Surprising Volumetric Effects in Cinema 4D*
Part 2: Erupting Plasma and Sliced Clouds

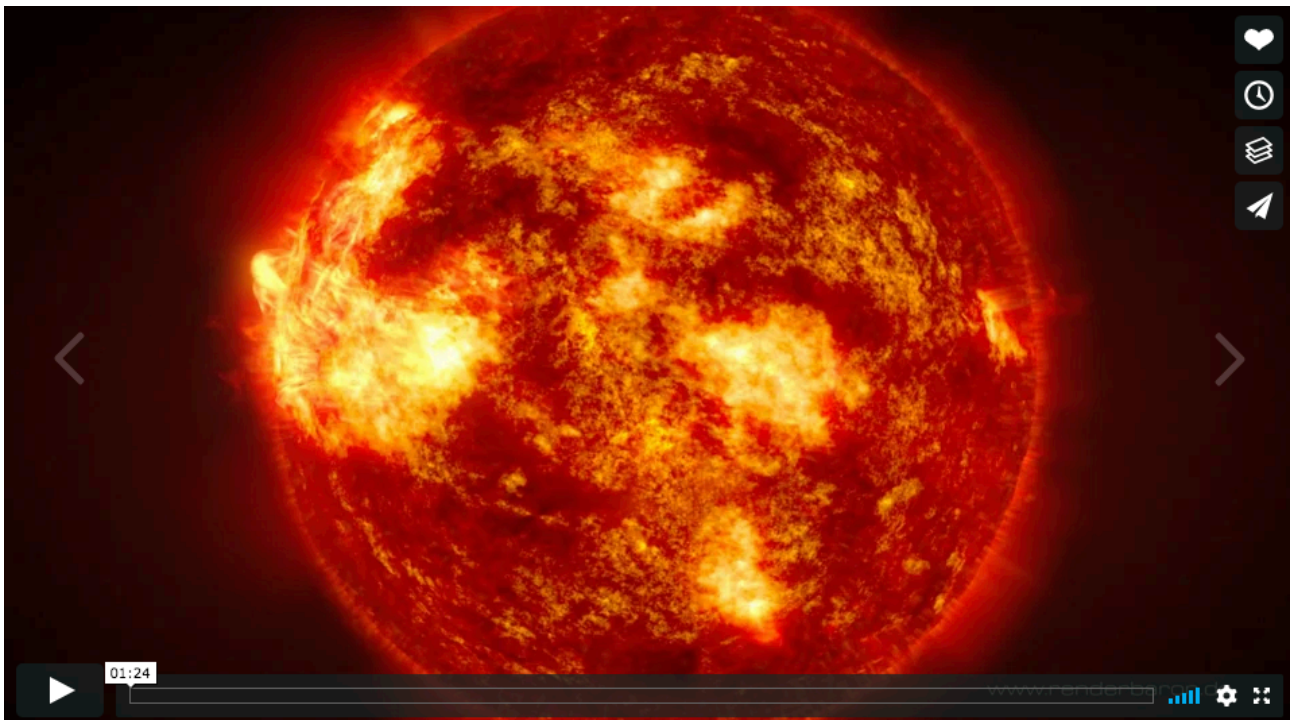
by Marc Potocnik, Intel® Software Innovator

Creating Solar Flares and Puffy Clouds in 3D Digital Art

The Sun - Animated Use of Visible Lights with Volumetric Shaders Part 1: The Magic of Visible Lights explained how visible volumetric light can be used as an “aquarium” for three-dimensional shaders such as noises and gradients. This type of Light Container enables the creation of non-shadow casting clouds or even complex self-illuminating structures such as stellar nebulae. This includes creating complex moving and growing structures such as an animated sun corona with its characteristic solar flares and eruptions. This article expands on what was covered in Part 1 and delves into creating more complex visuals like a realistic fiery and detailed sun.

Case Study: A question of Time (Sun)

Looking at contemporary science fiction movies or TV documentaries you will find that depicting a realistic animated sun with a high level of detail can be a demanding task for a visual effects (VFX) studio - a classic scenario for particles and fluids and many sleepless nights of simulation. However the intro-sequence of this german TV-documentary ZDF „Terra X" staging stellar nuclear fusion only uses the techniques we´ve in Part 1.



Browsing through NASA footage you will find that the surface of the Sun contains a lot of visual detail that can be recreated with Cinema 4D* noise shaders. For example, the pPhotospheres granule can be reproduced by a Displaced Voronoi and distorted by a VL Noise.
It's the same with phenomena that are more volumetric, such as solar flares, rays of plasma, etc. as noises and gradients can be used as the content of a volumetric light source.

A Concentric Scenario

In the stellar nebula recreated in Part 1, we restricted several phenomena such as "Small Fogs Blue" and "Small Fogs Violet" to different areas of the nebula by using 3D – Linear Gradients without worrying about texture projection. Contrary to that, texture projection matters when it comes to a spherical body like the Sun with different zones of activity and concentric light emission.

As volumetric light is now depicted by using concentric Omni Lights we have to take care of proper geometrical texture projection in the Texture Tag: UV - or in our case – Spherical. This is necessary for two reasons:

Light Rays

As for other volumetric effects we use the light source as a container for noises, also in order to create concentric light rays. Noises applied in Object Space would show their specific 3D structure in the volume of the Omni Light, but not as concentric rays, as shown on left in figure 1. Concentric lights rays are only possible if the noise is applied in UV (2D) mode as shown on right in figure 1. This requires a correct geometrical texture projection in the Texture Tag: UV or – in our case – Spherical.

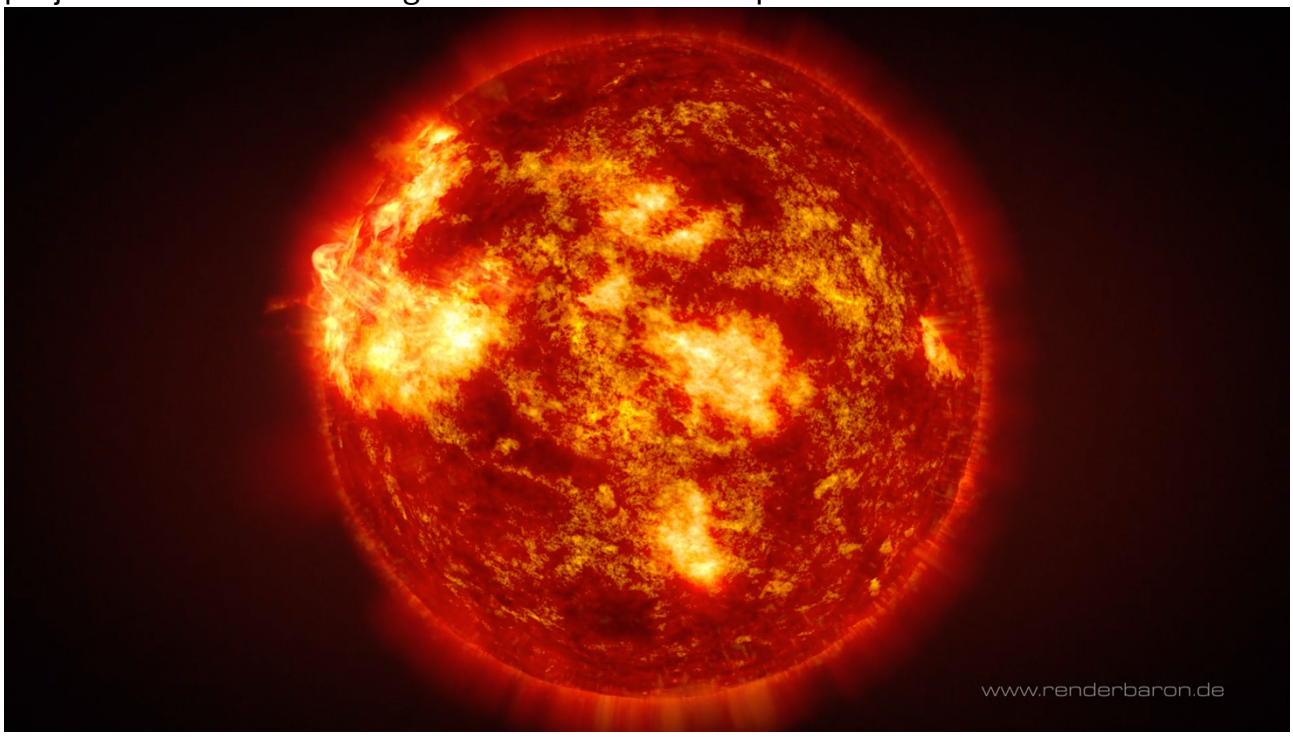


Figure 1: On Left –Noise in Object Space, On Right –Noise in UV (2D) space with spherical texture projection

2D Masks When restricting several phenomenon to the areas of the sun, such as the equatorial area, use a classic 2D (V) Gradient which also needs a UV or Spherical projection. Example: Solar flares are created with a 3D noise in Object Space but are masked by a 2D (V) Gradient as shown in figure 2.

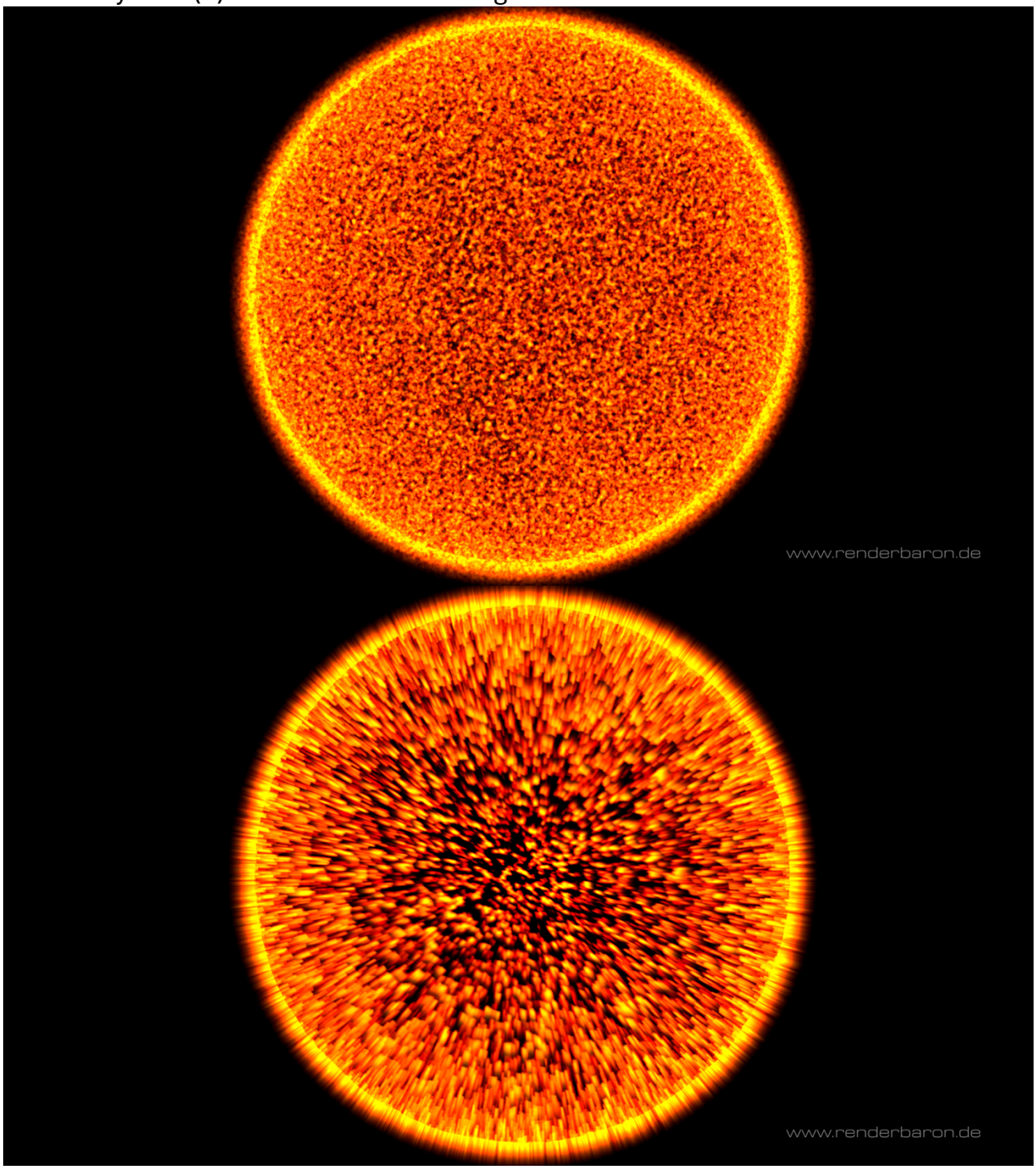


Figure 2: Shader properties for a solar flare

Simple Layers of Complexity

As with many visual effects, complexity stems from layering and combining different simple elements. For visual phenomenon such as light rays of different sizes, solar flares, eruptions etc., use one Omni Light each, more or less bigger than the sun's body as

shown in figure 3. All of the volumetric complexity is done with just eight volumetric Omni Lights, as shown in figure 4.

