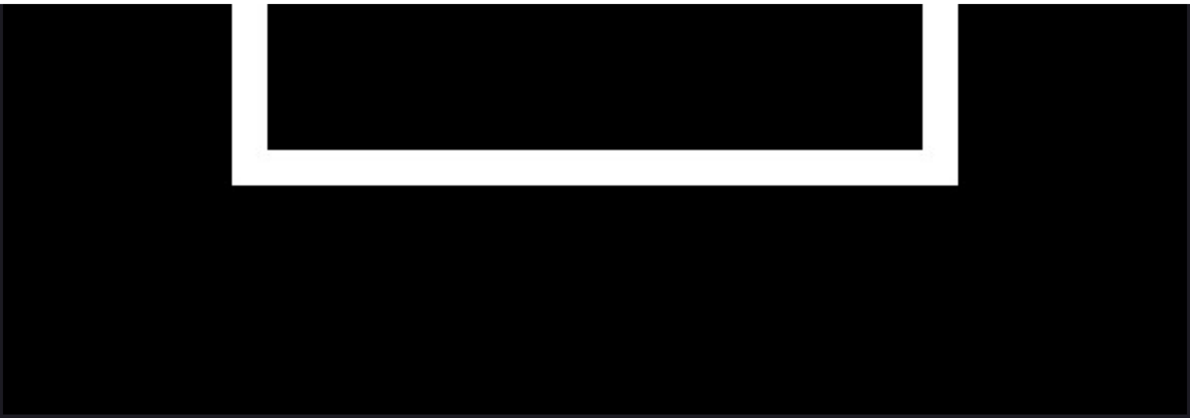
Vector Math: Multiply, Vector: -0.400, -0.400, -0.400

This is my node setup for, perhaps, the most basic parallax map possible. It will work on a standard **blender plain,** without any unwrapping required, and is fully compatible with Eevee. It consists of two layers. The top layer is just an ordinary image texture applied to the surface in an ordinary way. The bottom layer is the exact same image texture, however, this time we are intercepting its UV coordinates and adjusting their **location** based on the viewing angle (**incoming**). When the two images are mixed, one is displaced, which creates a 3D ‘drop shadow’ effect.

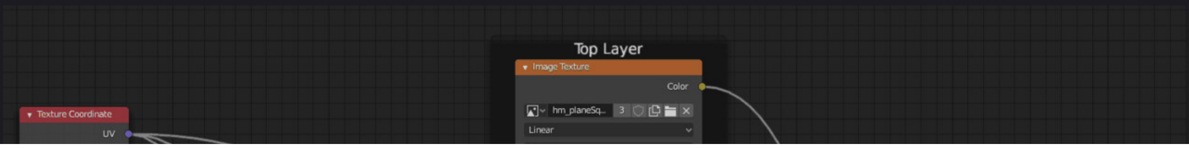


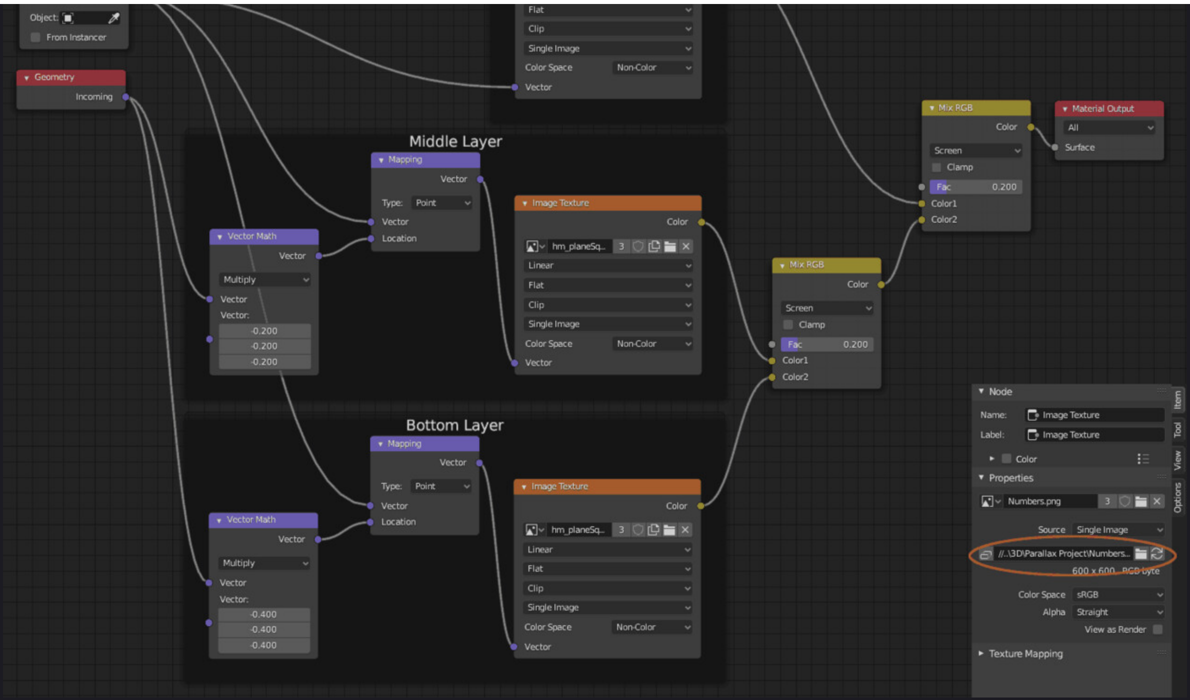
The apparent depth of the ‘shadow’ layer can be adjusted in the **vector math** node (in this example it is set to -0.400). Using a negative number here will cause the layer to appear below the level of the plain. A positive number will cause it to appear above the level of the plain. Generally speaking, parallax maps work better when all the details appear below the true surface. This is because details appearing to protrude beyond, or float above the true surface, risk being cropped when viewed at steep angles.



