

# **Volumetric Spotlight/Fake Volumetric lighting**

By Joseph Y.

## Overview

The Volumetric Spotlight or fake volumetric lighting is a shader effect that mimics the lighting found in volumetric lighting. The reason to fake it is to save on computer resources, computational power, be able to run on lower end devices, or sometimes artistic direction. The light can have different colors and the ability to add in some noise or texture onto the lighting effect. It is designed to run on the Unity URP pipeline and Unity 6, but theoretically, it could run on Unity 2022 or older versions of unity but it is untested. This effect utilizes ShaderGraph to create the shader effect.

---

## Installation

Before you begin:

1. Ensure you use the Universal Render Pipeline (URP) and Unity 6 (6000.0.25f1).
2. Ensure you have installed the necessary packages, such as Shader Graph.

The effect is contained in a Unity package you can import into Unity. To do this, navigate to Assets > Import Packages, select the Unity package, and open it. A popup will appear showing the contents of the package. Choose to import everything from the package and click OK. They will be imported into the Unity project under the Fog folder.

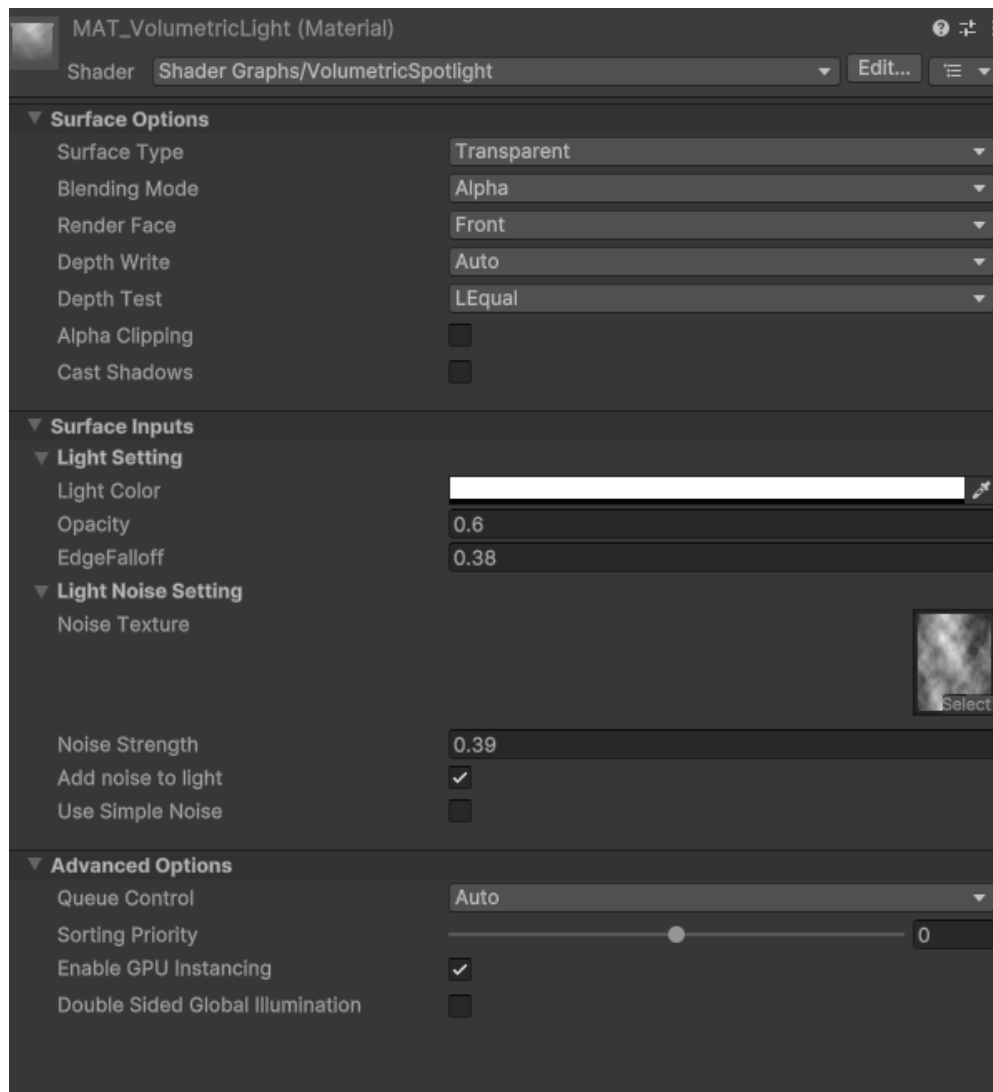
After importing, you should see several specific items, including a folder for prefabs (one with a cone shape and one with a rectangle shape), a cloud noise texture, a material for prefabs, a cone mesh for prefabs, and the shader graph itself.

