appleseed-maya

Release 421e4d4a3297b2ffd3d9f4488f035c655df613a6

Sep 09, 2019

Contents

Documentation generated on 2019-09-09 at 02:25

CHAPTER 1

Contents

1.1 Features

appleseed-maya is an appleseed plugin for Autodesk® Maya®.

1.1.1 Main exporter features

Presently the exporter supports:

- Rendering to Maya's RenderView
- Batch Rendering
- Appleseed project export (*.appleseed files) or packed appleseed project files (*.appleseedz)
- Image Based Lighting via appleseed Physical Sky, and Dome Light
- Support for most of Maya's hypershade nodes (see *Supported HyperShade Nodes*)
- Several appleseed specific materials, texture, and utility nodes. See *Custom appleseed shaders* for detailed information
- Shader overrides with diagnostic modes, ambient occlusion geometric information
- Final render denoising *[BB17]*
- Support for custom OSL shaders as Maya shading nodes
- Support for OpenColorIO (or OCIO)
- Partial support[1] for Maya's SynColor[2] color management system.

[1] For now, the input device transforms supported for the ingested material are listed in *asTexture Color Management options*.

[2] Maya's native color management system, providing equivalent functionality to OpenColorIO. Not all input transformations are supported. See Autodesk Color Management Supplemental Information for more details.

1.1.2 Limitations

Presently appleseed has the following limitations:

- Swatch rendering is limited to material nodes, not texture node
- IPR is not supported yet
- XGen support is work-in-progress and is disabled in this release
- Only meshes are exported and rendered right now
- PTex is not supported yet
- The rendering or working space is limited to Rec.709/sRGB RGB primaries with a D65 whitepoint

See also:

appleseedhq GSoC 2018 wiki if you want to assist implementing any of these features.

References

1.2 Installation

The plugin is a Maya module.

1. Uncompress the zip file.

2. Add the path containing the `appleseed-maya.mod` file to your `Maya.env`[1] file or to the `MAYA_MODULE_PATH` environment variable.

3. Launch Maya and enable the plugin in the Plug-in Manager.

1.3 Supported HyperShade Nodes

Support status for the Hypershade nodes, and some extra plugin nodes. If the node support status is empty, it means that the node might or might not be supported at a later date.

1.3.1 Utilities

Nodes	Support Status
Add Double Linear	Supported
Add Matrix	Supported
Angle Between	

Continued on next page

[1] See Setting environment variables using Maya.env.

Table 1.1 – continued from previous page

Nodes	Support Status
Array Mapper	
Blend Colors	Supported
Blend Two Attr	
Bump2d	Supported
Bump3d	Supported
Choice	
Chooser	
Clamp	Supported
Clearcoat	Supported
Color Profile	Supported (**WIP**)
Compose Matrix	
Condition	Supported
Curve Info	
Decompose Matrix	
Distance Between	Supported
Double Switch	Supported (**WIP**)
Euler to Quat	
Four by Four Matrix	Supported
Frame Cache	**Unsupported**
Gamma Correct	Supported
Height Field	**Unsupported**
HSV to RGB	Supported
Inverse Matrix	Supported
Light Info	
Luminance	Supported
Mult Double Linear	Supported
Mult Matrix	Supported
Multiply Divide	Supported
Particle Sample	
2D Placement	Supported
3D Placement	Supported
Plus Minus Average	Supported
Projection	**Unsupported**[1]
Quad Switch	Supported (**WIP**)
quatNodes Add	
quatNodes Conjugate	
quatNodes Invert	
quatNodes Negate	
quatNodes Normalize	
quatNodes Prod	
quatNodes Sub	
quatNodes to Euler	
Remap Color	Supported
Remap HSV	Supported
Remap Value	Supported
Reverse	Supported
RGB to HSV	Supported
Sampler Info	Supported
SetRange	Supported

Continued on next page

Table 1.1 – continued from previous page

Nodes	Support Status
SingleSwitch	Supported (**WIP**)
Stencil	Supported
Surface Info	
Surf.Luminance	**Unsupported**
Transpose Matrix	Supported
Triple Switch	Supported (**WIP**)
Unit Conversion	
UV Chooser	
Vector Product	Supported
Weighted Add Matrix	Supported

1.3.2 Image Planes

Nodes	Support Status
Image Plane	

1.3.3 Surface

Nodes	Support Status
GLSL Shader	**Unsupported**
ShaderFX shader	**Unsupported**
StingRayPBS	**Unsupported**
Anisotropic	Supported
Bifrost Aero	
Bifrost Foam	
Bifrost Liquid	
Blinn	Supported
CgFX Shader	**Unsupported**
Hair Tube Shader	
Lambert	Supported
Layered Shader	**Unsupported**[2]
Ocean Shader	
Phong	Supported
PhongE	Supported
Ramp Shader	
Shading Map	**Unsupported**
Surface Shader	
Use Background	

[1] An alternative to Maya's projection node will come on the form of a custom appleseed shader.

[2] An alternative appleseed node is provided, *asBlendShader* for the layering of shaders, and two shaders are provided to blend, and to composite textures. Namely *asBlendColor* and *asCompositeColor*. These provide functionality that greatly exceeds this node's capabilities.

1.3.4 Volumetric

Nodes	Support Status
Env Fog	
Fluid Shape	
Light Fog	
Particle Cloud	
Volume Fog	
Volume Shader	

1.3.5 Displacement

Nodes	Support Status
Displacement	
C Muscle shader	

1.3.6 2D Textures

Nodes	Support Status
Bulge	Supported
Checker	Supported
Cloth	Supported
File	Supported
Fluid Texture 2D	
Fractal	Supported
Grid	Supported
Mandelbrot	Supported[3]
Mountain	Supported
Movie	Supported[4]
Noise	Supported
Ocean	**WIP**
PSD File	Supported
Ramp	Supported
Substance	Supported[5]
Substance Output	Unsupported[6]
Water	Supported

---

[3] The *Mandelbrot2D* node is partially supported only.

[4] The *Movie* node requires OIIO built with FFMPEG. Reliable access to individual frames of a video file, or *frame-by-frame scrubbing* works as expected with I-frames only videos.

[5] In order to use Substance materials, you need to use the Maya's *Substance* node image file output. This will be covered on its own tutorial.

[6] For now, The *Substance Output* is unsupported. This might change in the future.

1.3.7 3D Textures

Nodes	Support Status
Brownian	Supported
Cloud	Supported
Crater	Supported
Fluid Texture 3D	
Granite	Supported
Leather	**WIP**
Mandelbrot 3D	
Marble	Supported
Rock	Supported
Snow	Supported
Solid Fractal	Supported
Stucco	Supported
Volume Noise	Supported
Wood	Supported

1.3.8 Environment Textures

Nodes	Support Status
Env Ball	
Env Chrome	Supported
Env Cube	
Env Sky	
Env Sphere	

1.3.9 Other Textures

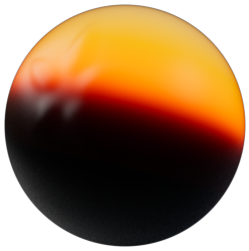
Nodes	Support Status
Layered Texture	Supported

1.3.10 Lookdevkit Nodes

Nodes	Support Status
Simplex Noise	**WIP**

1.3.11 Lights

Nodes	Support Status
Ambient Light	**Unsupported**
Area Light	Supported
Directional Light	Supported
Point Light	Supported
Spot Light	Supported
Volume Light	



1.4 asBlackbody

A blackbody radiator material *[WW11]* shader, with an optional glossy specular term on top, which also outputs the black body radiator color besides the output closure.

1.4.1 Parameters

Incandescence Parameters

Incandescence Type The type of color for the emitance distribution function, it can be

1. Constant
2. Blackbody

When in *Blackbody* mode it uses physically based values, that might have extremely high intensities as the temperature increases.

Color The color to use when in *Constant* mode, ignored in *Blackbody* mode.

Incandescence Amount The overall intensity of the incandescence, it can be over 1.0.

Temperature Scale A scaling factor for the temperature value used to compute the black body radiation. Unlike *incandescence amount*, this actually affects the output color, with lower values generating warmer colors and lower energy levels, and higher values generating cooler values with higher energy levels.

Temperature The temperature to which the perfect black body is heated to emit electro-magnetic radiation.

Normalize Area When objects are deformed, their surface area might change. Without this option, the intensities would stay the same regardless of the deformation of the object to which the shader is attached. It this option enabled, the intensities will change taking into account the object surface area, in order to keep the amount of emitted energy constant.

Tonemap Color As temperature increases, the black body radiator emits color in bluer wavelenghts, but the amount of energy emitted is extreme. In order to avoid extremely hight intensities, this option allows the user to *tonemap* the resulting black body color.

Specular Parameters

Specular Amount The intensity of the specular term.

Specular Roughness The apparent surface roughness of the specular term.

Index of Refraction Absolute index of refraction

Note: This term is a simple dielectric specular term using the *GGX [HDEon14]* microfacet distribution function, with energy compensation for high roughness values.

Bump Parameters

Bump Normal The unit length world space normal of the bumped surface.

Matte Opacity Parameters

Enable Matte Opacity Parameter that toggles matte holdouts.

Matte Opacity Matte opacity scaling factor.

Matte Opacity Color Holdout color.

Advanced Parameters

Ray Depth The maximum ray depth a ray is allowed to bounce before being terminated.