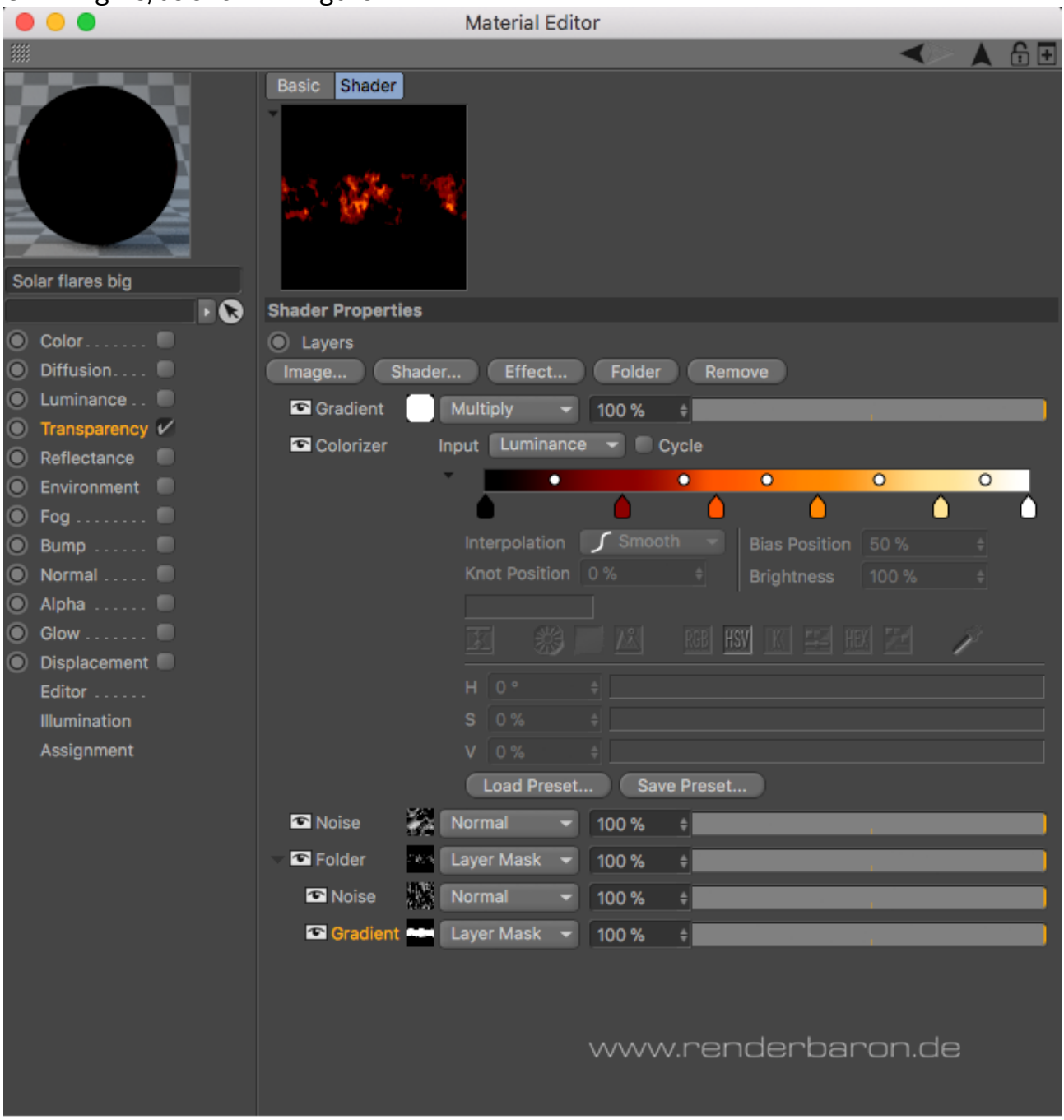


Figure 3: Light rays using Omni Light

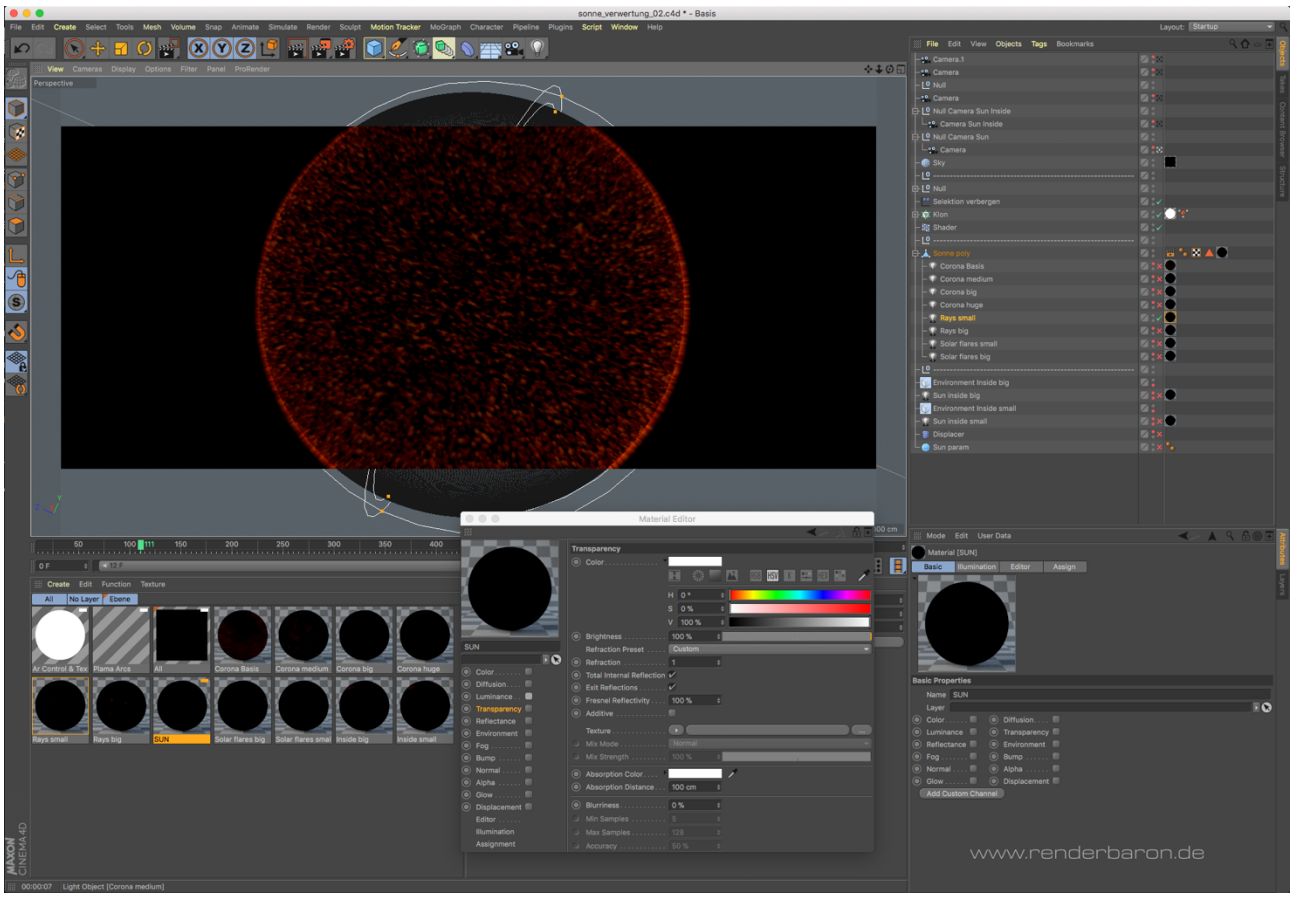


Figure 4: Volumetric complexity done with only eight Omni Lights

Have a look at a Breakdown of them in https://vimeo.com/297784795.

Concentric Restriction

Coronas, light rays and solar flares are applied as textures on volumetric Omni Lights with the technique we know from Part 1. As the **Visibility Tab** of Omni Lights consists of the **Inner Distance** and **Outer Distance** parameters volumetric effects can be faded out concentrically from the surface of the sun out into space.

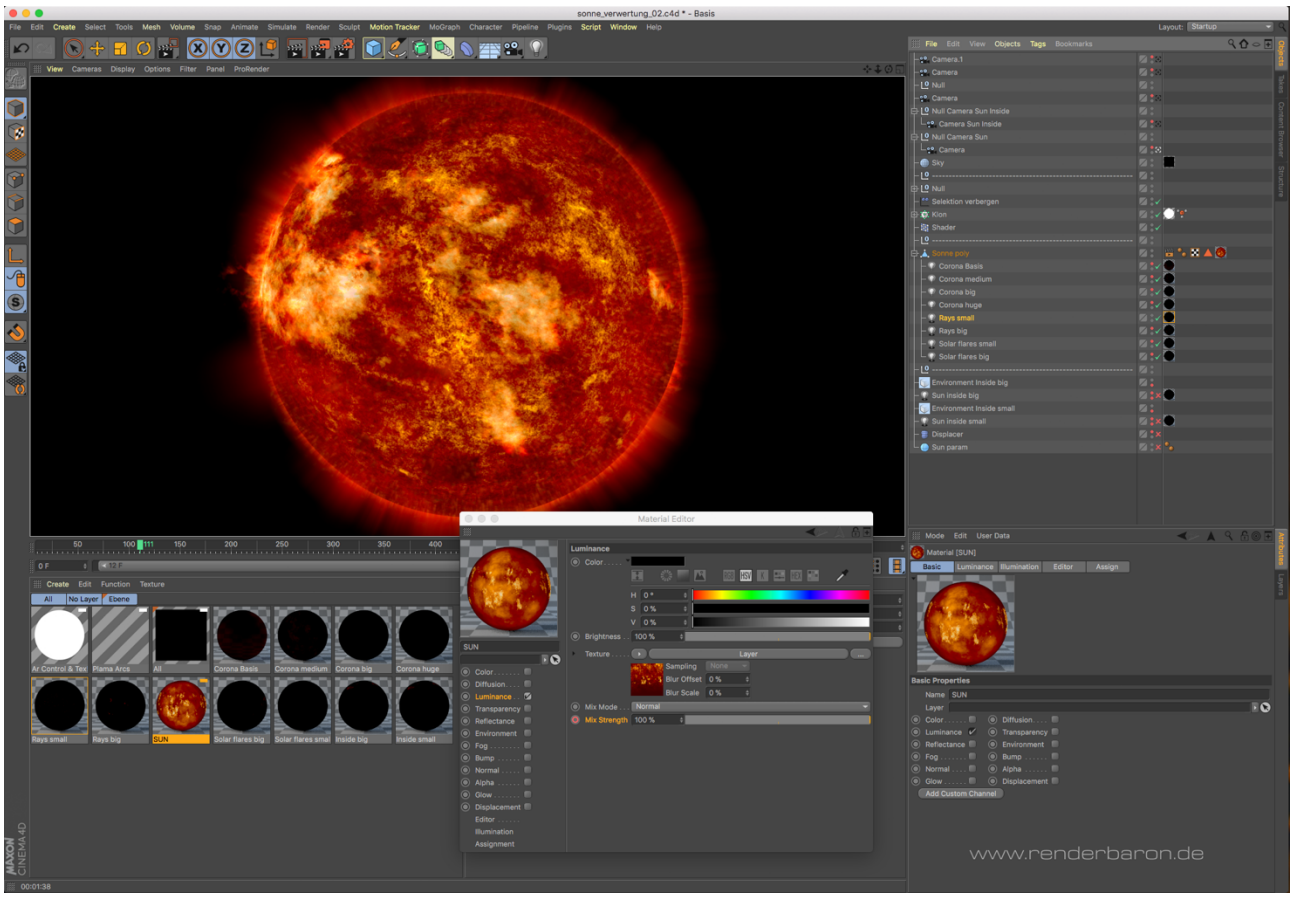
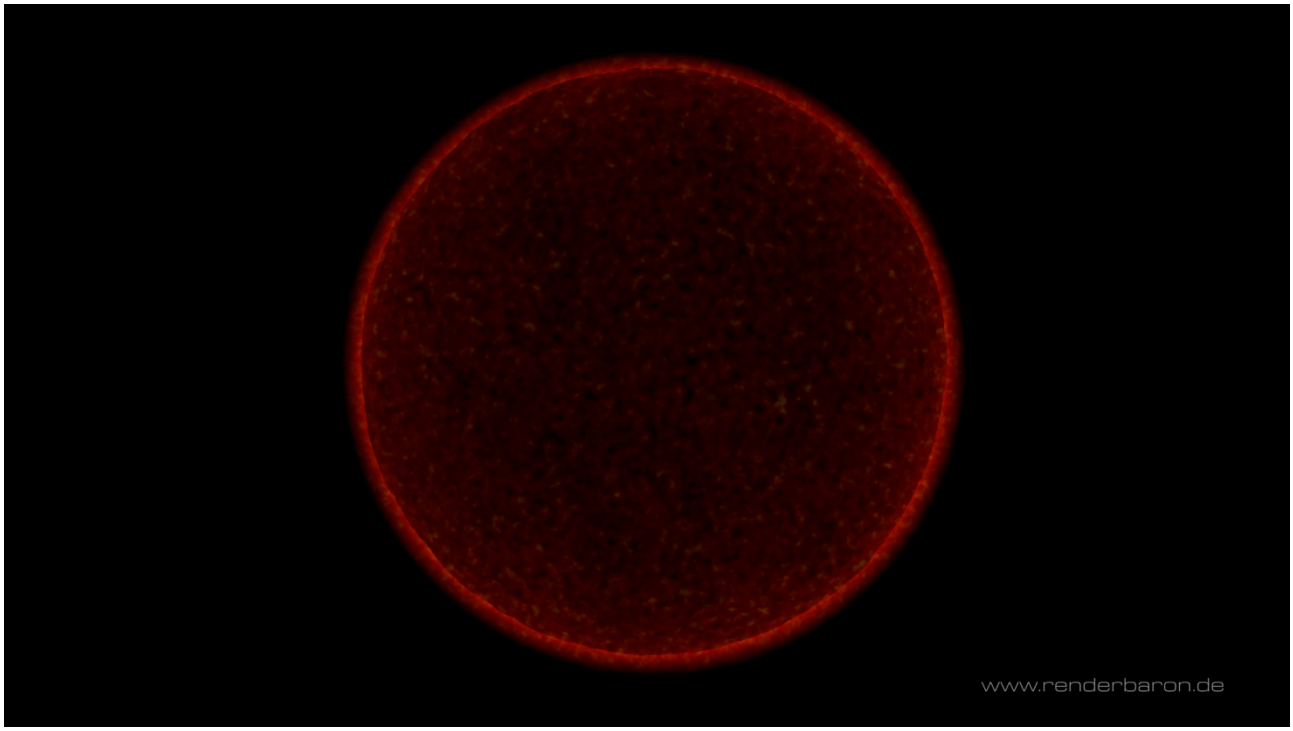
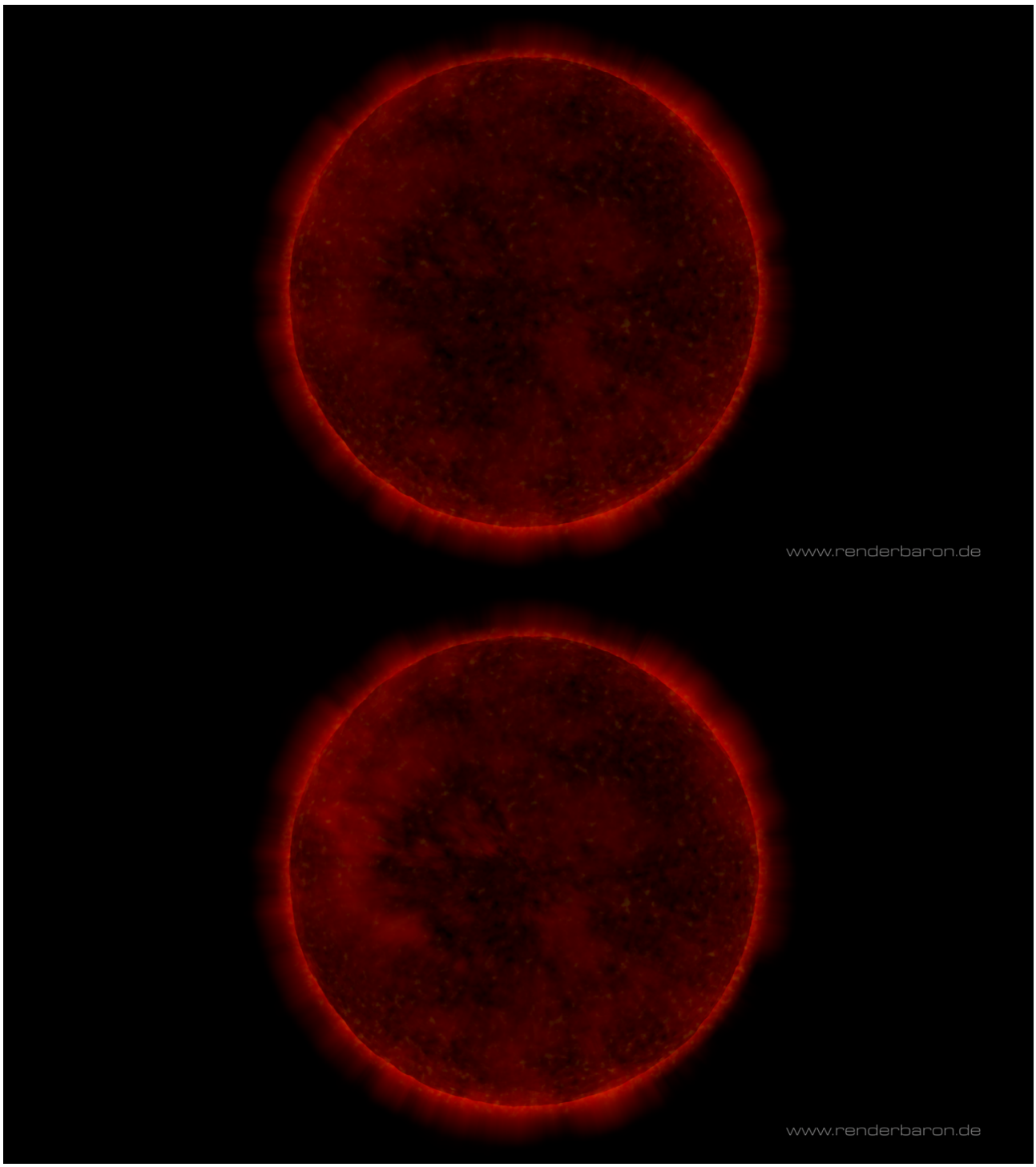
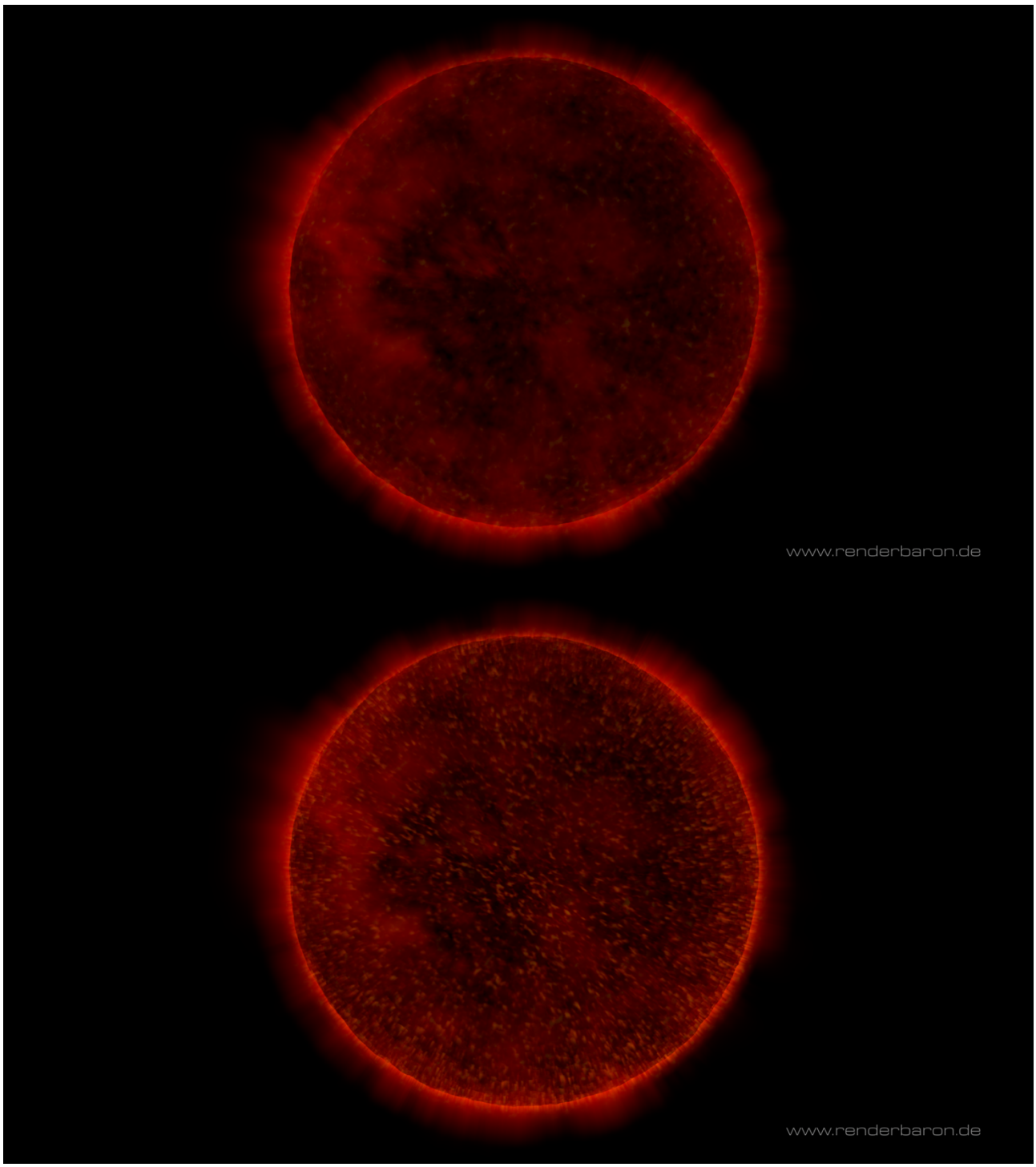