To apply the effect:

1. Create a new material from the shadergraph, or use the provided one if you want to.
2. Drag the material onto the mesh object/gameobject that you want to apply the effect on, or go to the object's inspector panel and change its material to the material with the volumetric spotlight shader.
3. Optionally, you can use one of the prefabs to have a volumetric spotlight by going to the prefab folder and drag one of them into the scene.

You can adjust the effect by going to its material and changing its parameters.

## Using the effect



This effect will behave similarly to how you would apply shaders or materials to mesh objects. If you change anything within a material with the shader one, you will affect all meshes that have the same material. So if you want certain volumetric spotlights to be separate, you will need to ensure that it's not running on the same material. You can use the provided noise texture or a custom one (refer to [Creating and Using custom textures](#)) and you can use the provided 3D mesh for the volumetric spotlight or use a custom one (refer to [Creating and Using custom mesh](#)). Here are the details about the parameters:

1. **Light Color:** Color for the volumetric spotlight. The color's alpha has no effect on the shader effect.

2. **Opacity:** Affects the light's opacity. 1 means more opaque/brighter while 0 means more transparent/darker
3. **EdgeFalloff:** Affects the softening of the lighting. Higher values means that the lighting will be more harsher with hard edges while lower values will be more softer with more softer edges.
4. **Noise Texture:** The texture containing our noise to apply onto the light. Refer to [Creating and Using custom textures](#) for more info.
5. **Noise Strength:** Affects how strong you want the noise to be.
6. **Add noise to light:** Affects whether or not you want to add some noise to the light.
7. **Use Simple Noise:** Uses a simple noise texture instead of the noise texture input. Only visible if the option for Add noise to light is enabled.

The effect works best at dark scenes. At day time or areas with lots of lighting, the effect will be less pronounced and may require increasing the values a bit more. This will affect how it will look in the night too, so try to strive for a good balance to optimally ensure that it is bright enough for both day and night time use.

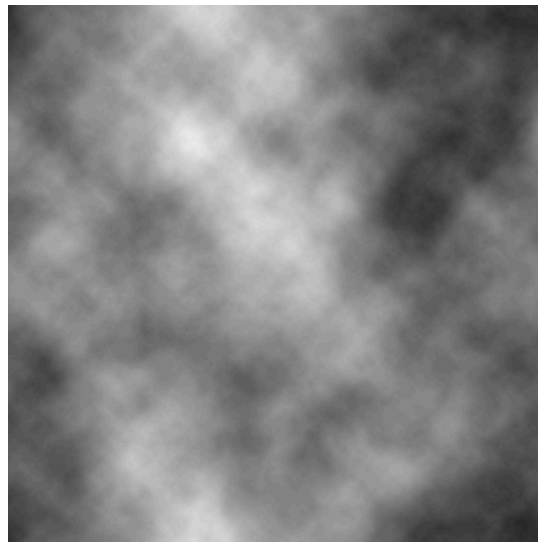
It is recommended to not play around with the effect's alpha clip threshold as we need it to be 0 for the effect to work. Whenever you make a new material of the shader, make sure that it doesn't cast shadows, has alpha clipping disabled, and optionally, use GPU instancing if possible for a bit more performance.