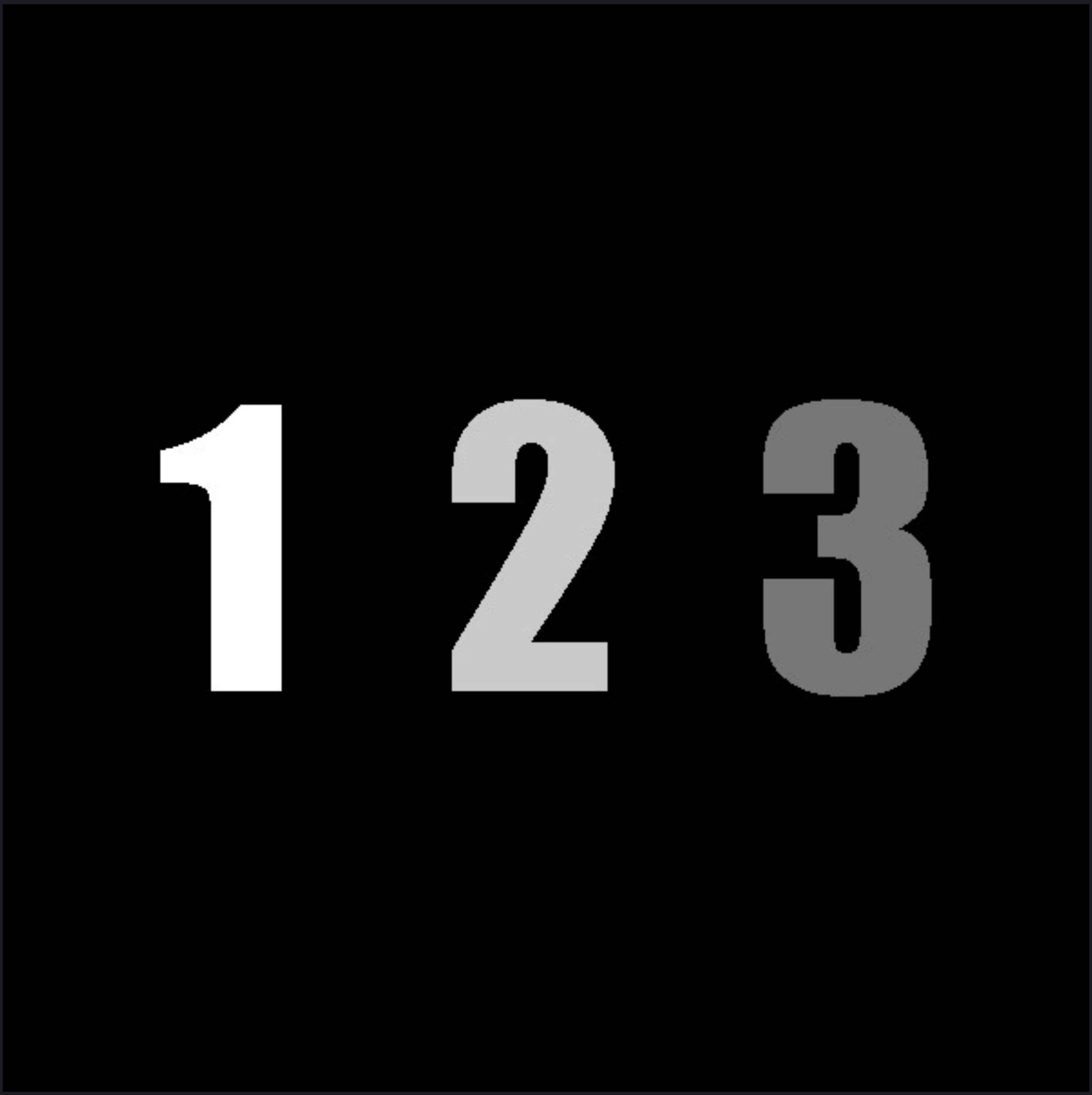


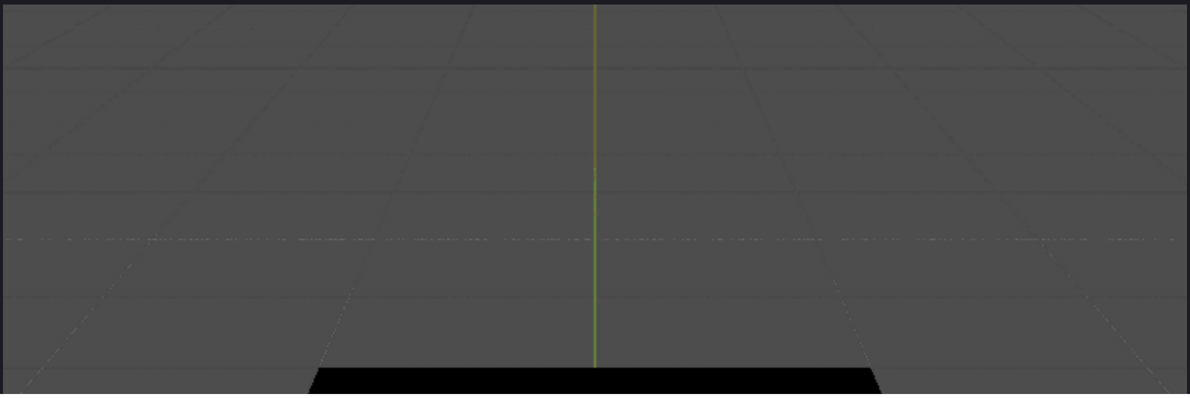
Eventually, I want this image-texture to function as a height map, so I am using greyscale images here. I am mixing them using the **screen** blending mode, so that lighter areas appear on top without any masking required. However, you can use different blend modes for different effects. You can even use a different image on each layer. For example, you could have the top layer be an iron grate, while the bottom layer is flowing water, then use a mask to combine them so it looks like the water is flowing a few feet below the grating.