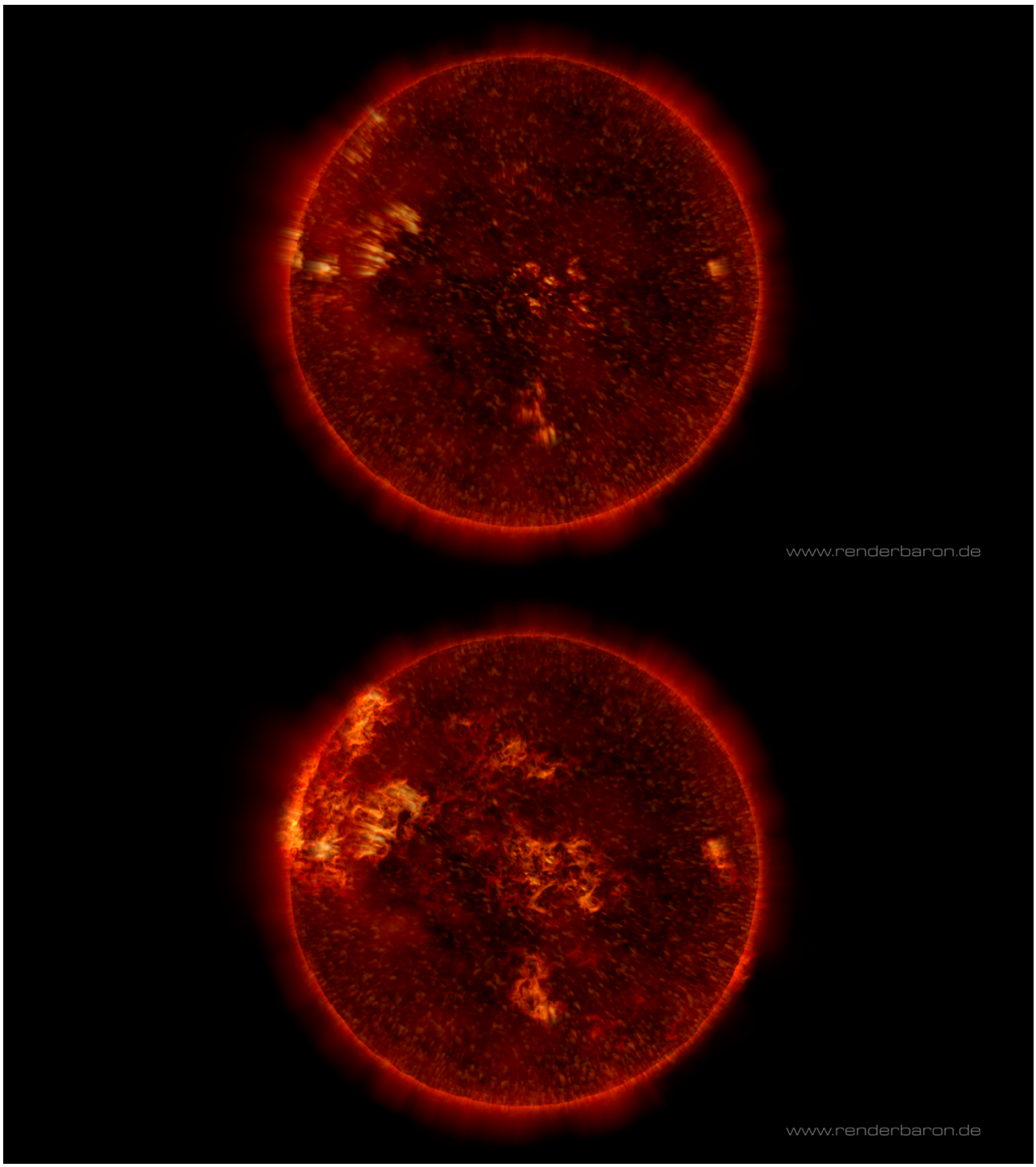
Figure 5: Lay settings for concentric restriction

To have more control about this behavior a **3D Spherical Gradient** can be applied as a mask in the texture, as shown in figure 8 and figure 9. This Gradient has the advantage of fine-tuning the spatial fade-out of the effects while offering the ability of applying a turbulence.