The effect uses two components, a light gameobject and a mesh object with the shader on. The light gameobject is the object that holds our actual light within the unity environment while the mesh object acts as an additive layer on our light by overlaying the light with a transparent mesh to simulate the look of a volumetric light. You need both to sell the effect effectively. But if you don't want to create your own custom prefab, you can use one of our provided prefabs to insert into your scene.

Do note that changing the mesh object's light color doesn't not change the color of light emitted by the light object, you will need to manually change both the light color in both the light and mesh object respectively.

## Creating and Using custom textures

The shader effect allows you to add in some noise to the light to make it look nicer. By default, the option to have it on is enabled, but if you want it off, you can tick off the Add noise to Light option. Now, you have an option to use a texture or a simple noise texture built into the shader. If you want to use a custom one, make sure that it is grayscale only with no transparency. Everything must be either white, black, or gray for the effect to work. An example of this would be the texture shown below which is a provided texture with the effect.



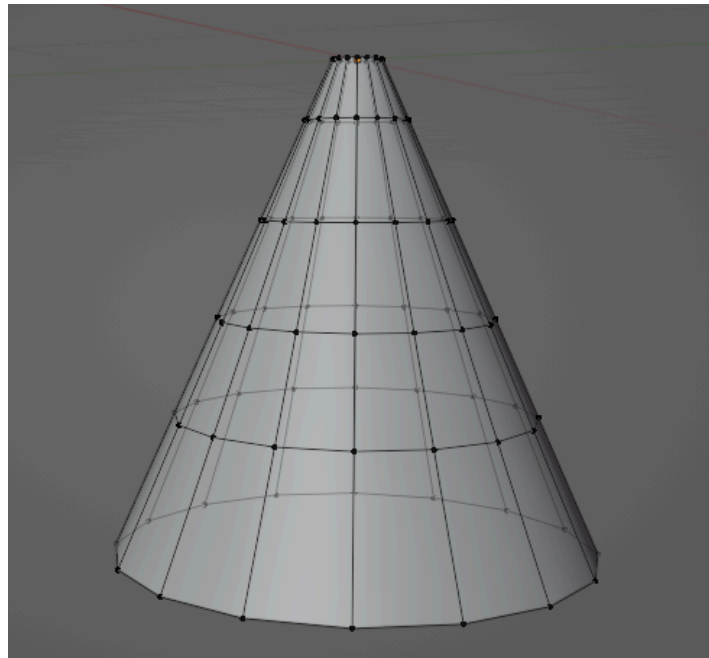
You can use all kinds of textures but it is recommended to be similar to a cloudy like texture, as shown in the provided example. Do note that you can also fine tune the strength of the noise by the noise strength parameter. It is also recommended to use textures that are powers of 2, like 512x512 and 256x256.

## Creating and Using custom mesh

Since it uses a mesh to replicate the effect, you can use all kinds of meshes to apply the effect on. The provided mesh with the effect is a simple cone shape mesh. If you want to use a custom one, you will have to create one from 3D softwares like blender or maya. You can also use Unity's probuilder, but currently as of testing it on Unity 6 (at the time of writing this document), the mesh it outputs is not as great as those from the older unity versions (the probuilder mesh is a huge mess in unity 6 compared to unity 2022), so it is

highly recommended to create the mesh for the light in software like Blender. The custom mesh can be of any shape that you like and can be exported as OBJ, FBX, or GLTF as long as unity can import them or you have tools to import them into unity.

When you create the mesh, if you want your light to look smooth, make sure that the mesh doesn't contain too many hard edges. A good example of a good mesh for this is from the provided mesh in the effect, as can be seen below.