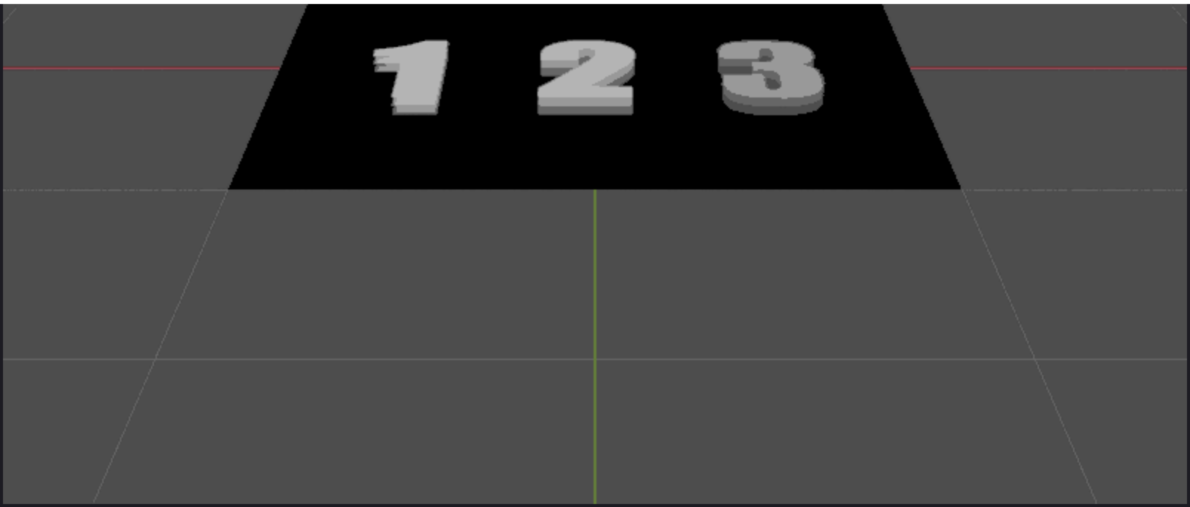




Up until now, our parallax material has only had two possible heights; if we want to be able to represent a range of heights then we need to ‘up’ the resolution by adding more layers. We will also need an image-texture with some different grey-values to represent different elevations. Adding an extra layer can be done by duplicating the bottom layer, then adjusting the ‘new layer’s’ height on the **vector math** node. The image textures can be changed individually, or you can replace all of them at once by selecting the texture node, opening the properties panel (N) and redirecting the source address to a new image.

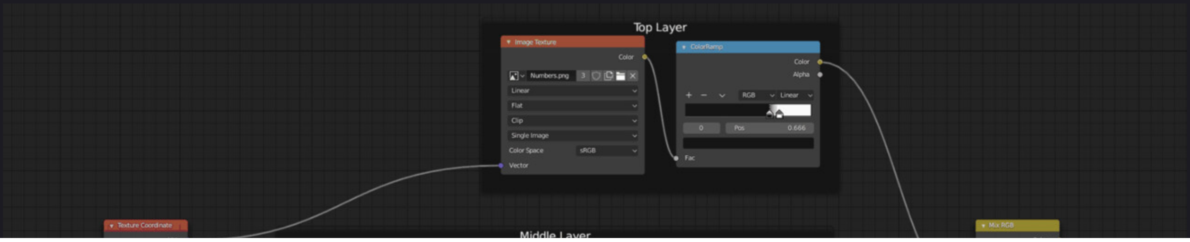


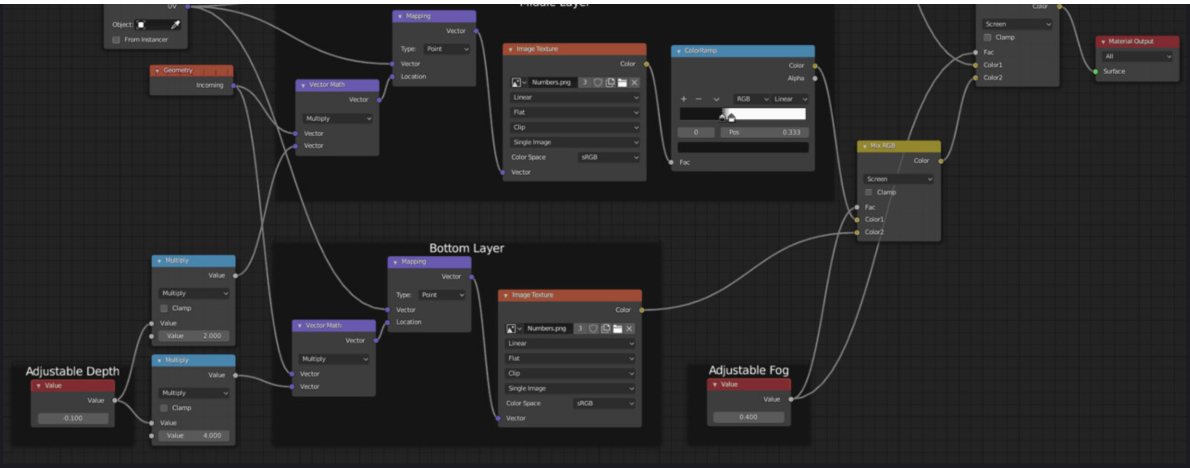




I have moved the layers closer together in the above example, to show how layers help to fill out shapes and give them volume. The more layers you use, the better the effect.

As you can see, the numbers in the image are still all the same height, despite being different values of grey. If we want the grey-values to affect height, then we need to filter darker values out of higher layers. An easy way to do this would be to add colour ramps to each layer, which can exclude darker tones.