www.renderbaron.de
www.renderbaron.de


www.renderbaron.de
www.renderbaron.de


www.renderbaron.de
www.renderbaron.de

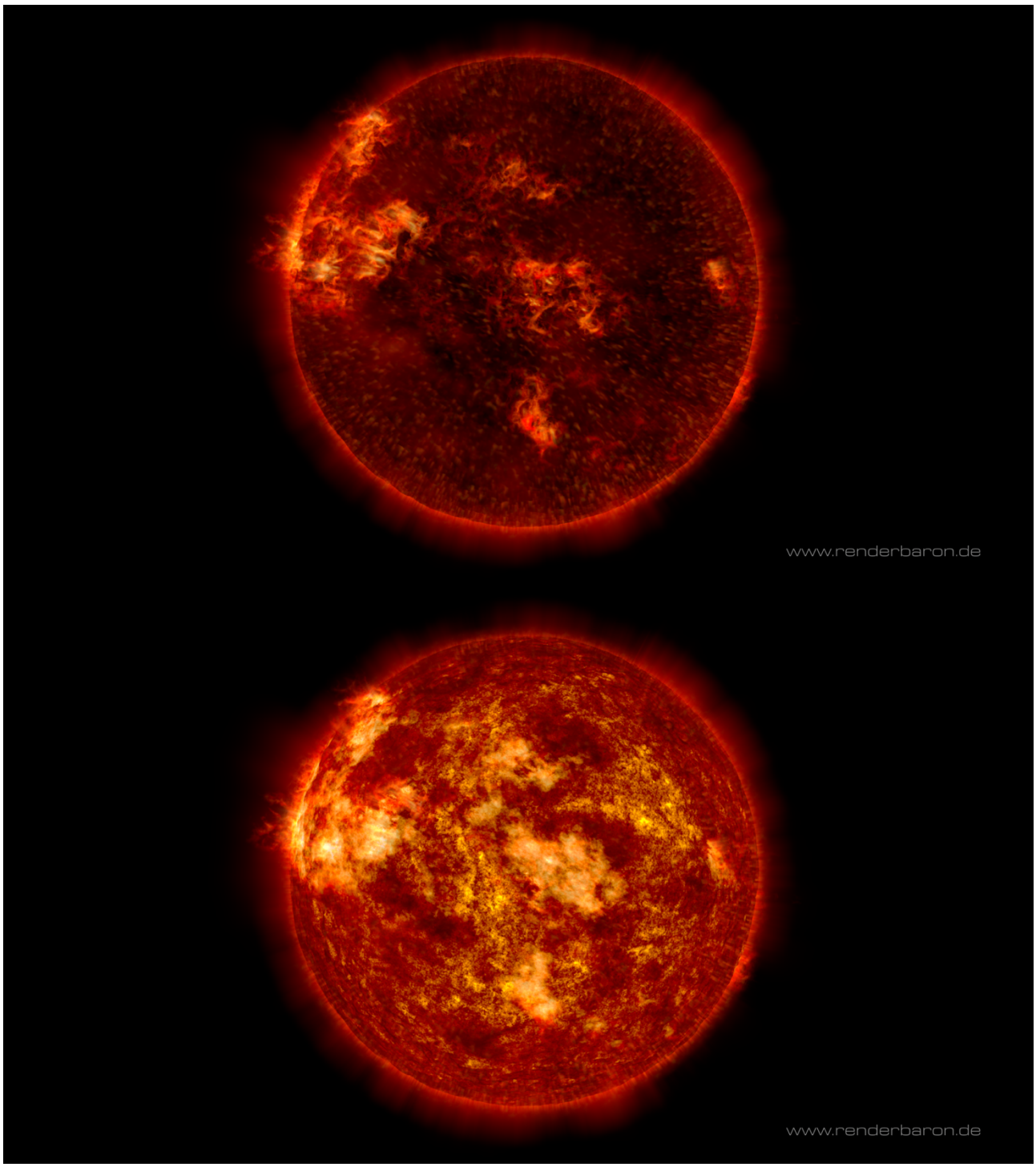


Figure 6: Masking a small scale Turbulence Noise with a spherical 3D-Gradient

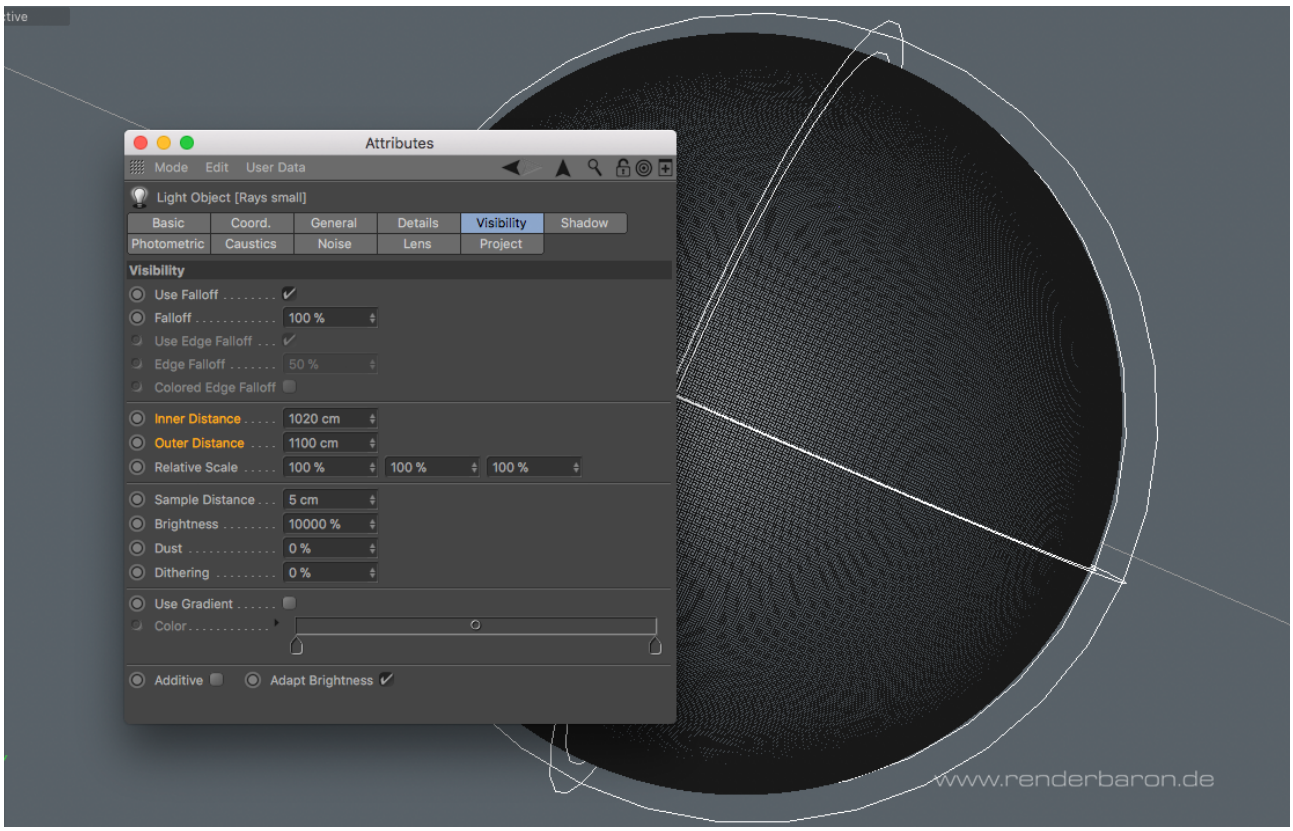


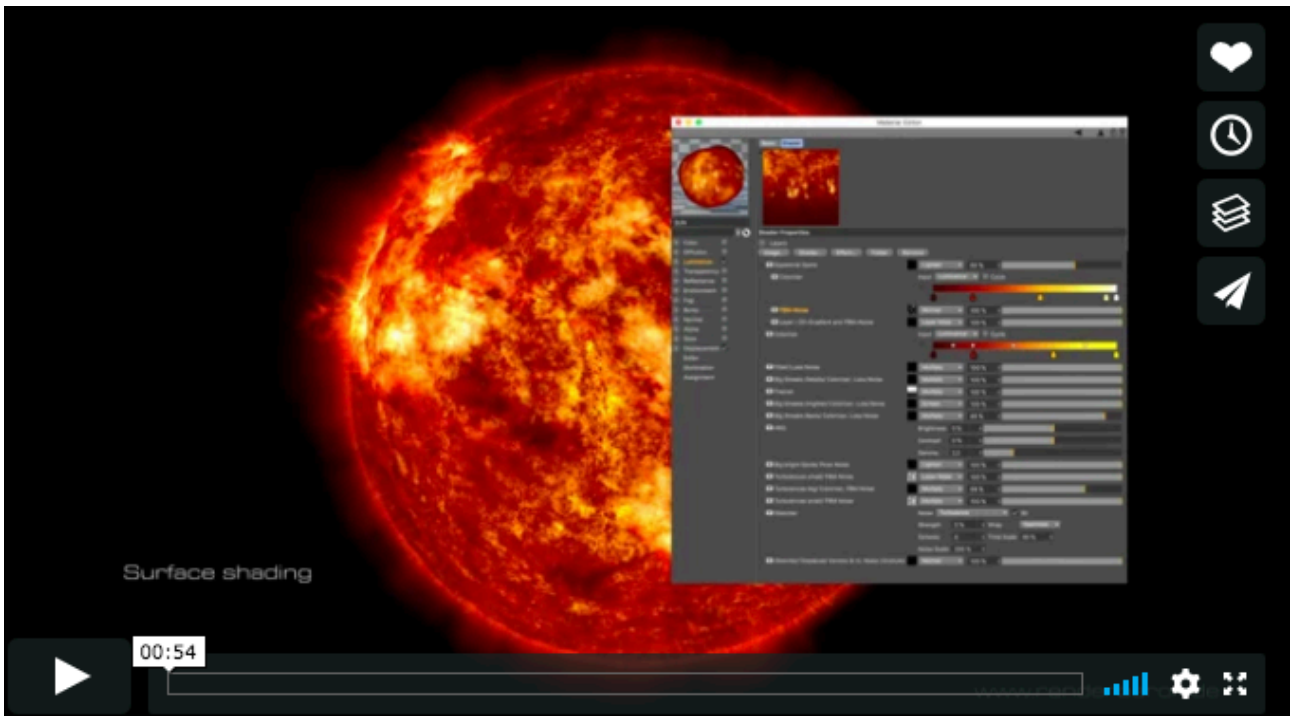
Figure 7: 3D gradient properties for small rays

Solar flares

The effect of solar flares may look quite complex but in fact are very simple, consisting of an Omni Light containing a large scale **Naki Noise** slightly moving by **Animation Speed** and used in **Object Space**. The Naki sits inside a **Layer Shader**, colored by a **Colorizer** effect and masked by a **2D - V Gradient** to the more or less equatorial area, as seen in figure 2 above.

The **Global Scale** of the Naki Noise is animated over time, meaning that the noise is concentrically growing from the center of the sun. This alone is already creating the illusion of plasma-like large scale structures.

To have more control over the way solar flares fade out in space, the growing Naki can be masked with a static but highly turbulent **3D Spherical Gradient**, also applied in Object Space. The result will be solar flares that are evolving from the sun and turbulently fading out into space.



The Surface of the Sun

Use the shader setup to create an effect similar to the actual surface of the sun, with everything taking place in the Luminance Channel. Similar to volumetric shading, a Layer Shader is key here as noises, gradients and layer shader effects are stacked onto each other with different layer modes and opacities. Also, several Layers are used to mask the ones above as shown in figure 8.

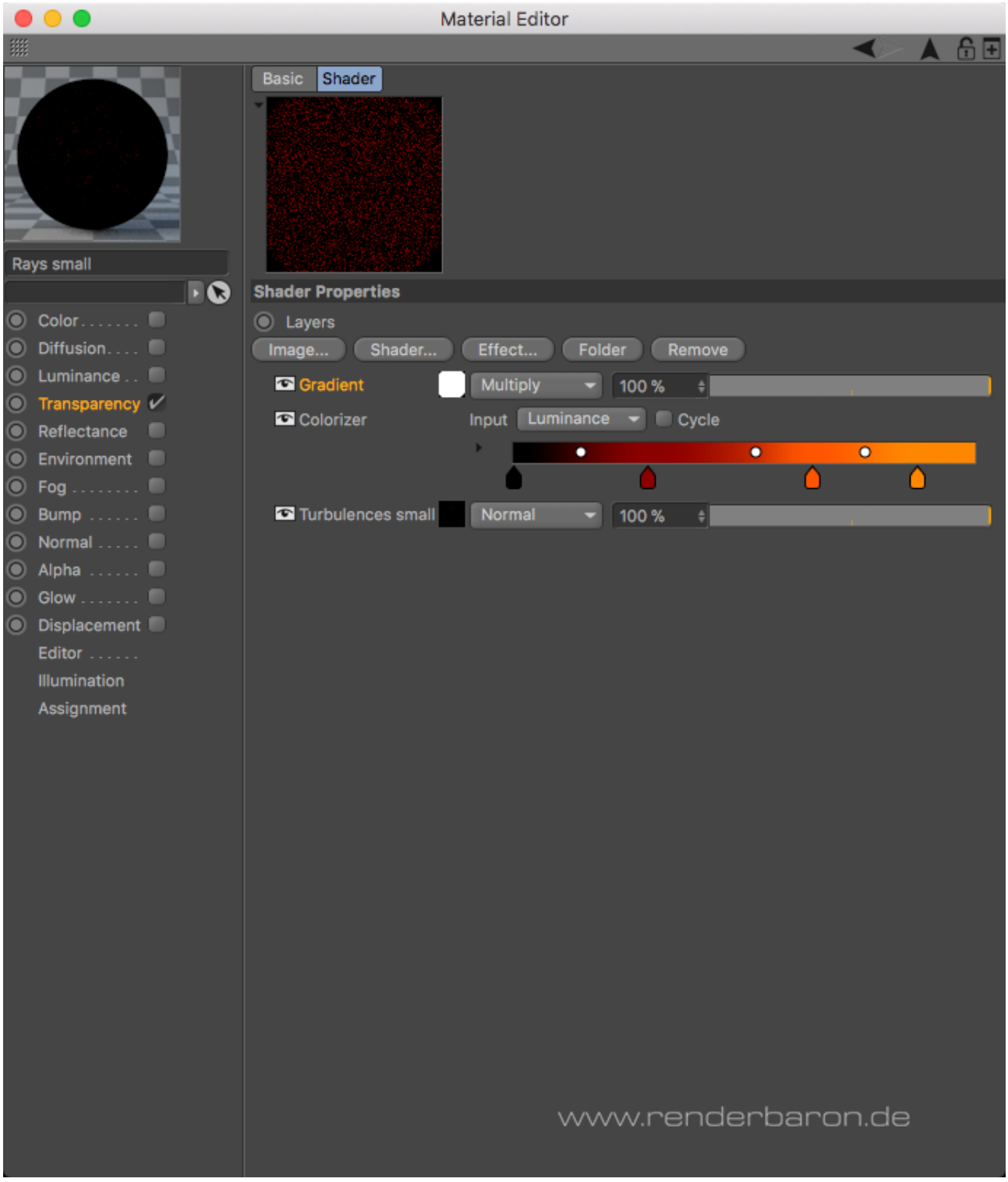


Figure 8: Shader properties for the sun´s surface

In the Displacement Channel, several noises inside the Layer Shader are deforming the polygonal Hires geometry of the Sun according to the visual details of the Luminance Channel as shown in figure 9.

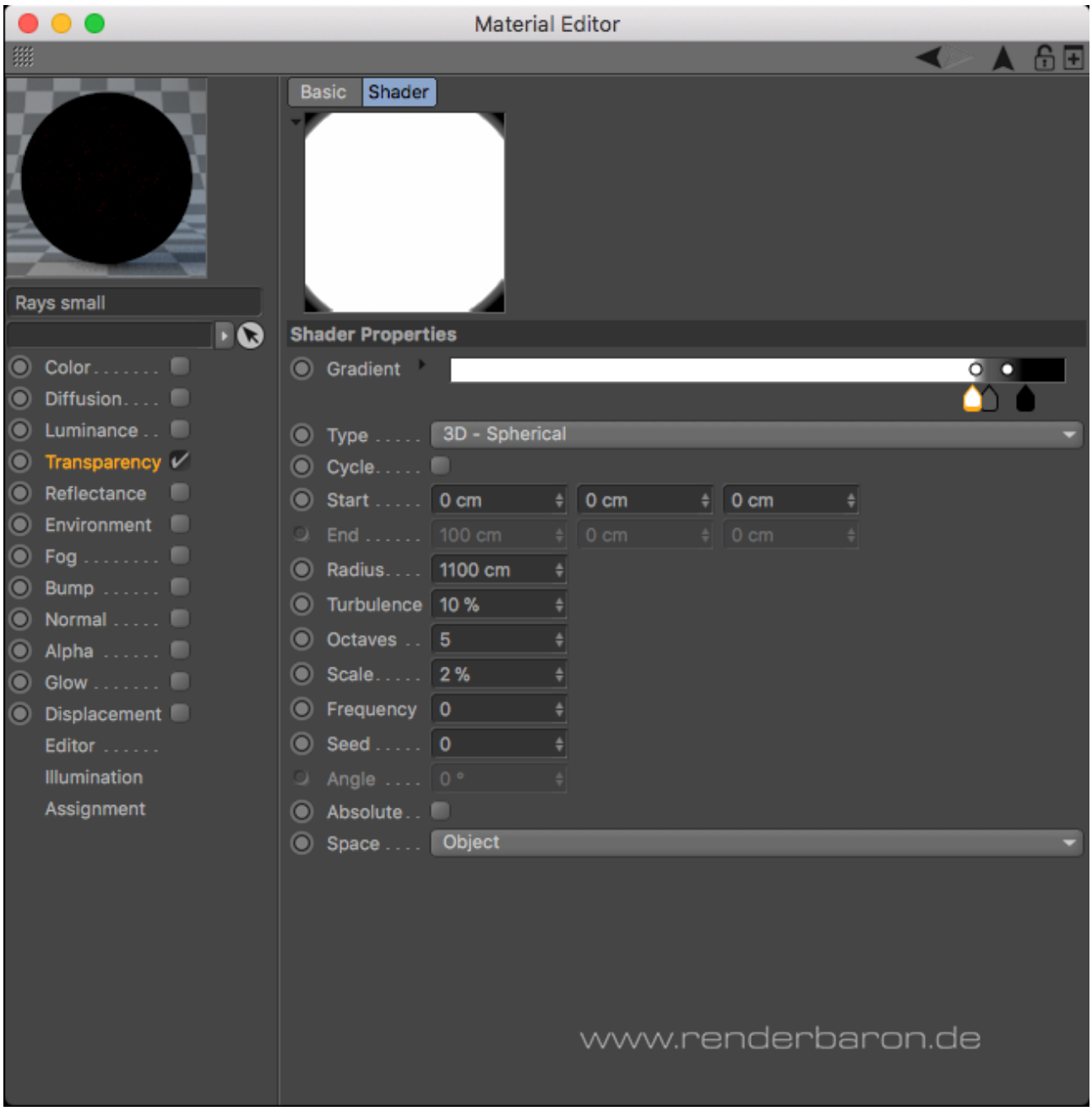


Figure 9: Parameters in the Displacement Channel

Large scale arcs of plasma are procedurally modeled by connecting MoGraph Clones across the surface with the pCONNECTOR plugin for Cinema 4D and putting them into aSweep Object.

Traveling into the Sun

In Min 00:11 the animated camera is entering the volume of the sun. This transition is done in compositing with an animated blending to a sequence similar to the stellar nebula discussed in Part 1. It is important to note that for this transition the end velocity of the outside camera has to have the same direction and linear speed as the start of the inside camera.

Fast Rendering with Standard Renderer and Intel® Embree Acceleration

Sample-based volumetric light is a feature of CINEMA 4D since Release 5 (1998) and this kind of sample is different than modern samples such as area shadows, rough reflections and ambient occlusion. Accordingly CINEMA 4D handles them differently, meaning: rendering this scene the „older" Standard Renderer is by far faster than the „younger" Physical Render.

In addition to that, the Standard Renderer has been accelerated by Intel® Embree technology since Release 19. Intel Embree is a ray tracing library optimized for the latest Intel® processors. Rendering speeds are significantly faster - in some case up to 100% compared to Release 18. Intel Embree is breathing new life into the veteran Standard Renderer.

Noises and procedural shaders also have a SAT texture interpolation by default, so for rendering there is no need for anti-aliasing other than geometry quality. In other words, Standard Renderer is rendering a scene at light speed – especially related to the volumetric complexity created here.