If you have any hard edges in the mesh or problematic topology, it can be apparent when you have the effect on. Unless you prefer the hard edgeness in your lighting, make sure that the mesh is clean and is smooth. You can check it before importing by observing how it looks in your 3D software, if its lighting and shading looks off, the mesh still has a problem. Also make sure that its UV is good to avoid any potential issues that could arise from problematic UVs.

Once you import it into Unity, you can click and drag the mesh into the scene as an object, which you then can apply the effect onto the mesh by clicking and dragging the material onto the mesh.

## Update Log

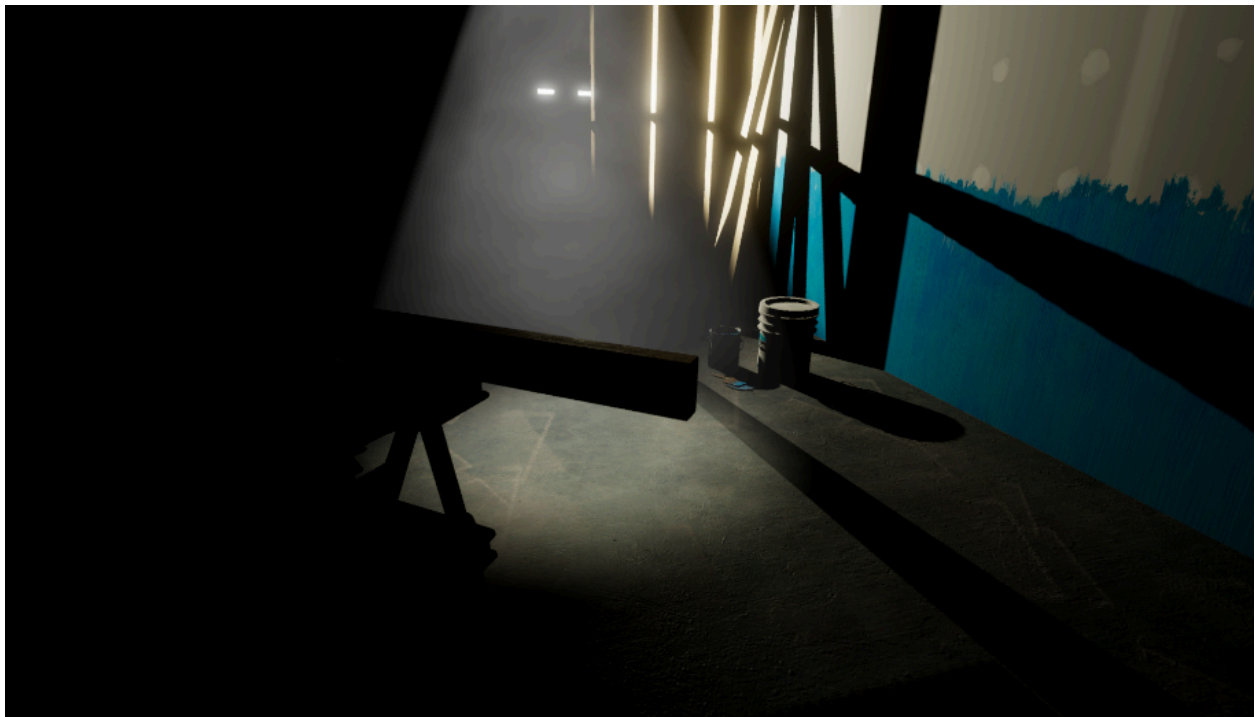
V0.9: The base implementation of the effect

## Credits

Some of the shadergraph code is based on MrTriPie's Unity Volumetric Light Shader Graph for LWRP Youtube Video (<https://www.youtube.com/watch?v=rihJzWq7sE4>).

The provided volumetric light shaft mesh is also from MrTriPie's Unity Volumetric Light Shader Graph for LWRP Youtube Video tutorial through its provided download link.

## Example Screenshots



What the effect looks like normally