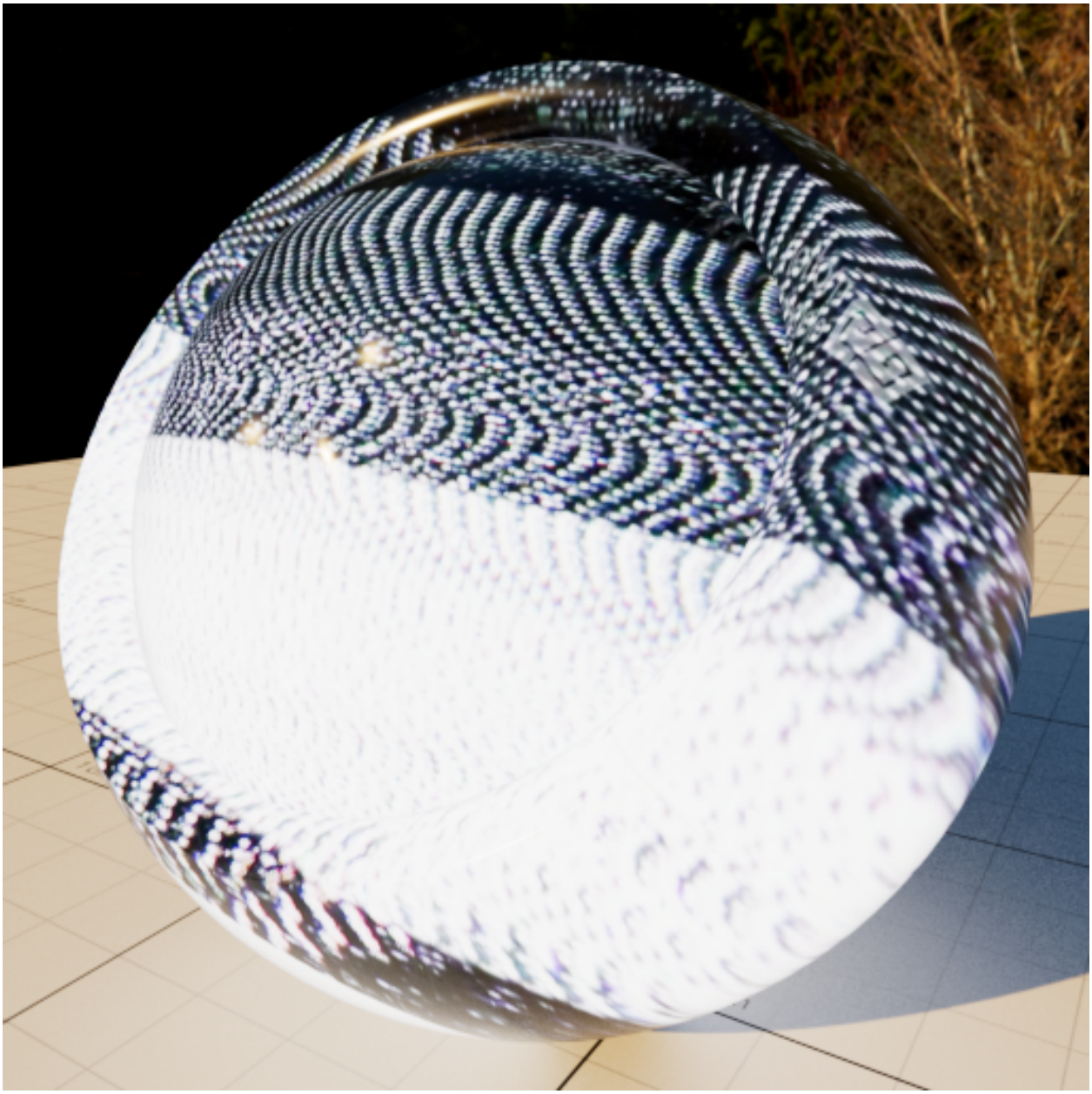
1.4.2 Outputs

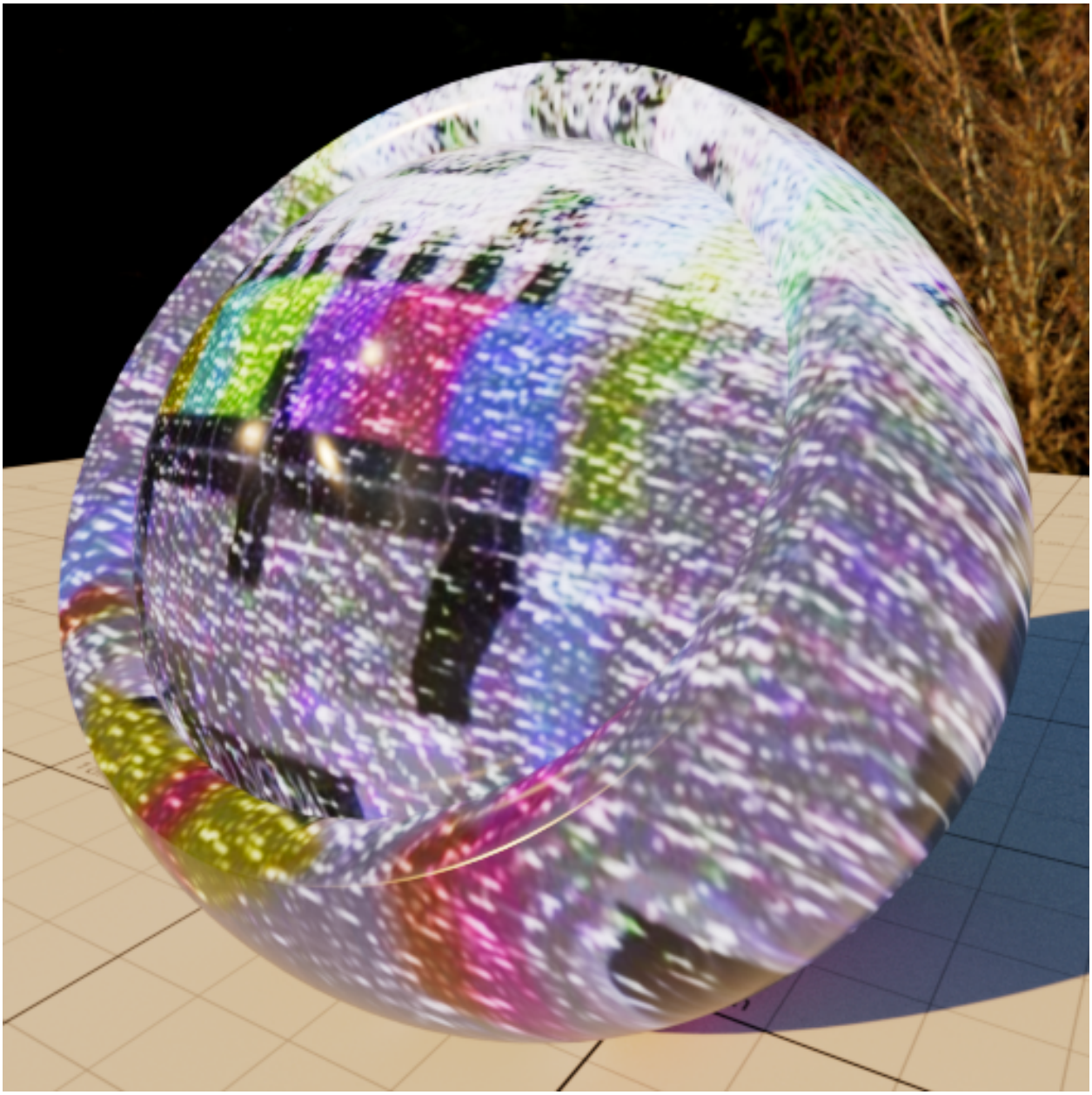
Output Color The combined EDF+BRDF output color.

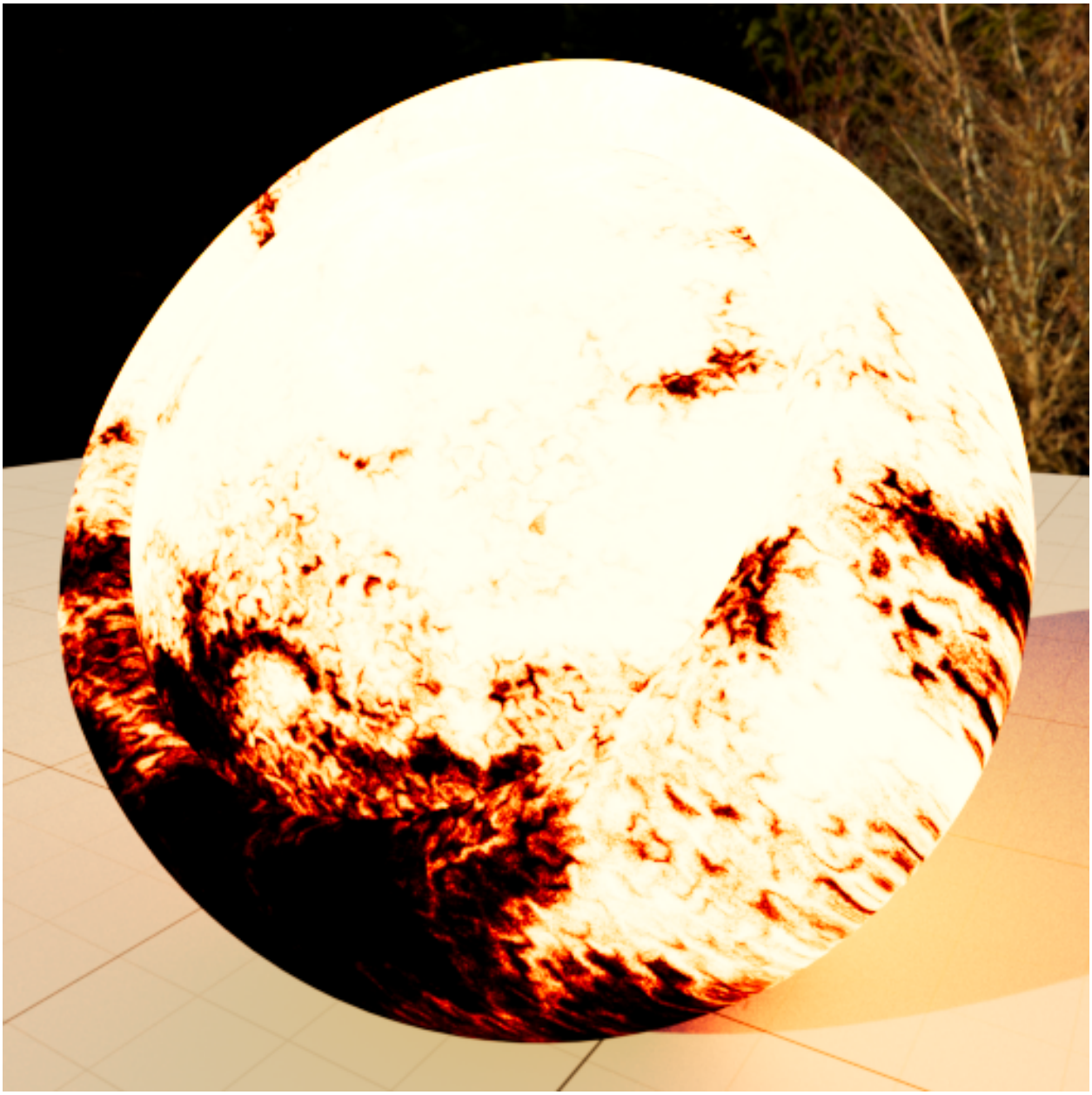
Output Matte Opacity The matte holdout.

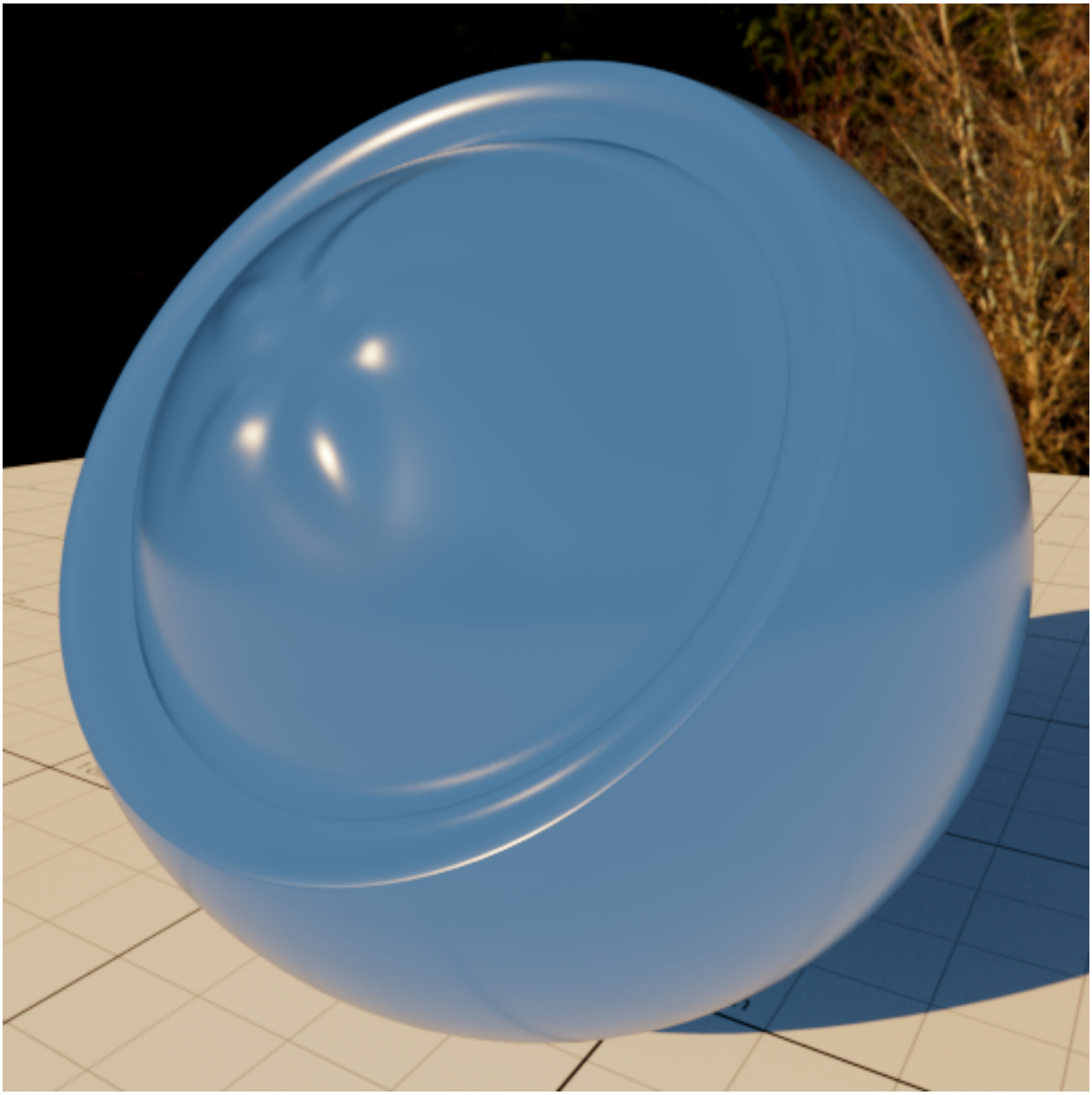
Output Blackbody Color The black body radiator color.