Puffy Clouds - MoGraph Cloners as Containers for Volumetric Shader Setups

The technique applied for both the stellar nebulae and the Sun made use of the volume of a volumetric light source as a container for 3D noises. While the use of such a light container is actually straight forward and easy to use, it has one downside - all created structures and effects are actually visible light and are not capable of being opaque and casting shadows. So if you create a cloud with this technique this cloud could never cast a shadow on the ground or on itself.

To create a typical shadow casting puffy cloud for a sunny, summer sky use a technique that uses the same basis – volumetric shaders – but a different container: a MoGraph Cloner.

MoGraph – Procedural Cloning and Animation

Within the Cinema 4D proprietary toolset, MoGraph, offers motion designers a rich palette of powerful tools for procedural non-destructive animation. A core function of MoGraph is the cloning of objects by certain rules, such as along a spline, in a honeycomb array or on the surface of an object. Effectors such as random, shader or time bring variation and movement to the clone system.

While MoGraph offers a huge palette of exciting functions inviting you to play and experiment, the technique explained here is probably the most boring one: cloning simple planes to a linear stack which will be the container for the 3D noises.

Case Study: Scotland - The Myth of the Highlands

This project for german TV-documentary ZDF „Terra X" features the geographical origins of Scotland and England. . Volumetric clouds and atmosphere in the “Close-Up” shots are created with the MoGraph-based technique shown below.

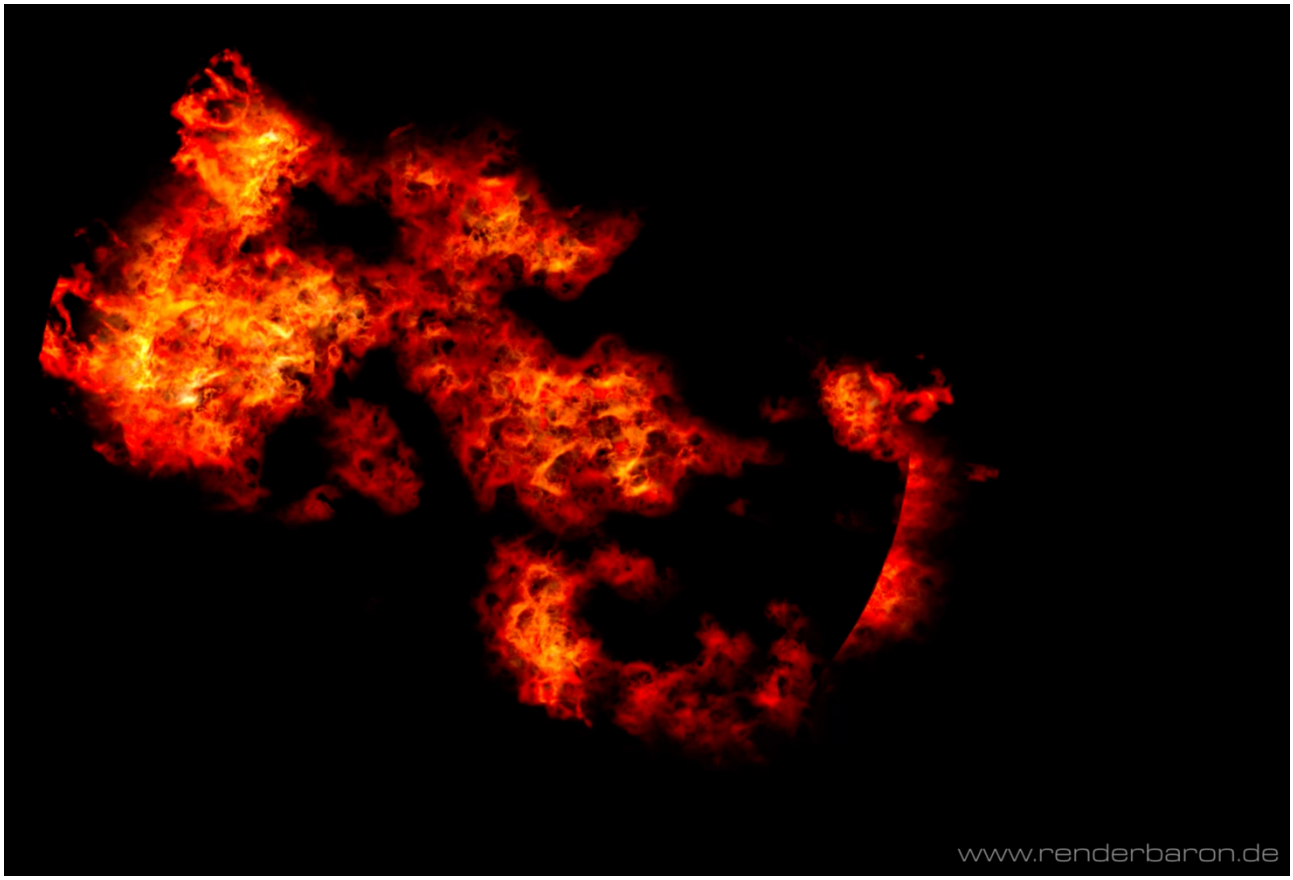


Figure 10: (Video)Volumetric clouds seen from space - created with MoGraph and 3D.

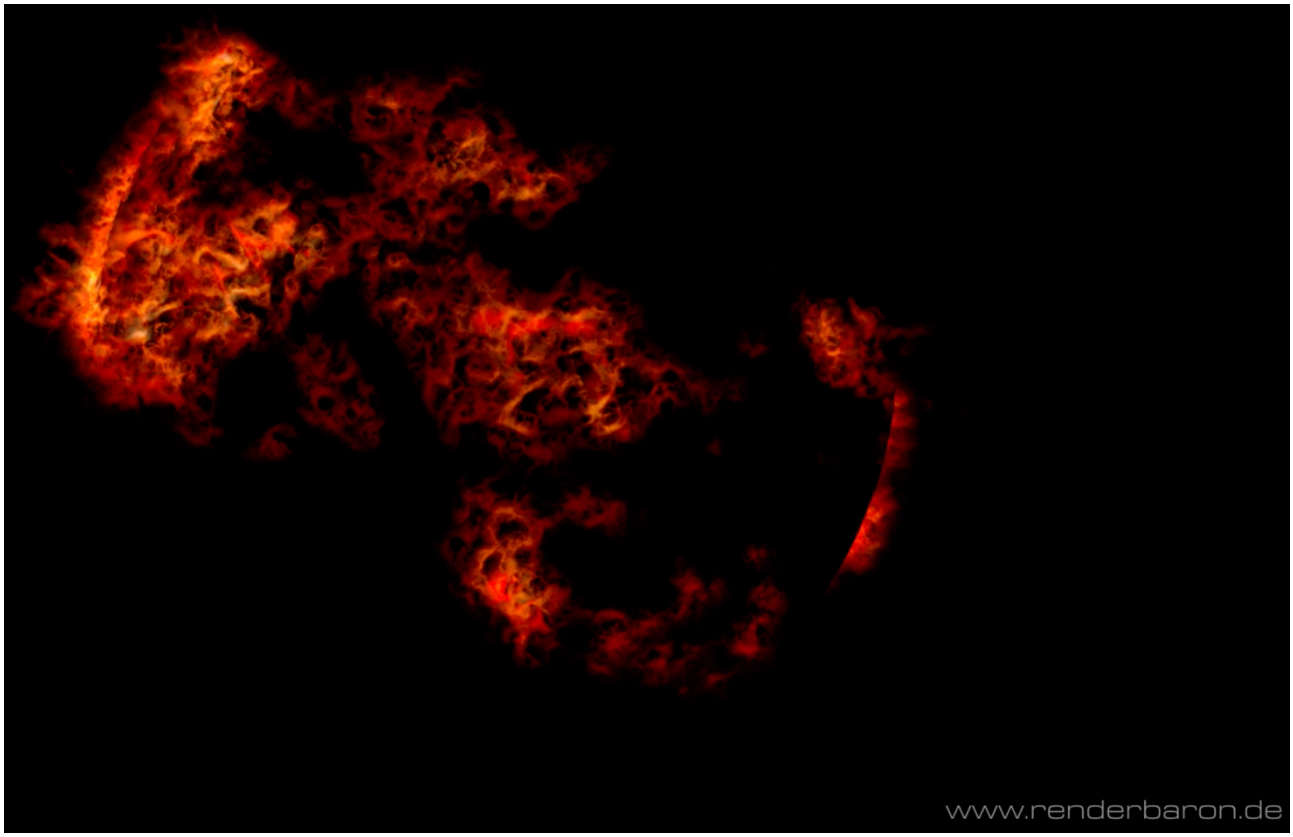


Figure 11: (Video) Volumetric clouds created with MoGraph and 3D Noises above Loch Ness, Scotland.

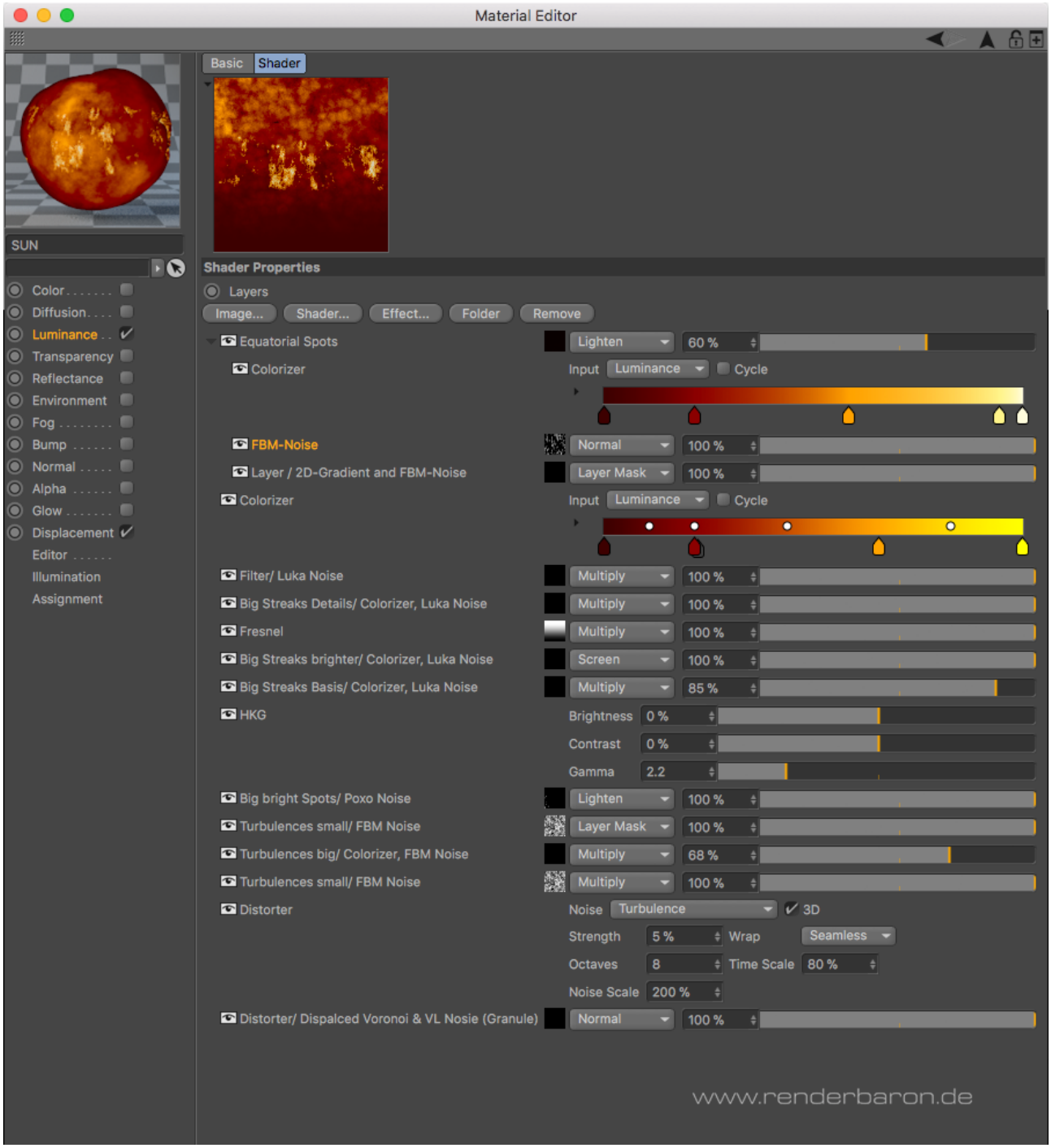


Figure 12: Volumetric clouds created with MoGraph and 3D Noises above Orkney, Scotland

Sliced Puffy Clouds Tutorial

In order to not only understand this technique but to get your fingers in it, follow along with this short tutorial.

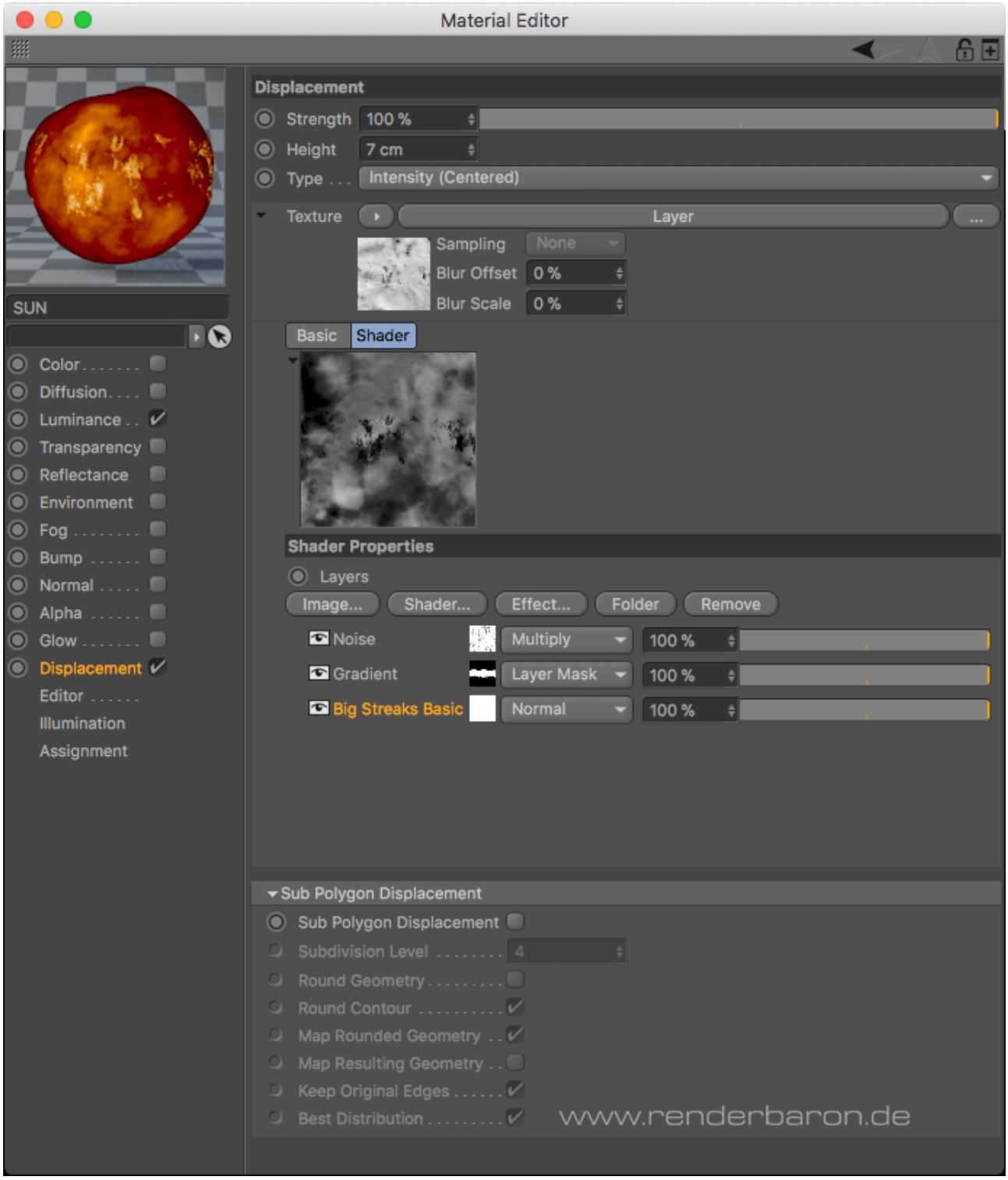


Figure 13: Color gradient for the sky object´s material

Creating an Environment

Create a new scene with a Sky Object and apply a 2D Gradient to it, an example can be seen in figure 13 above. Place a Plane Primitive with 1 x 1 segments and dimensions of 1200 x 1200 cm at -200cm in World Space. Apply a Material with blueish color. Create a camera and look through it from a flat top view. The empty scene should look like the example in figure 14 below. Create an Infinite Light and apply ray traced shadows for

now - just to have a fast Shadow effect, no matter how unrealistic it is.