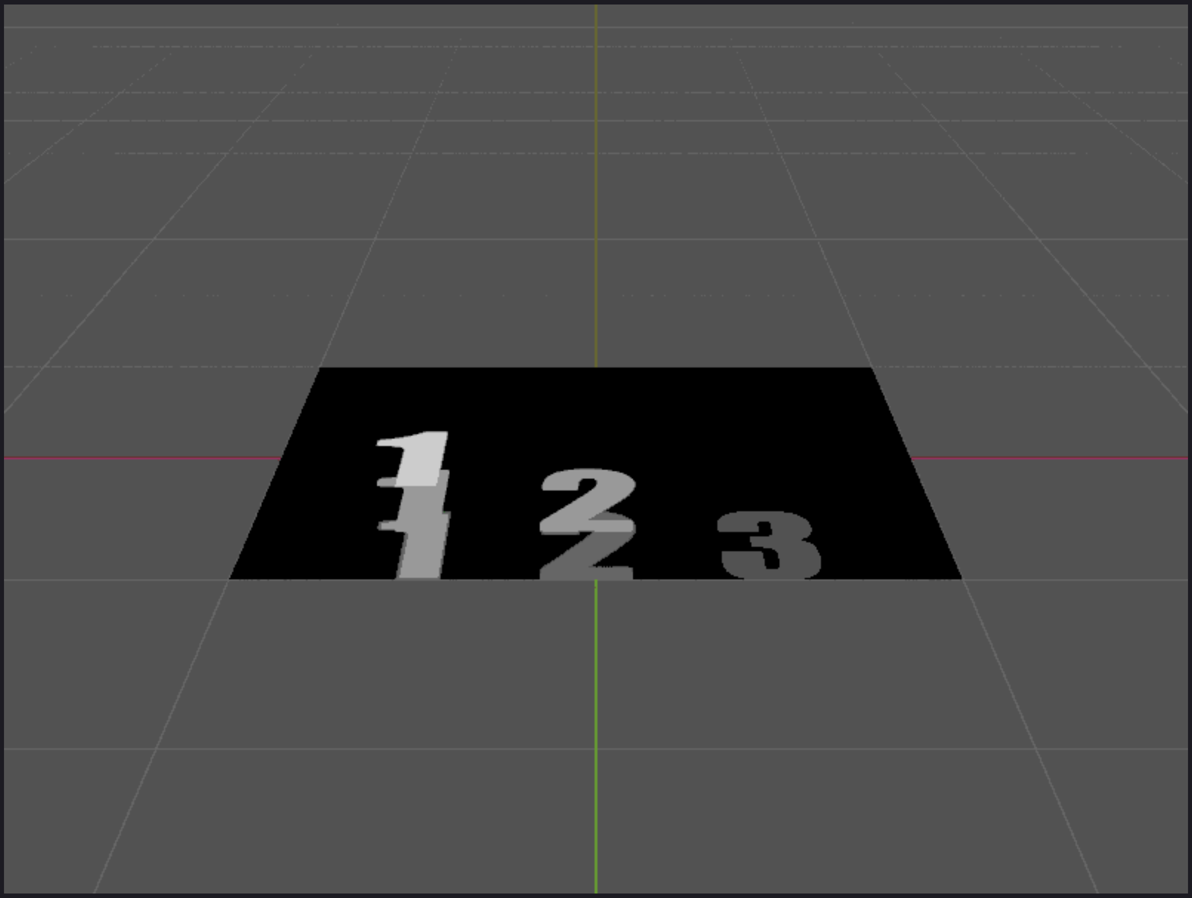




Here is the node setup with the colour ramps attached. Moving the black-point up will filter out all the values below. Since there are three layers in this example, I have excluded approximately two-thirds of the values from the top layer, one-third from the middle layer, and zero-thirds from the bottom layer (or no colour ramp).

Moving the white point down will effect the ‘minimum’ opacity of the remaining grey values. For hard edges, like text, it is better to tighten the values so that your layers are nice and defined. For softer (sloping) edges, it might be better to move the white point away, so that the layers blend together.

You will also notice that I have added some red **value** nodes, these are just there for convenience, so I can change the values on multiple nodes without having to edit each one individually.



Here is final result.

In Part 2 I would like to take this a step further: adding textures to parallax objects and getting them to react to scene lighting. I would also like to go over some tips for working with multiple layers and organising nodes, so that they don’t turn into node-spaghetti...