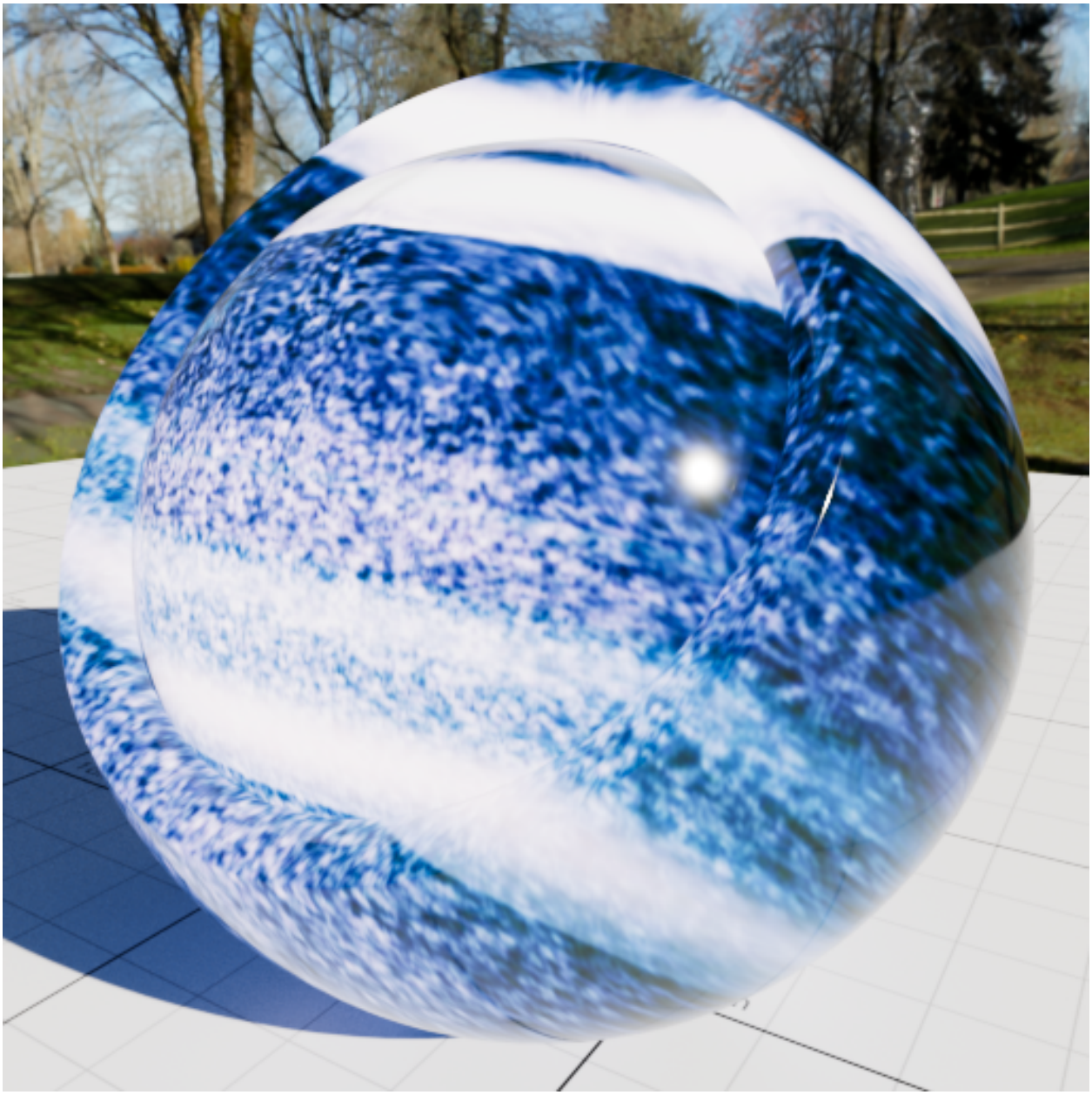
1.4.3 Screenshots

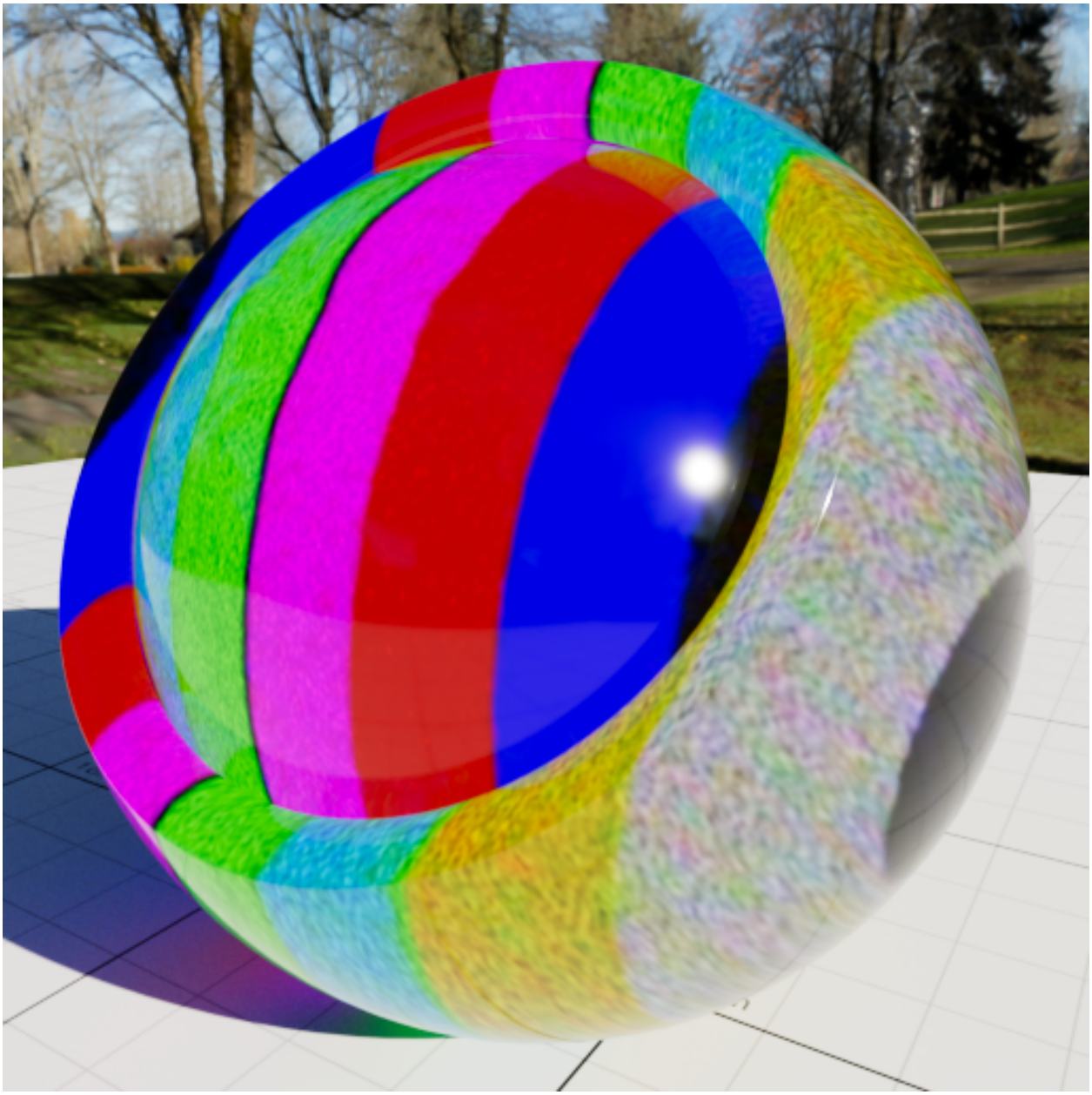


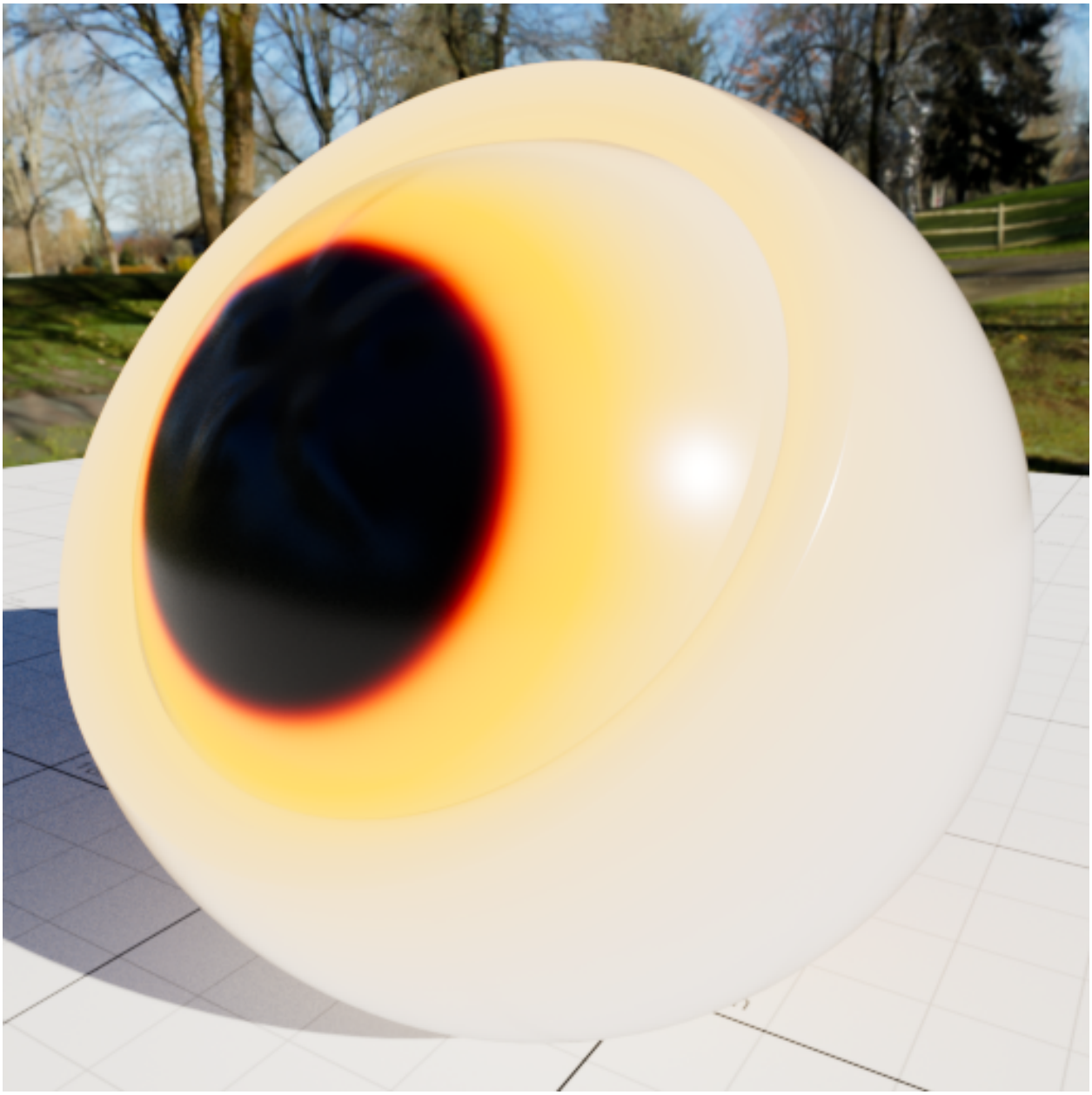












References

1.5 asBlendShader

A shader node allowing the user to blend up to 8 BxDFs[1]. Unlike with physically correct BxDF stacks or layers *[22]* , this node just blends the inputs according to the input mixing weights. There is no interaction in terms of light transport between the top and bottom shaders.