Figure 14: Example of the empty scene

Stacking Planes
Create a MoGraph Cloner under Main Menu/ MoGraph/ Cloner and subordinate a simple Plane Primitive with the same parameters as the floor plane. Instantly your Plane gets cloned 3 times along the Y-axis of the Cloner.

Click on the Cloner in the Object Manager and go to the Cloners Attribute Manager. You will note that Cloning Mode is set to Linear by default, just the way we want it. Leave the Clone Count for now at 3. Set the Distance Mode from Per Step to End Point, go to P.Y and type in 100 cm. This distributes all cloned Planes along the Worlds Y-axis inside this range.

Creating a Cloud Material
Create a new material by activating the Materials Alpha Channel and create a noise shader. Click on the noise preview. Inside the noise choose Naki, set Space to World and the Global Scale to 2500%. Set Seed to 668, Low Clip to 50%, and High Clip to 100%. This narrows the levels of the noise a bit so you get nicely separated white/opaque clouds inside a black/transparent Space (see figure 15).

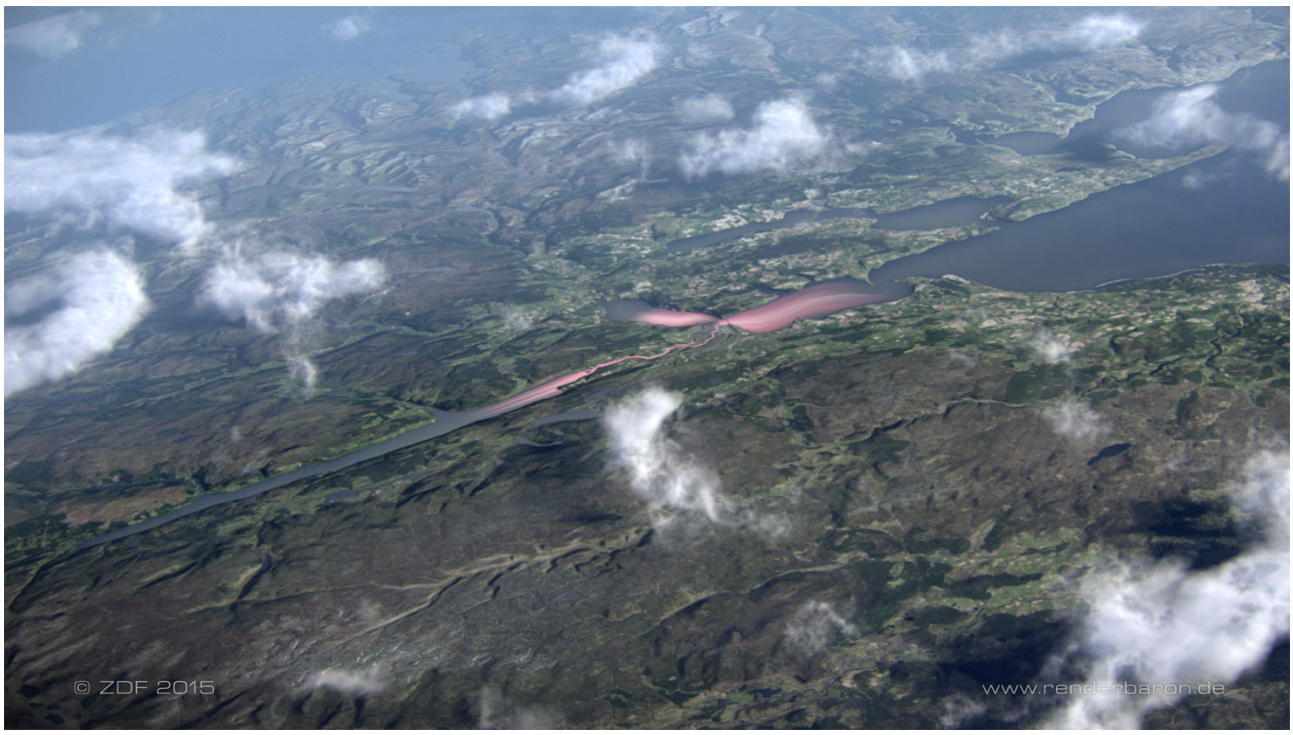


Figure 15: Parameters of the clouds basic noise

Adjusting the Viewport for Better Interaction
Go to the Viewports Menu/ Options and choose Enhanced OpenGL (if not already activated). Choose Noise and Transparency.

Making it Puffy
Apply the material to your cloner. You will now get a different display of the noise, change from plane to plane.

Increase your clone count step-by-step to 50 and see what happens... Lightbulb! The slices of the MoGraph Cloner – the Planes - act as a kind of “spatial samples” for the 3D noise, creating sliced puffy clouds. The more clones you apply, the smoother the result gets. So count is now acting as a kind of “Sample Count”.

Increasing Ray Depth
When rendering you will get some nice, soft cloud puffs, but you will note that here and there black sharp artifacts are appearing, as shown in figure 16. This is because the Ray Depth of the Ray tracer is used up. This means after passing a limited number of transparencies (our cloned Plane with Alpha-Channel on) the ray quits and renders a black Pixel.

Figure 16: Sharp black artifacts in clouds

To avoid this, go to your Render Settings (Cmd – B) and navigate to Options. Adjust the Ray Depth parameter to the count of your clones plus one, in this case 51. Do this for Shadow Depth as well to ensure that shadows are not cut off behind a certain number of transparencies, as seen in figure 17. Because this is increasing the render time we use a ray traced shadow in the light source for now. You can change it to realistic Area Shadows for final rendering.

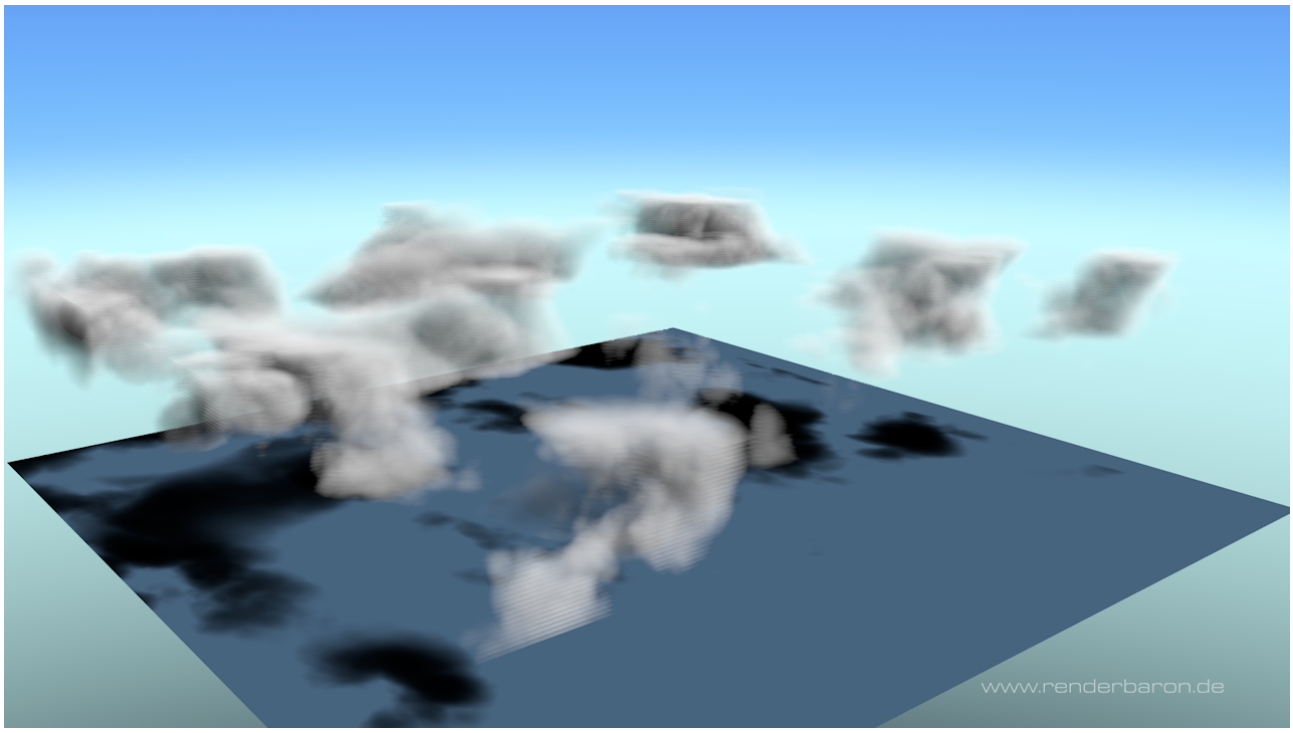


Figure 17: Changing the ray depth and shadow depth can avoid the black artifacts

Restricting Clouds Vertically
What we can clearly see is that the clouds seem to be cut off on top and bottom by the vertical borders of the cloner stack. To circumnavigate this mask the 3D noise with a vertical 3D gradient.

In the Alpha-Channel of your cloud material click on the triangle button beside “Texture” and choose Layer. By this you have put your noise shader inside a layer shader. Briefly head back to your Alpha-Channel and uncheck image alpha to ensure that the greyscale information of the layer shader is considered as Alpha-Information.

Inside the layer shader double-click the noise and name it “Cloud-Noise”. Then click on shader and create a gradient shader. As type choose 3D – Linear. We now have to think of how to adjust the gradient to mask our noise from bottom to top correctly.

As the lowest plane of the cloner sits at Y=0 and the cloner itself has a height of 100 cm adjust the gradient on Y-axiz to start at 0cm and end at 100 cm. The gradient itself is going from black to White to Black as shown in fig. 18. Set the gradients mode to Multiply and render.

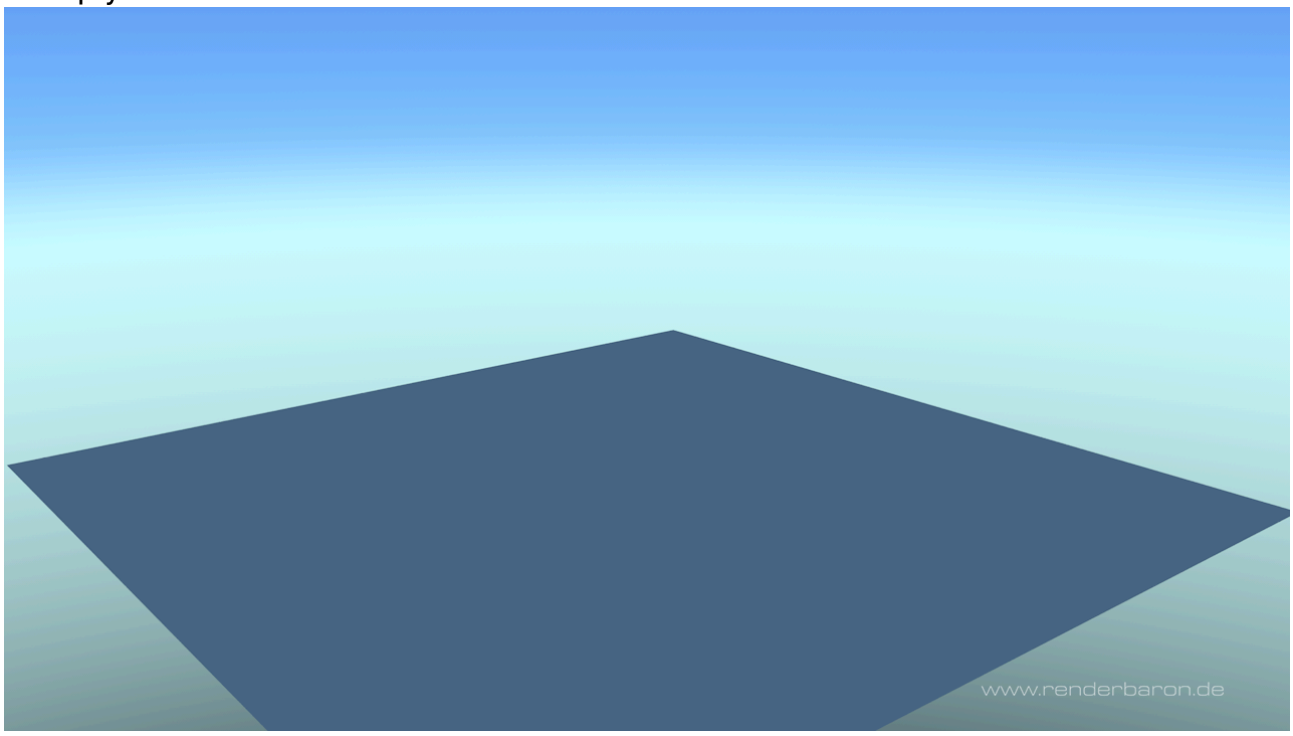


Figure 18: Gradient for vertically restriciting the clouds

The result is better as we now softly mask the Cloud Noise from the bottom to the top as shown in figure 19. But the mask is much too soft and uniform. Instead of applying a built-in turbulence to the gradient we will do something more sophisticated.