1.5.1 Parameters

Blend Parameters

Layer 0 The first layer shader, from bottom to top order, with layer 0 at the bottom, and layer 7 at the top.

Layer 0 Weight The first layer weight.[2]

Layer 0 Visibility A flag that toggles the respective layer contribution on or off.

(. . .)

Note: The subsequent layers follow exactly the same structure and parameterization.

Layer 7 The last layer shader, the layer at the top.

Layer 7 Weight The last layer weight.

Layer 7 Visibility The last layer visibility flag. Toggling the contribution of this layer on or off.

Warning: This node does **not** layer colors, that is, it is meant to layer BxDF, EDF or other *closures*, not ordinary colors. Without a valid connection[3] to the input, nothing will be layered in the respective layer.

[1] See BSDF definition.

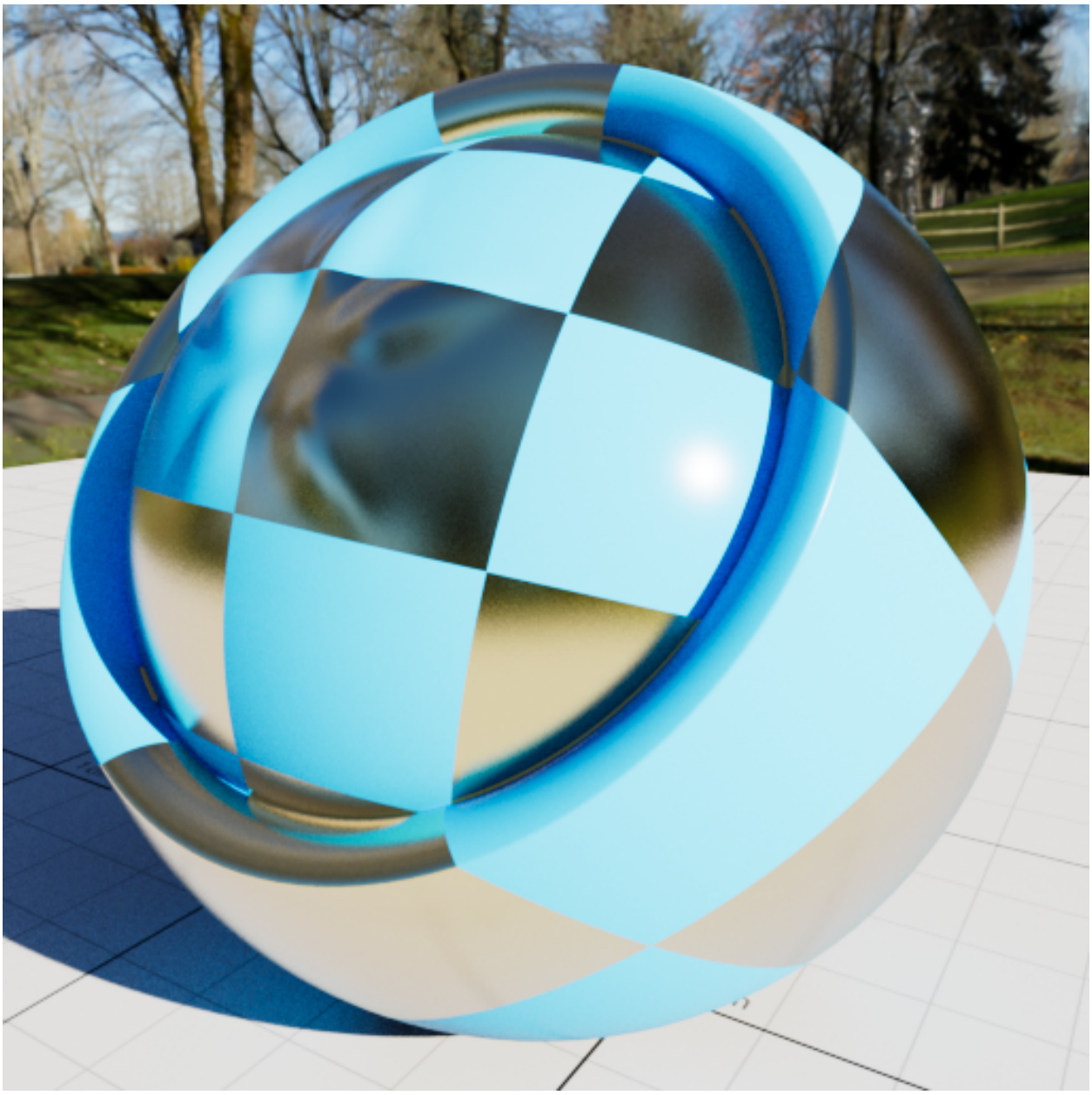
[2] The *layering weight* on the first layer, *Layer 0*, will make a blend between 0 (black), and the *Layer 0 Color*.

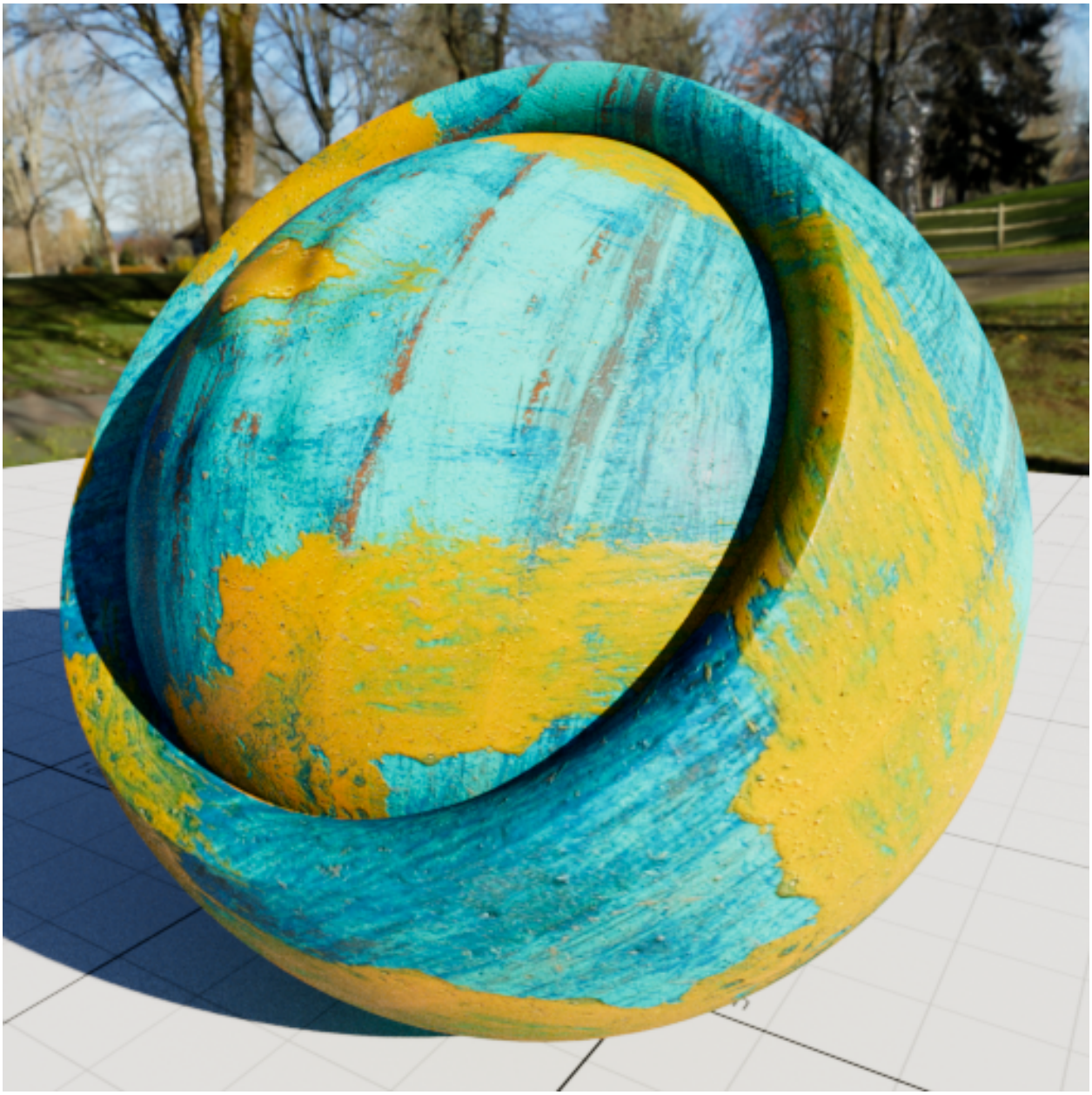
[3] A valid connection would be any *closure*, so any material shader, such as Maya's Blinn, or appleseed's asSubsurface, asMetal, just to name a few. Color nodes such as Noise2D, simple selected colors, and so on, are **not** valid choices. To blend or composite colors, use the *asBlendColor* or *asCompositeColor* nodes instead.

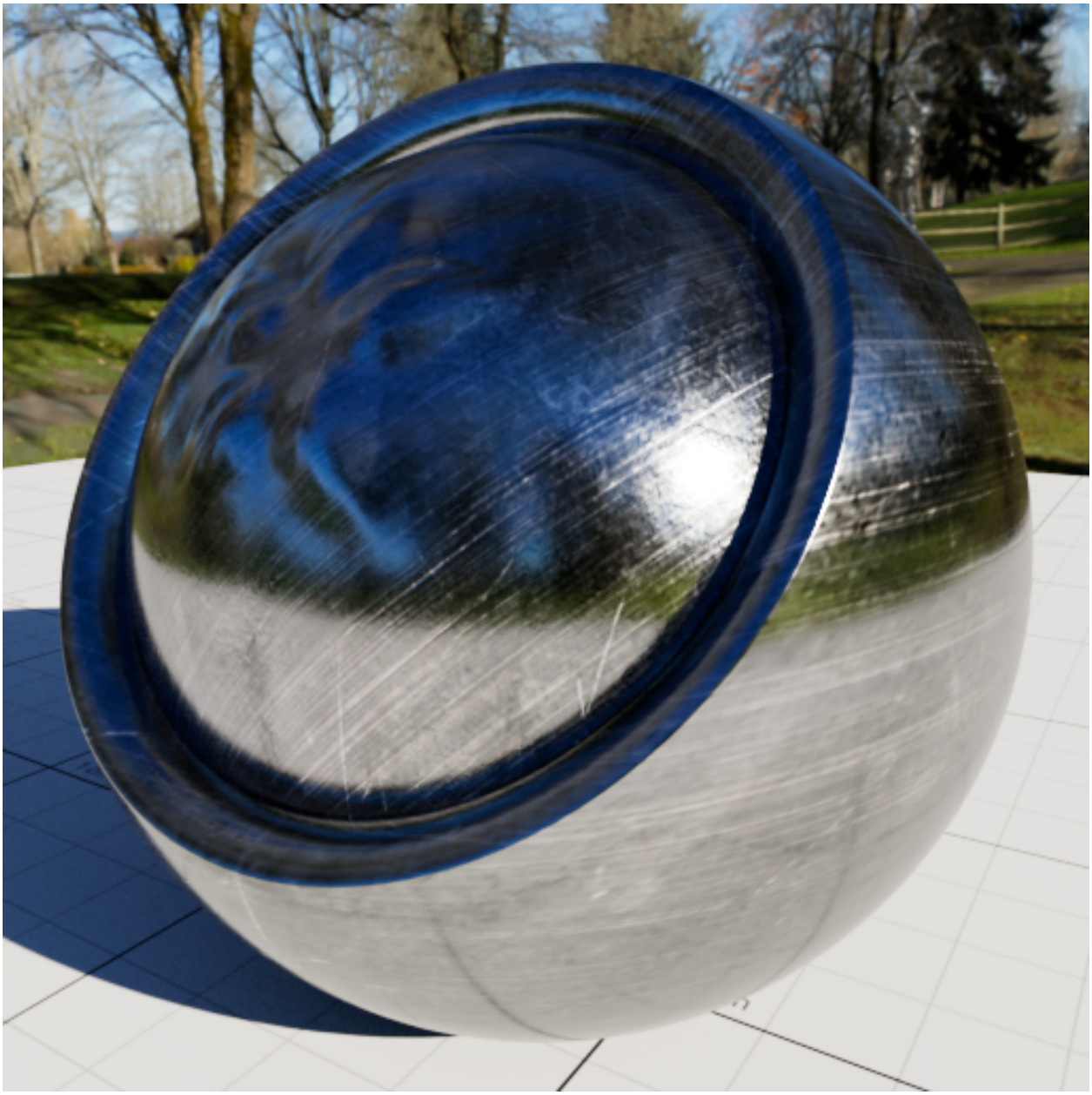
1.5.2 Outputs

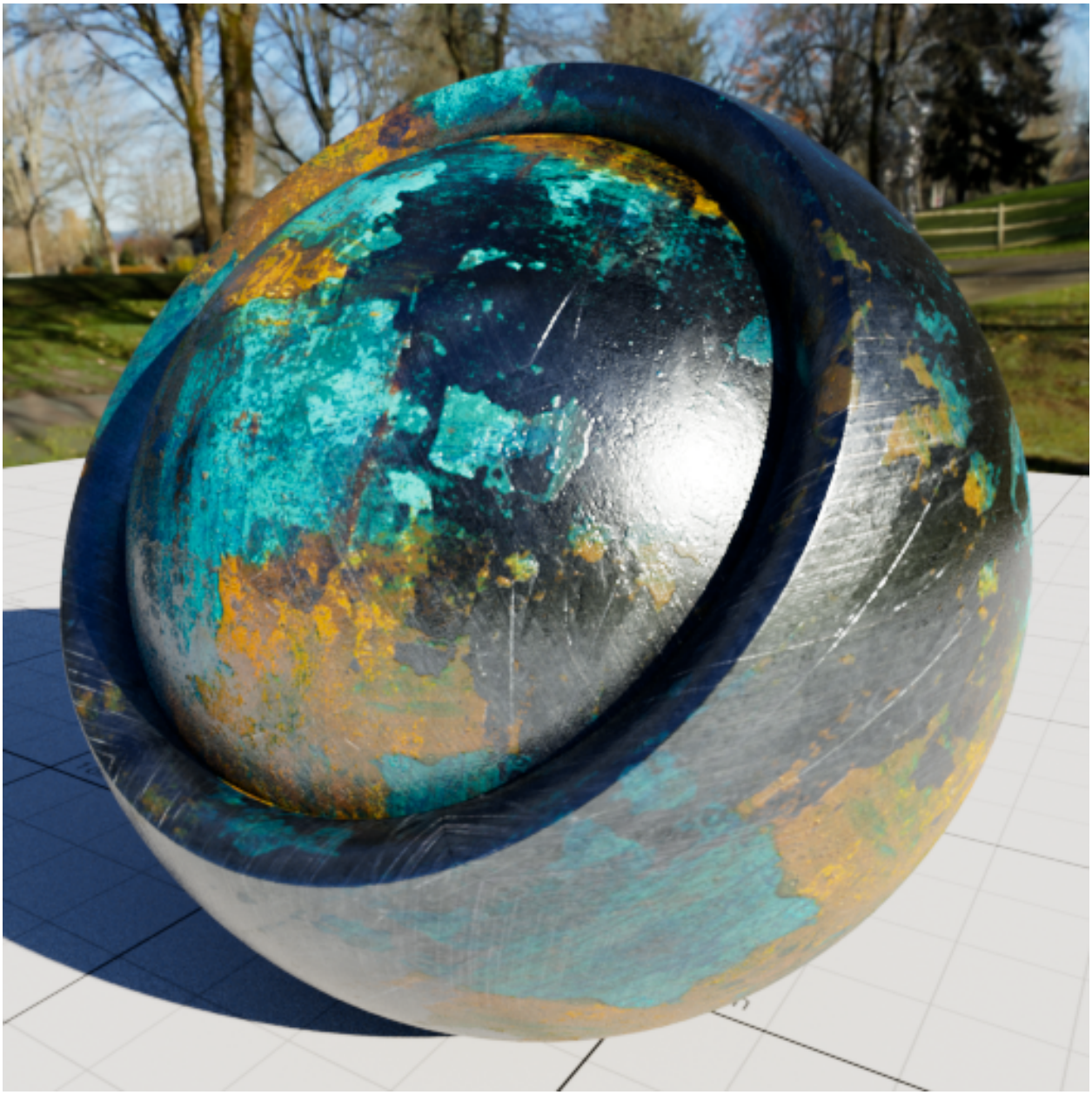
Output Color The combined layers output.