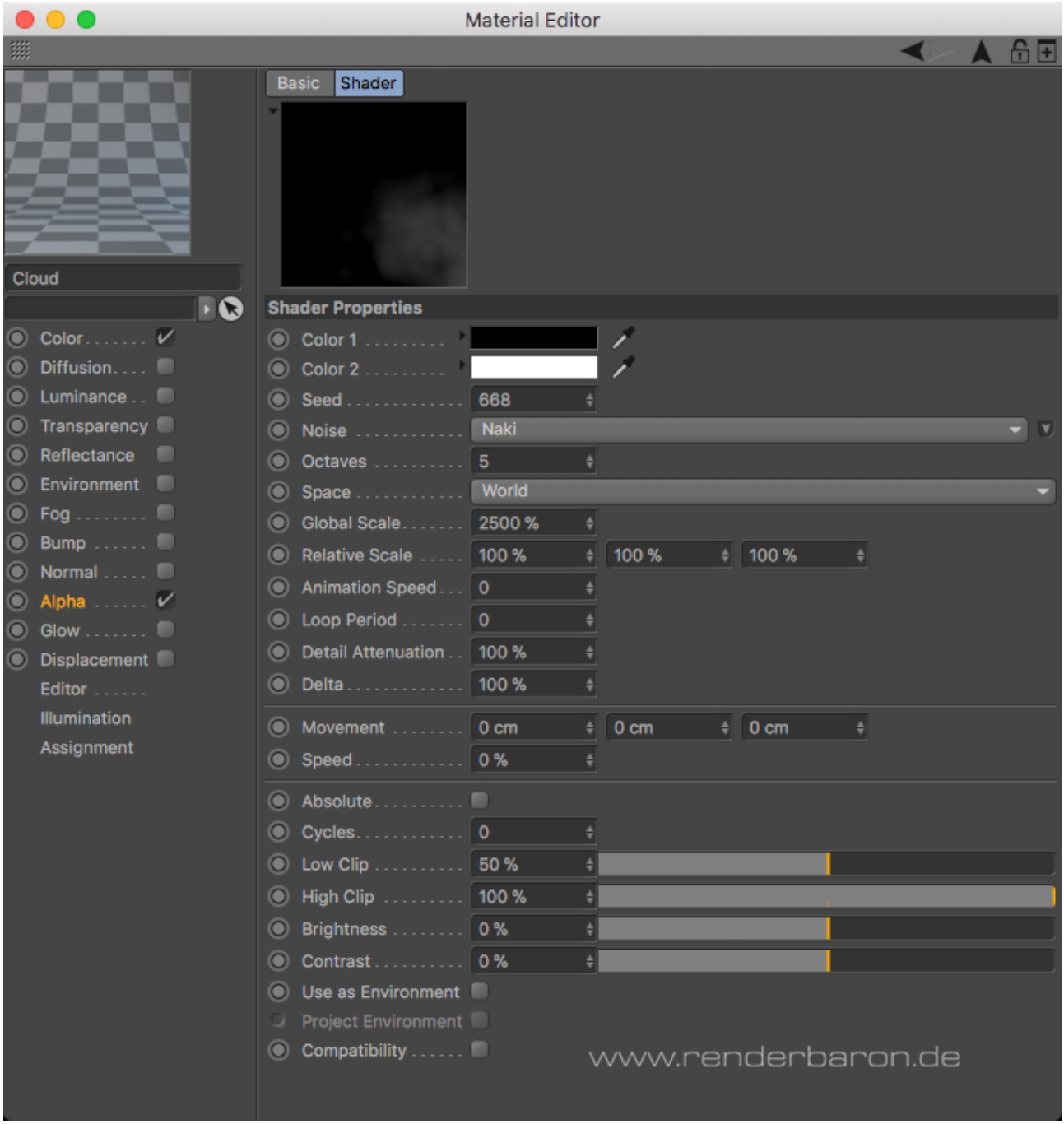


Figure 19: Softly mask the cloud noise

Inside the Layer Shader copy and paste Cloud-Noise by right-clicking it and choosing Copy Shader/ Paste Shader. Double-click the noise and name it “Mask-Noise”. Put it above the Gradient.

Adjust the noise to a Global Scale of 500% and invert it by tuning Low Clip to 85% and High Clip to 0%. Head back to the layer shader with the arrow button on the right top corner of the Material Editor. Set the layer mode of the gradient below to normal. Your setup should look like figure 20.

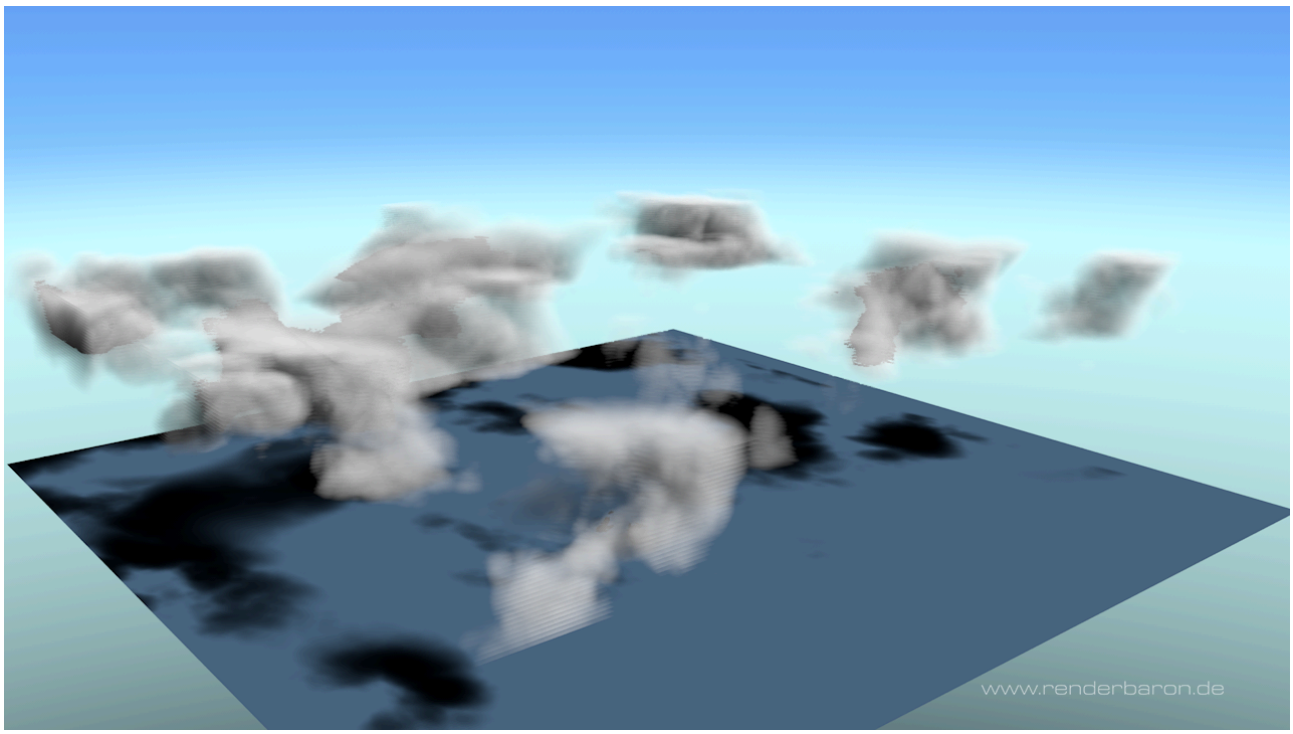


Figure 20: Intermediate setup of the Cloud shader

Now set the Layer-Mode of the Mask-Noise to Levr and see what happens: a kind of high-contrast version of the noise is cut away from the gradient applying more structure and modulation to it. Lower the Opacity of the noise to 70% to get a softer edge, as shown in figure 21. Set the Mode of your gradient to multiply again. Before rendering go to Cloud-Noise and set the Low Clip to 20%. Then render again. Your result should look like figure 22.

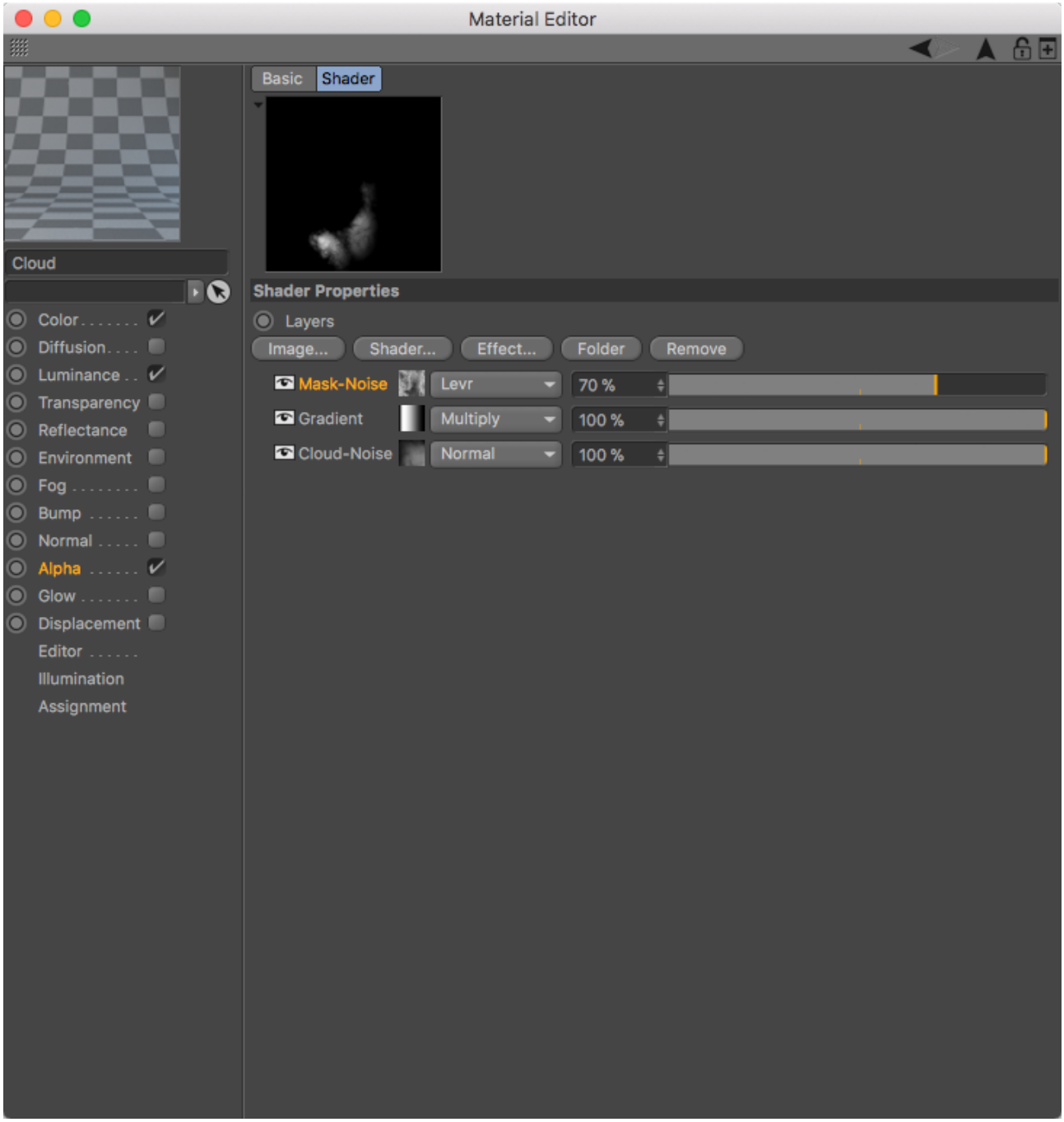


Figure 21: Final setup of the Cloud shader

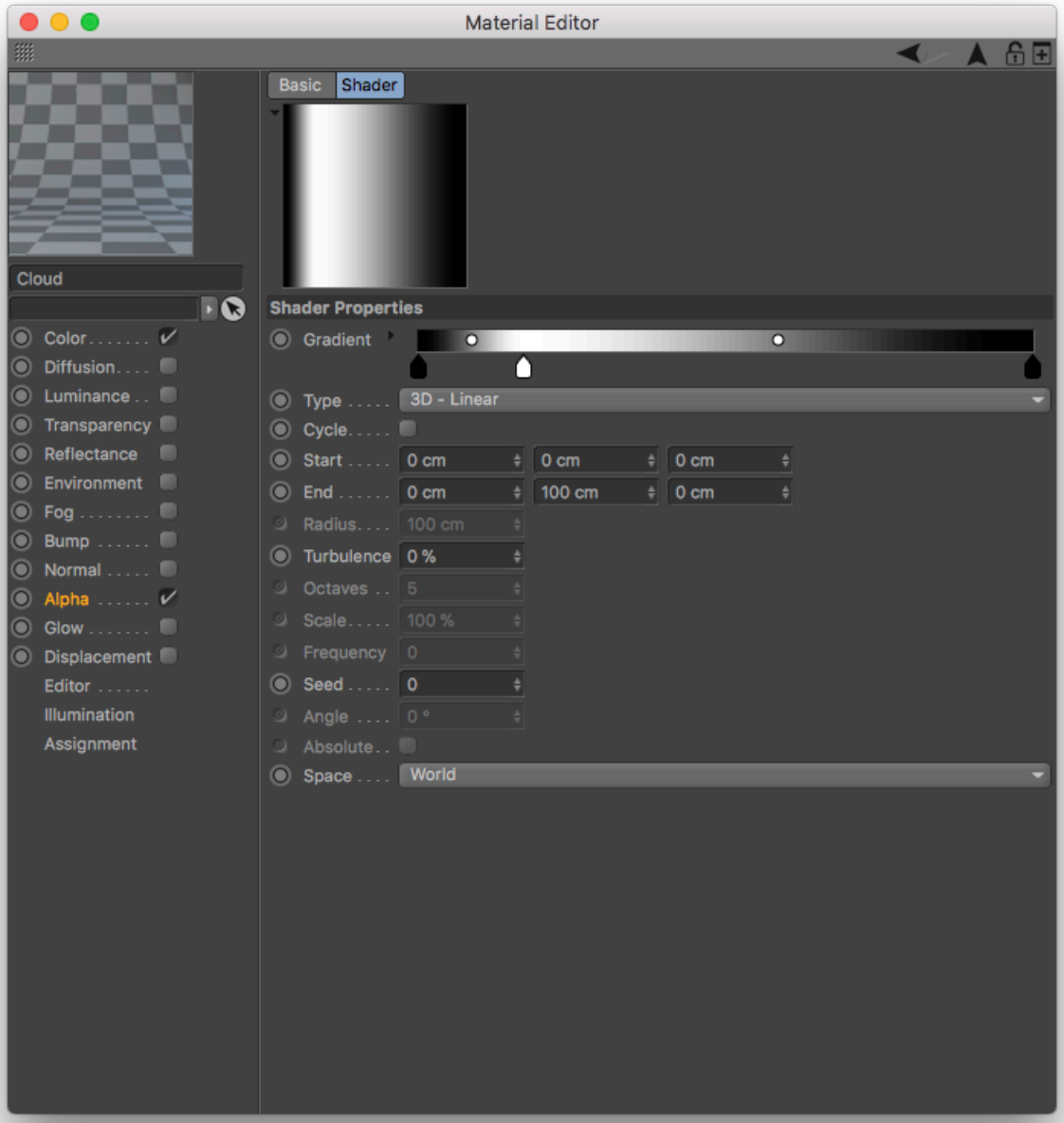


Figure 22: Result of the vertical restriction

Shadow Luminance - Simulate a Kind of Subsurface Scattering

The clouds are getting more and more realistic but are looking kind of grey and dark. This is because every MoGraph slice casts a shadow on the slice below – that's the way the clouds are created. To brighten up shadow areas exclusively use a shader setup to simulate a kind of self-illumination on shadow sides of objects: Shadow Luminance.