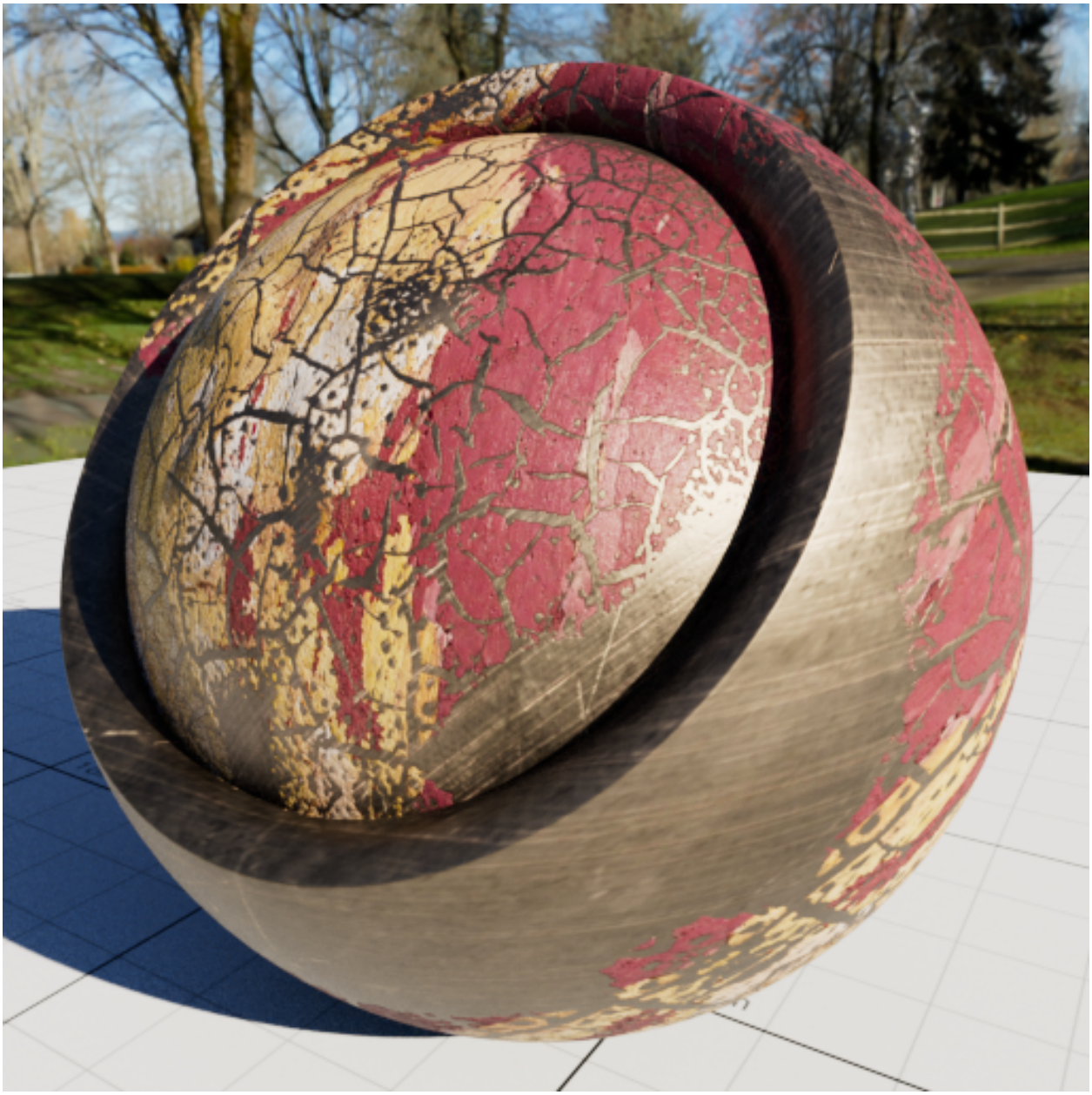
1.5.3 Screenshots

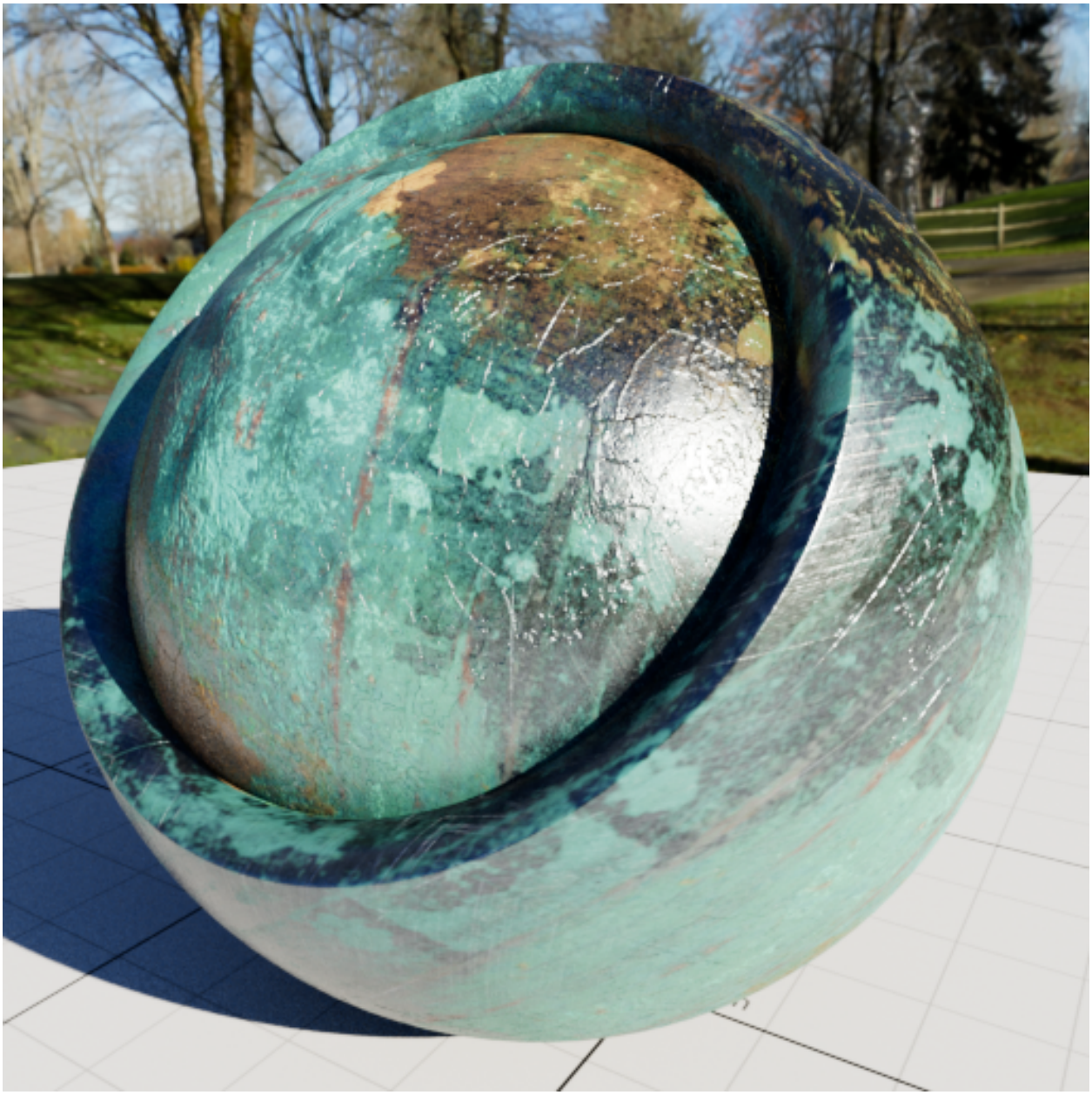


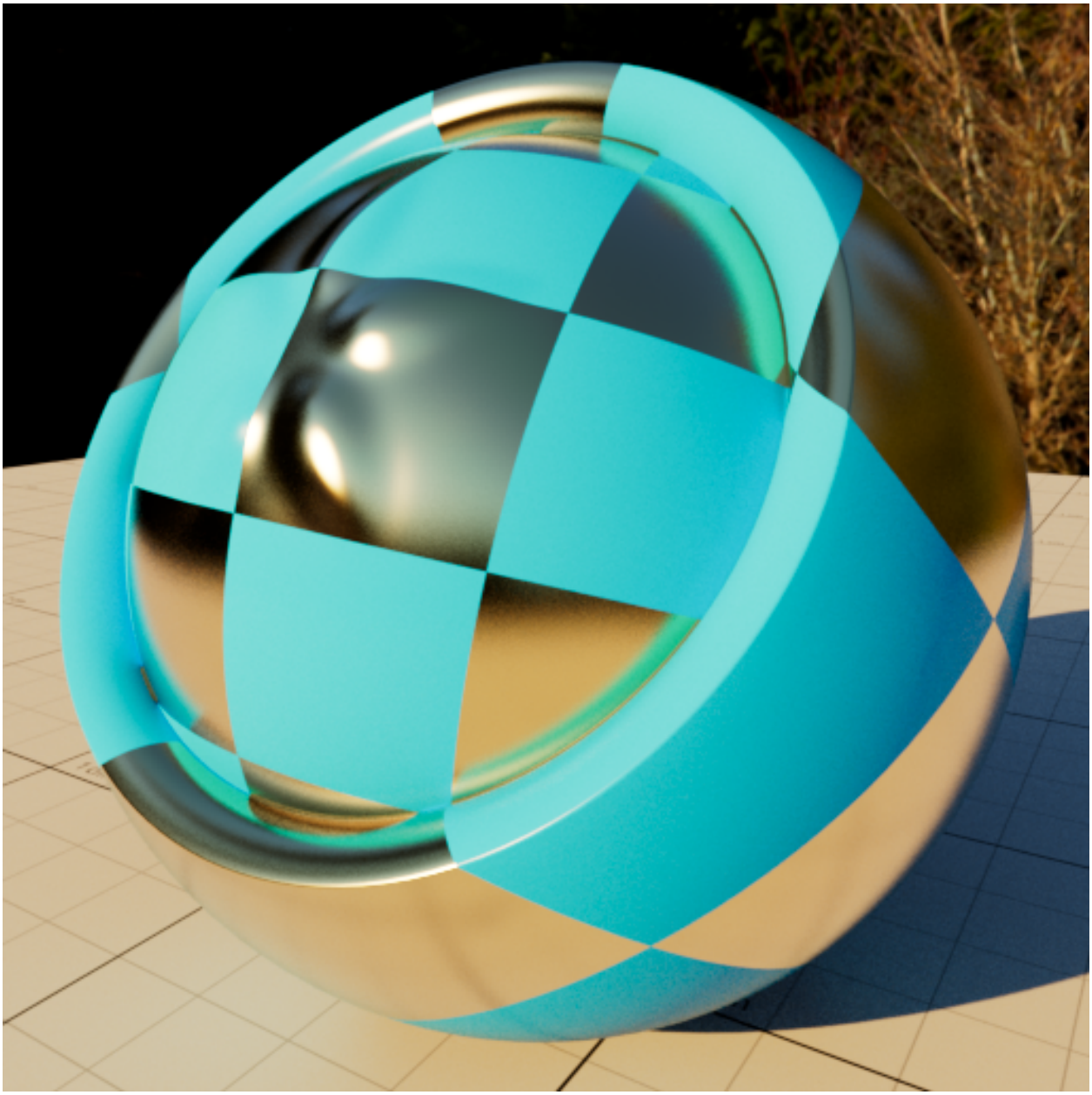


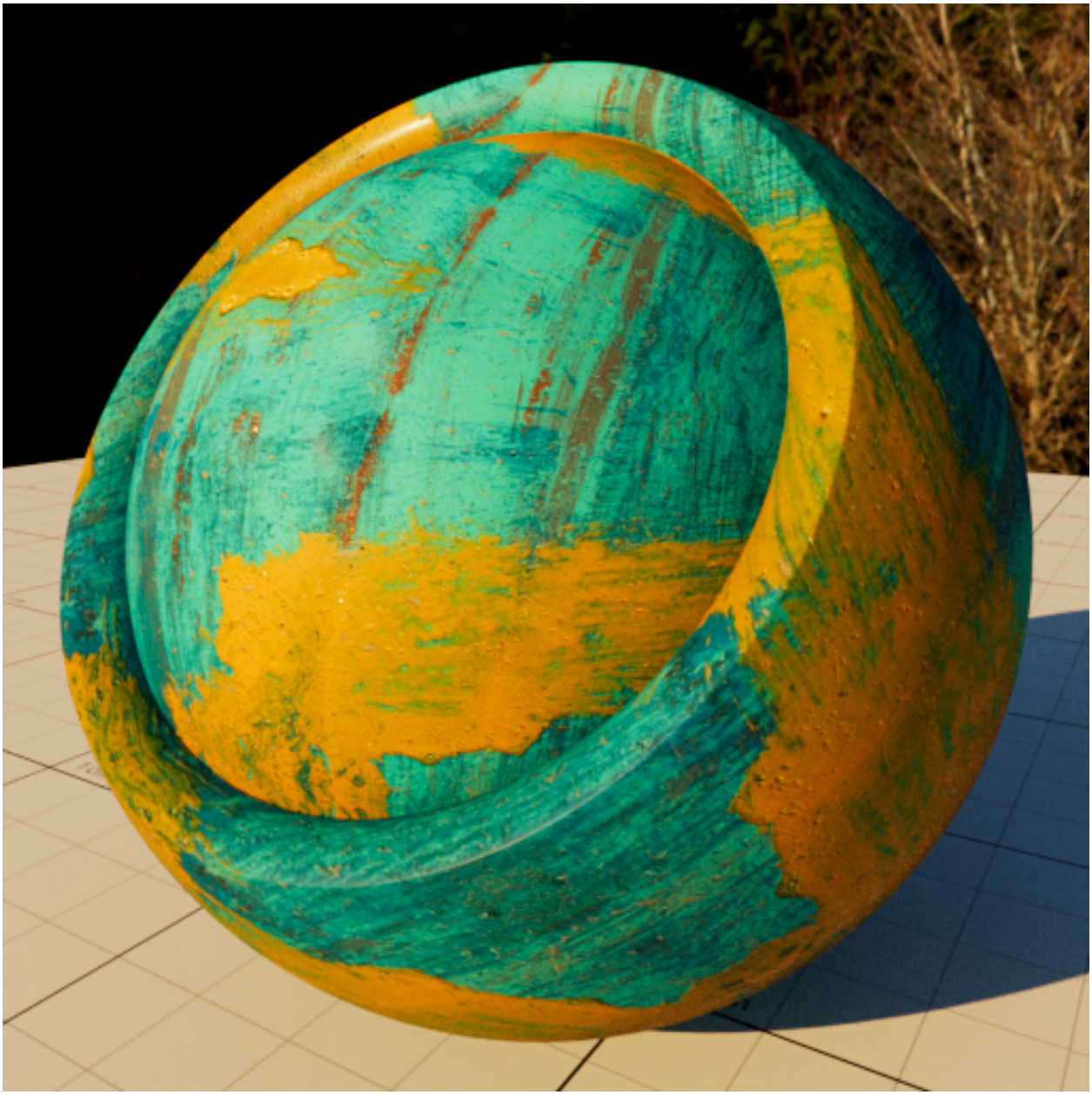


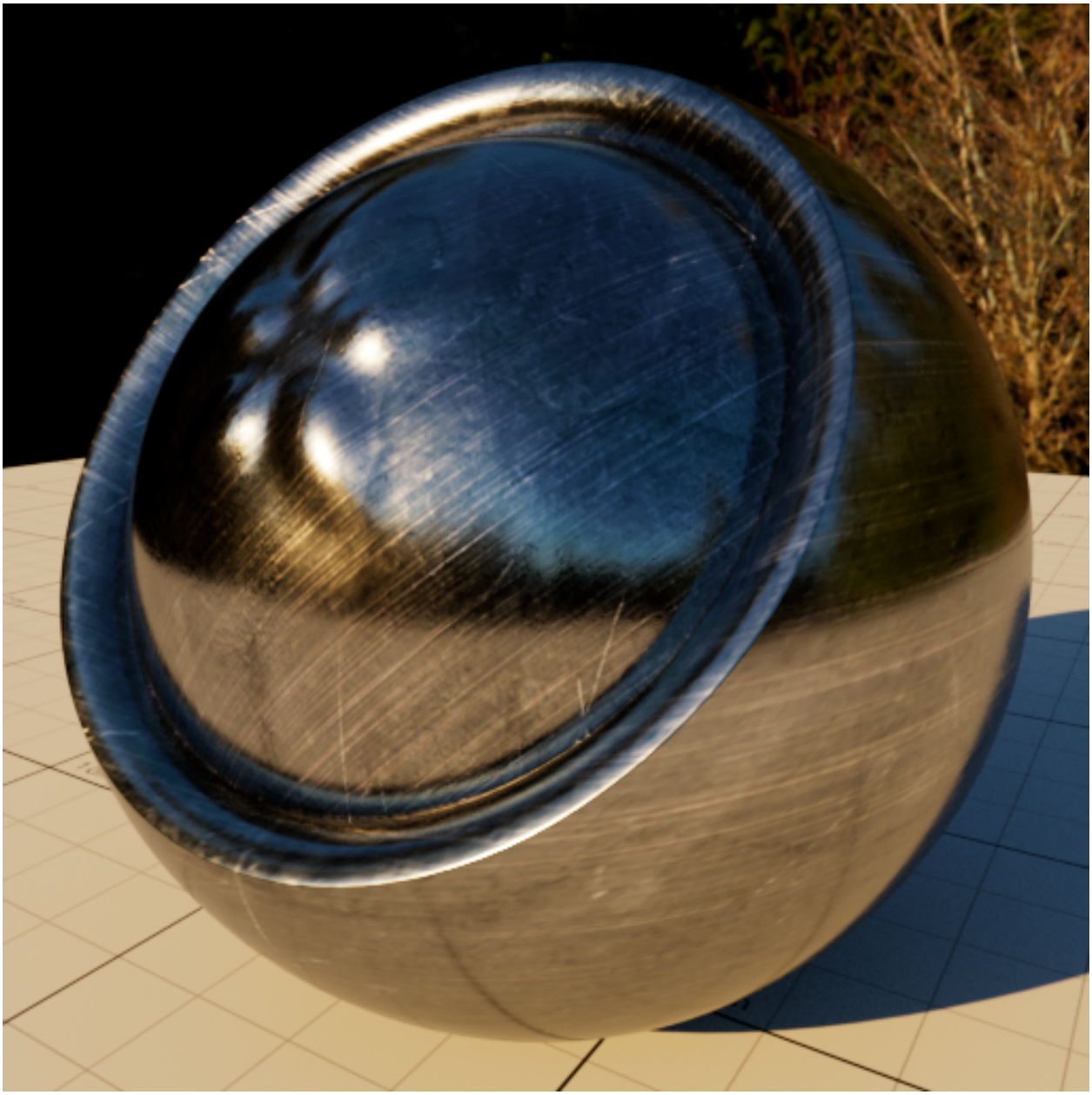


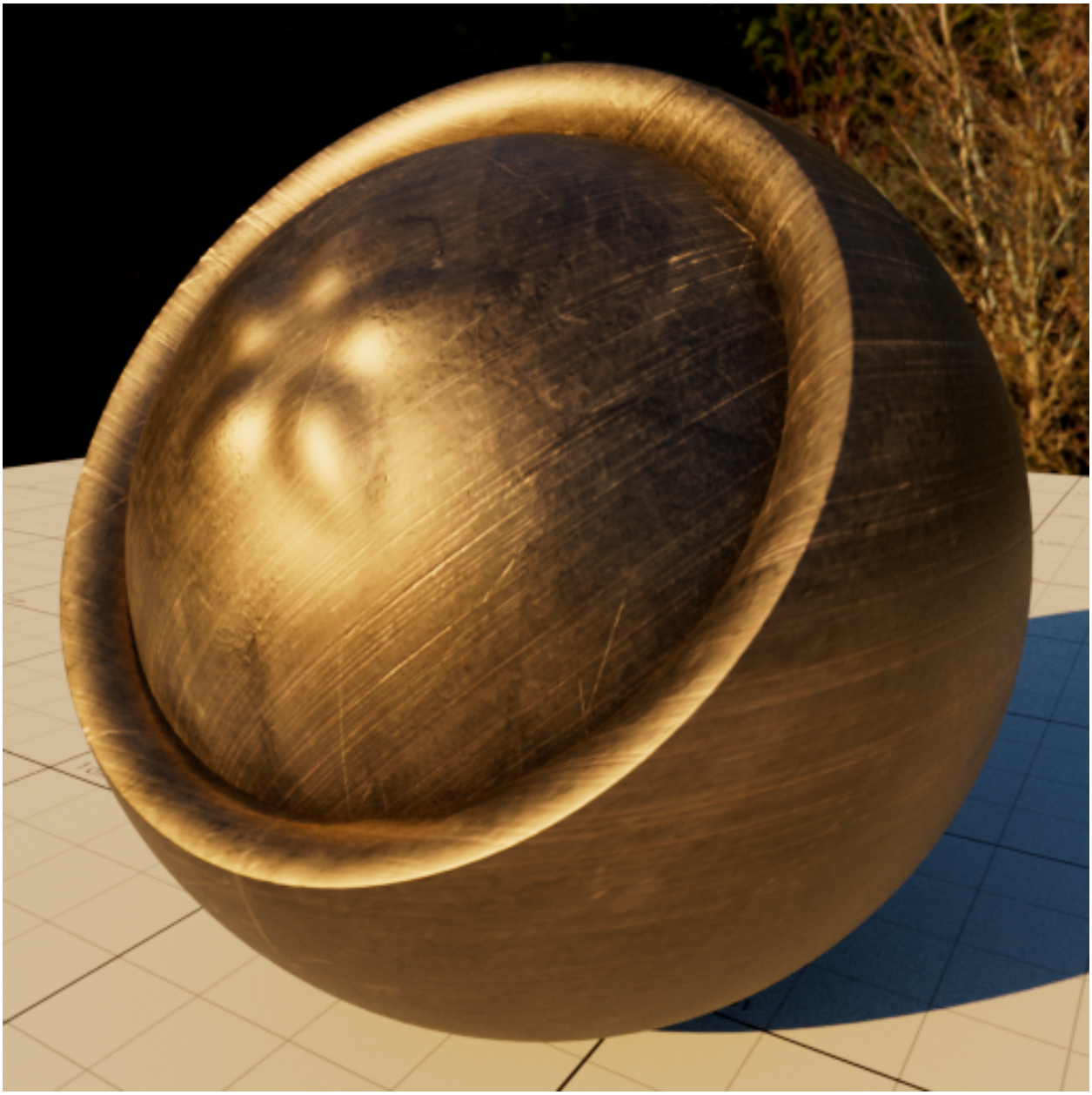


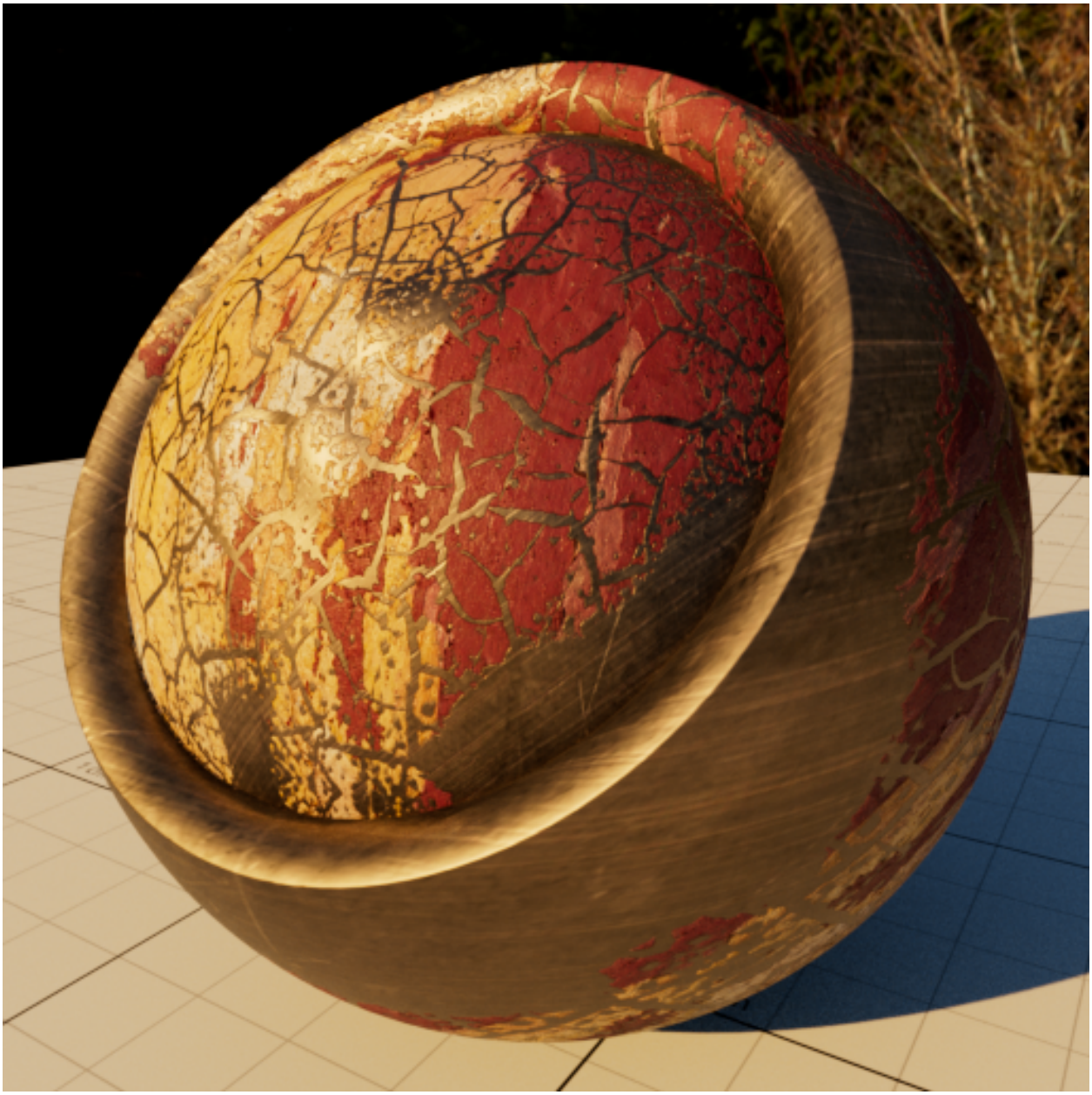


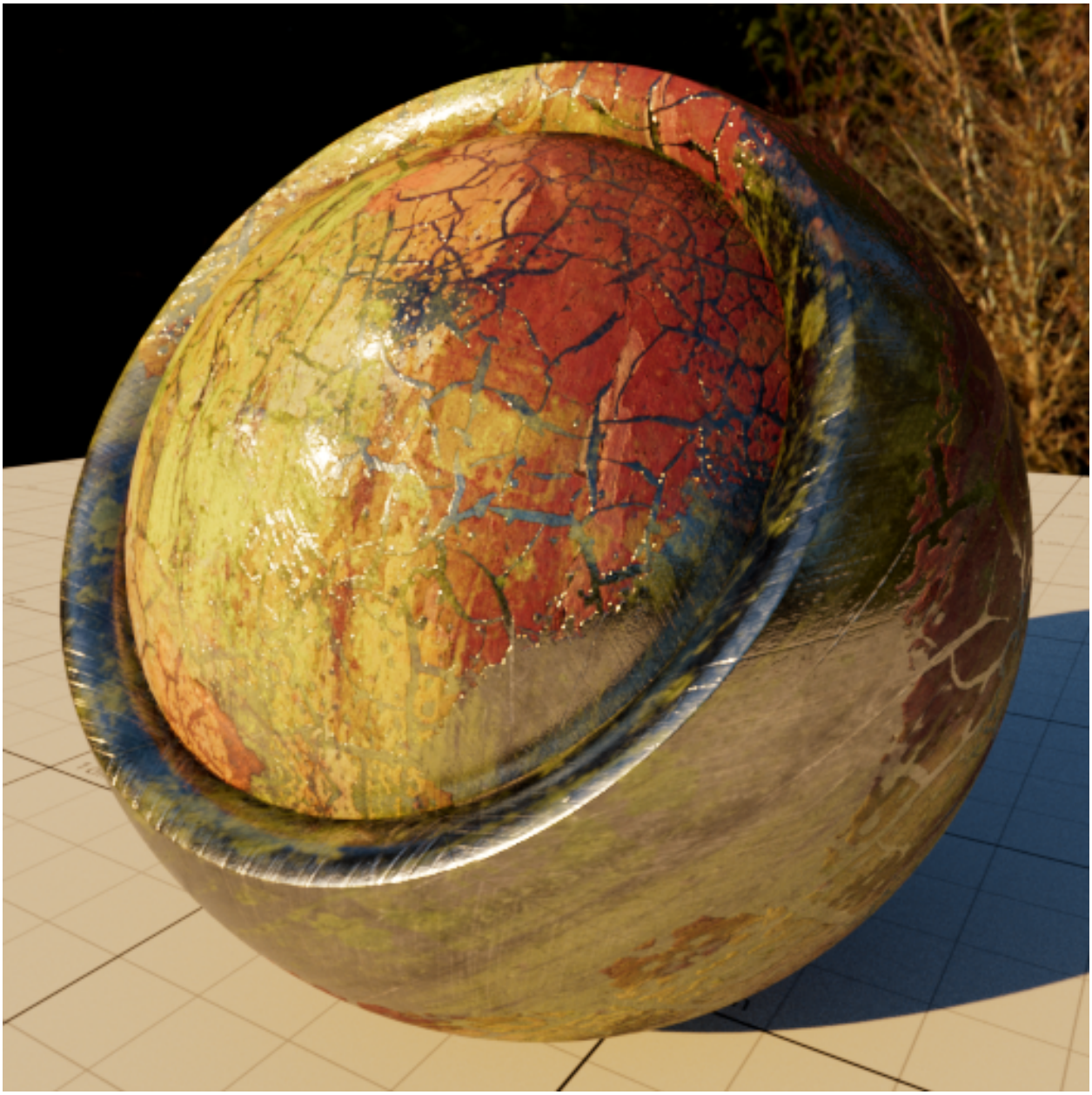


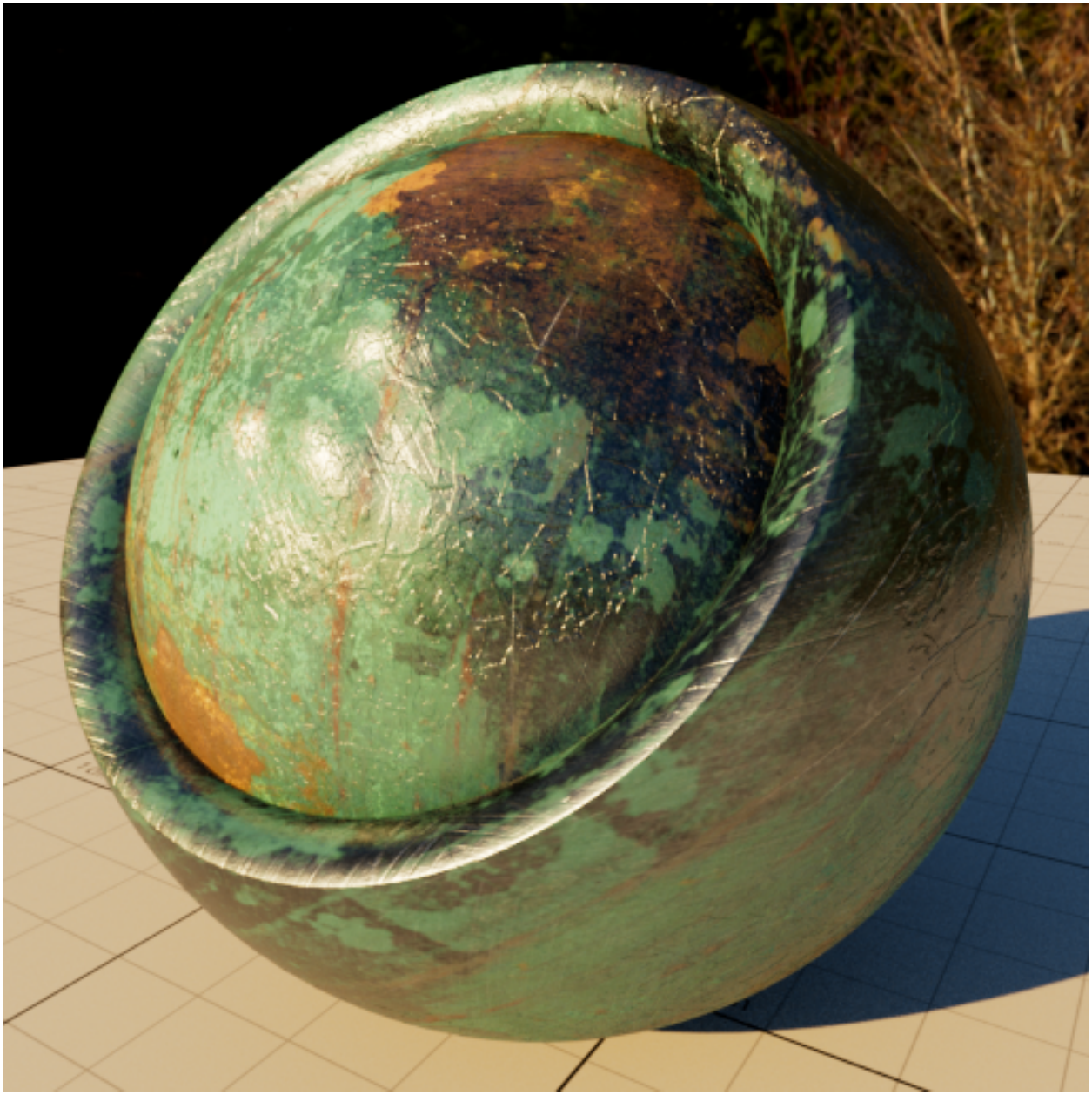




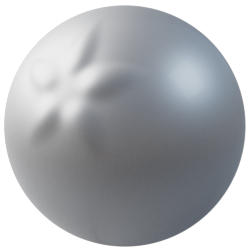








References



1.6 asDisneyMaterial

The Disney *principled* BRDF *[MHH+12]*, a physically based material.

1.6.1 Parameters

Common Material Parameters

Surface Color The surface color

Subsurface Amount The amount of sub-surface scattering[1]

Specular Parameters

Specular Amount The amount of specular highlights

Specular Roughness The apparent surface roughness affecting the specular highlights