While this shader setup is originally intended to simulate very diffuse light on shadow sides of objects you can also use it to simulate a kind of subsurface scattering when used on clouds. Shadow Luminance consists of two important components:

- **Lumas Shader:** detects where there is light and where there is shadow. The shader

tab in Lumas acts like the color channel, by saying “show me where I am in the light”. **Colorizer Shader:** inverts Lumas to “show me where I am in the shadow” and is used to mask a color to the shadow side of an object.

While Shadow Luminance is placed in the Luminance Channel the shader setup works as slight brightening of Shadow areas.

Activate the Luminance Channel of your Cloud Material and create a layer shader. Inside the layer shader create a Lumas Shader. Deactivate all specular aspects of the Lumas Shader. Under Tab Shader choose an Illumination of 100% and adjust the Color to a bright white. Head back to the layer shader and put the Lumas inside a colorizer shader by right-clicking on the Lumas and choosing Colorizer. Inside the Colorizer Shader adjust its gradient from white to black.

Go back to the layer shader, click the shader button and create a color shader. Adjust the color shader to a light blue. Back to the layer shader set the color shaders opacity to 8%. Choose layer mask as the layer mode for the colorizer. Your setup should look like figure 23. You are now masking a light blue to the Shadow sides of your clouds.

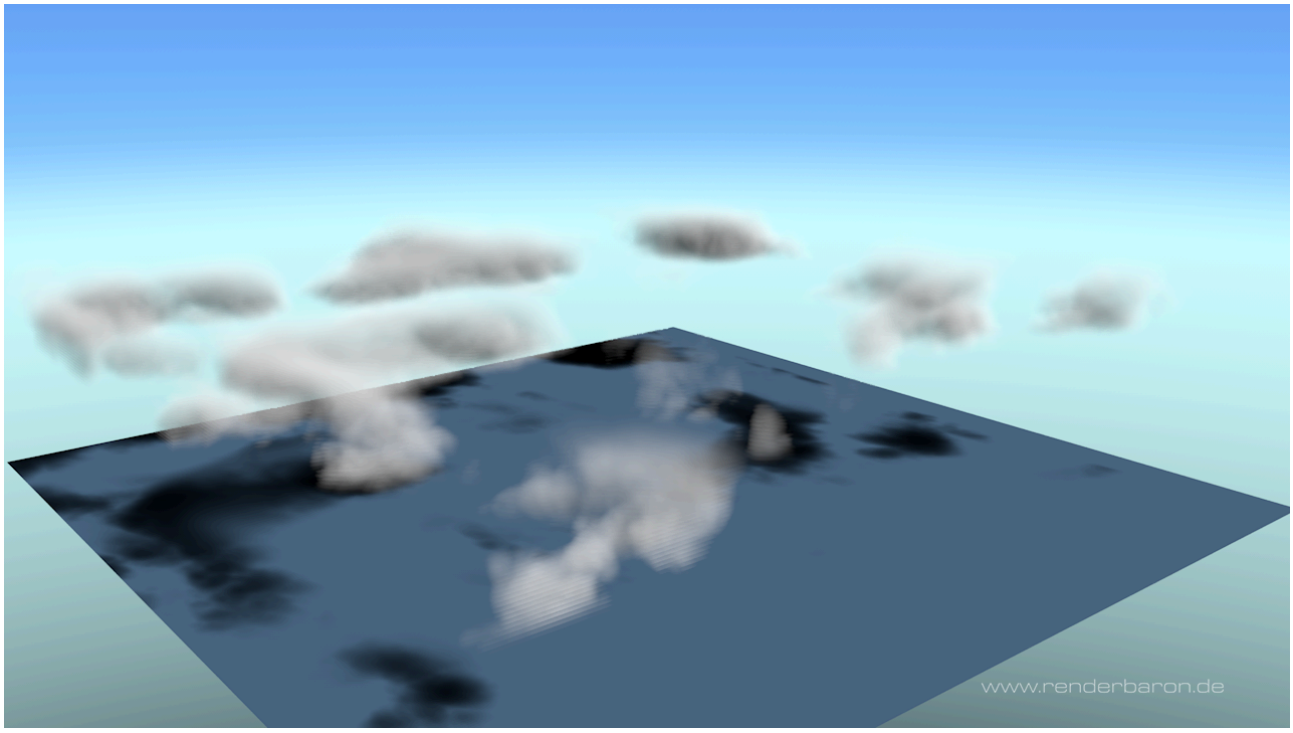


Figure 23: The Shadow Luminance SetUp

Increasing Light Sensitivity

Having a look at the color channel of the clouds material you will see that Illumination is set to Lambert. Lambert is a Bidirectional Scattering Distribution Function (BSDF). Or in simple terms, a function describing how light is distributed across the surface of an object, from its brightest to its darkest point. Lambert is simulating a perfectly diffuse surface while the other available BSDF, Oren Nayar, is additionally calculating micro-facets for a more satinated look.

While Lambert is a good choice for our bright clouds, Oren-Nayar has one advantage: the Diffuse Level and Roughness parameters.
With roughness you can seamlessly blend between Lambert (0% Roughness) and Oren-Nayar (100% Roughness) behavior. And with diffuse level you can adjust the light sensitivity. So with roughness set to 0% and diffuse level at 200% we have a Lambert Illumination with increased light sensitivity and brightness. (Pic 28)

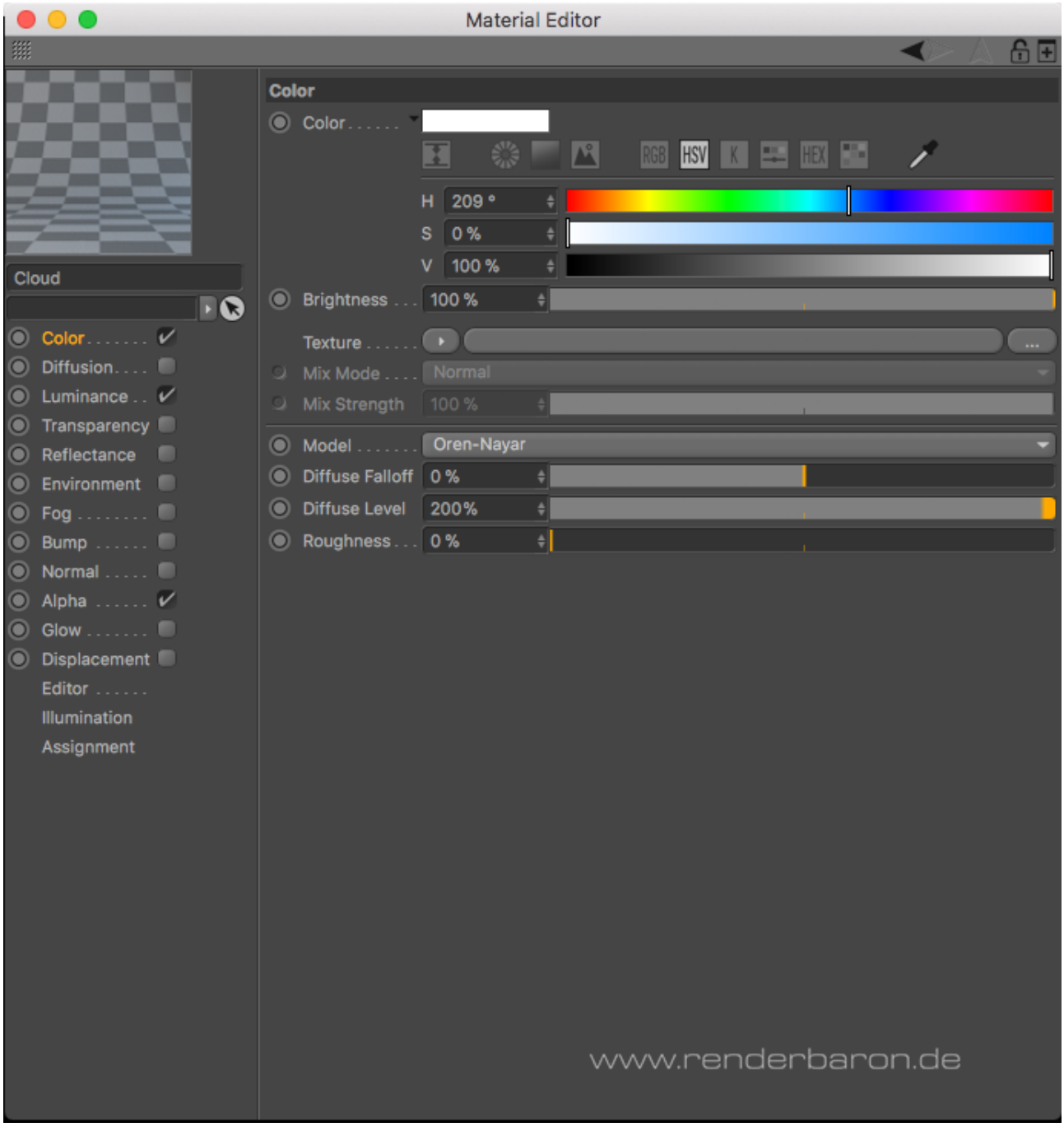


Figure 24: Parameters to increase light sensitivity

With an increased clone count (80), adjusted ray depth, and shadow luminance for the floor material and area shadows for the light source your final result should look like figure 25.

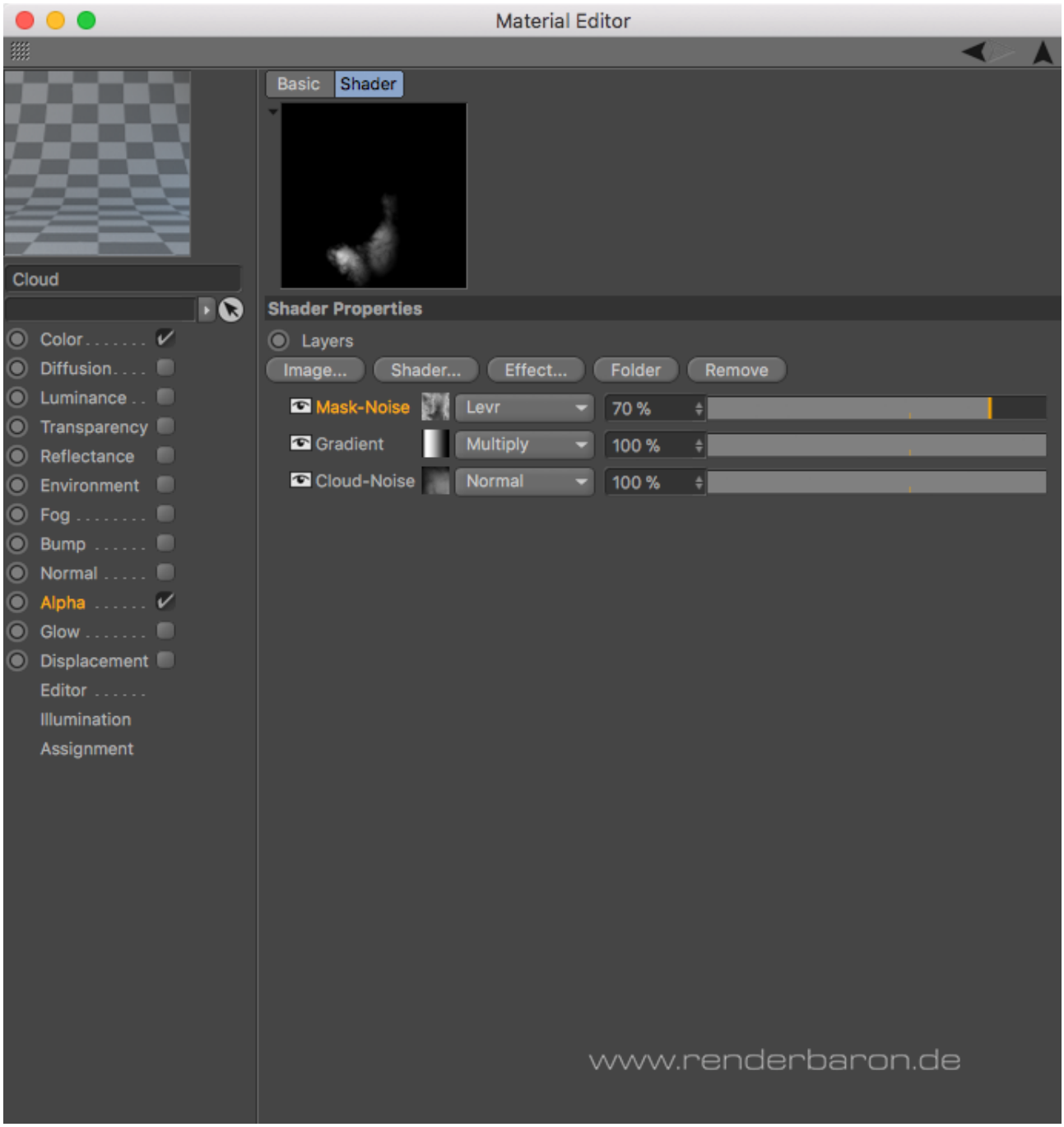


Figure 25: Final result for puffy clouds

Conclusion

Sliced puffy clouds are a matter of Clone count and Ray Depth. The smoother your result, the longer it takes to render. Applying realistic area shadows will cause your render times to increase further, therefore clouds are a balance between a low clone count/ray depth and smooth results .

Physical Render is the best choice in this case as it provides high quality anti-aliasing, faster handling of area shadow samples and Intel Embree acceleration.

Learn more about creating thin clouds, fog and nebulae in Surprising Volumetric Effects in Cinema 4D* Part 1: The Magic of Visible Lights. More detail on Marc and his work can

be found the Intel® Developer Zone, including Mountainvista, SIGGRAPH 2018, and Comparing 3D Rendering Performance Using the Cinema 4D Mountainvista Scene Workload. Also, check out the Cinema 4D Release 20 that was announced during SIGGRAPH 2018. Visit the Intel® Game Dev program to for the latest updates and resources to help you elevate your game development.

Resources
Intel® Core™ i9-7980XE Extreme Edition processor
Cinema 4D

About the Author

Intel® Software Innovator Marc Potocnik is a German designer and founder of animation studio renderbaron (www.renderbaron.de). renderbaron specializes in the production of high-quality 3D animation and visual effects for a wide range of high-profile companies including ZDF, Audi, Siemens, BMW, a.o.

Marc studied communications design at the University of Applied Sciences in Düsseldorf entering the world of 3D with Cinema 4D R4 in 1997. He founded renderbaron in 2001 and went on to become an authority in shading, lighting and rendering with Cinema 4D. Marc is qualified as a Maxon* Lead Instructor and has correspondingly written the Maxon QuickStart Training: Shading, Lighting & Rendering. He also regularly shares his knowledge at conferences around the world including SIGGRAPH, IBC and FMX. Follow along with renderbaron projects on Facebook.