Specular Tint The tinting amount of the specular highlights, with a value 0 having achromatic highlights, and a value of 1.0 inheriting the surface color.

Metallicness A low value is suitable for the appearance of dielectrics such as plastic, with a higher value producing more intensity specular reflections, and more suitable for conductors.

Anisotropy Amount The amount of anisotropy affecting the specular highlights, with a value of 0.0 producing isotropic highlights, and a value of 1.0 producing full anisotropic highlights.

Anisotropy Angle The angle affecting anisotropic highlights, which maps the values from [0,1] range into [0,360] range.

Anisotropy Vector Map A vector tangent field (color) map, with the *X* and *Y* anisotropy directions encoded in the *R* and *G* channels.

---

[1] In the shader, it's not a full BSSRDF, but an approximation using only single scattering.

Sheen Parameters

Sheen Amount The amount of sheen, see Westin velvet.

Sheen Tint The tinting (color) affecting the sheen effect.

Coating Parameters

Coating Amount The amount of (clear) coating on the material, producing an extra layer of (usually) sharp highlights.

Coating Glossyness The coating layer glossyness, with low values producing a dull like highlight, similar to a high surface roughness value, and higher values producing sharper highlights.

Bump Parameters

Bump Normal The unit length (bump) normal, usually passed from a *bump2d* or a *bump3d* node.

Advanced Parameters

Ray Depth The maximum number of bounces a ray is allowed to travel.

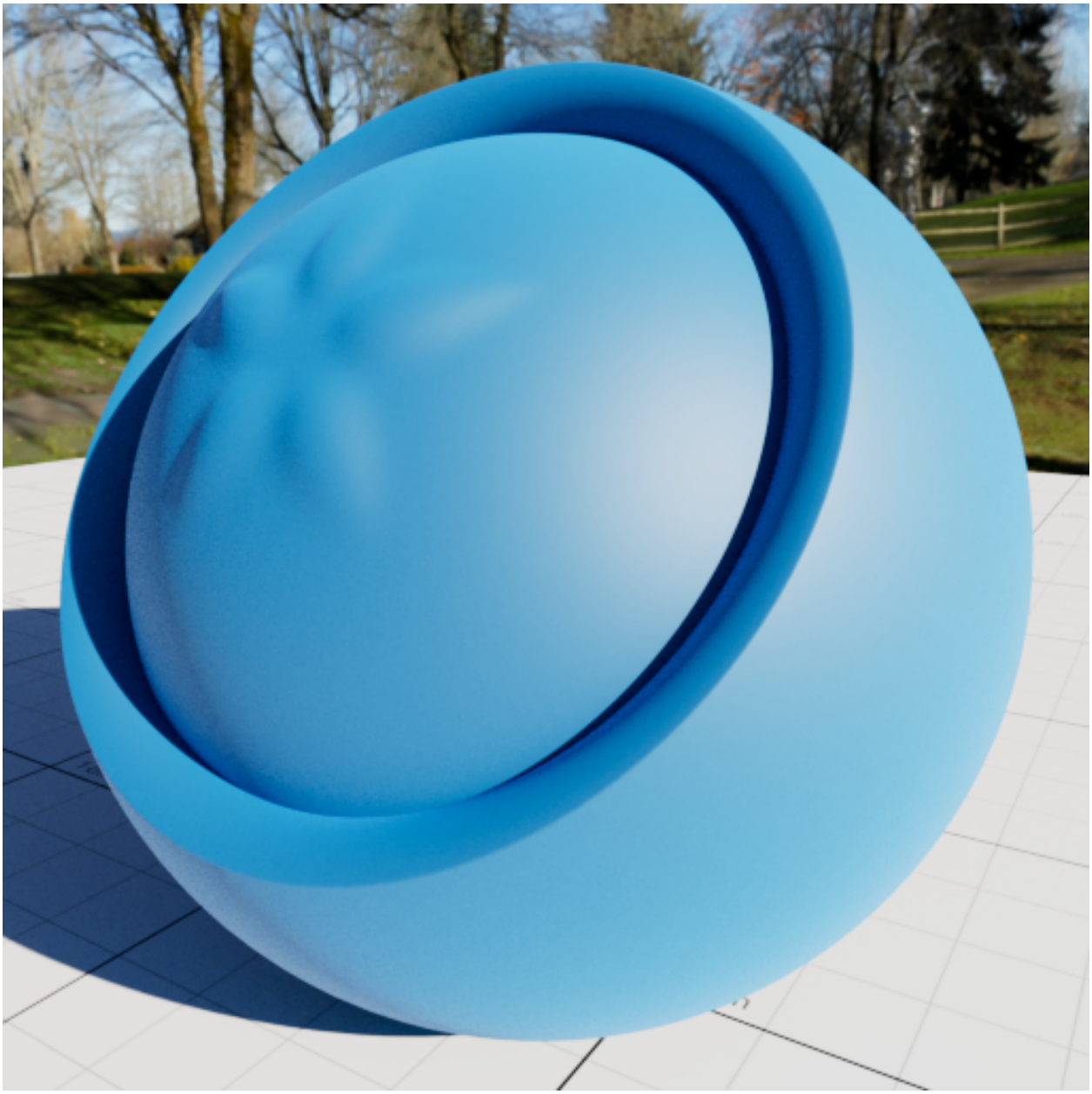
1.6.2 Outputs

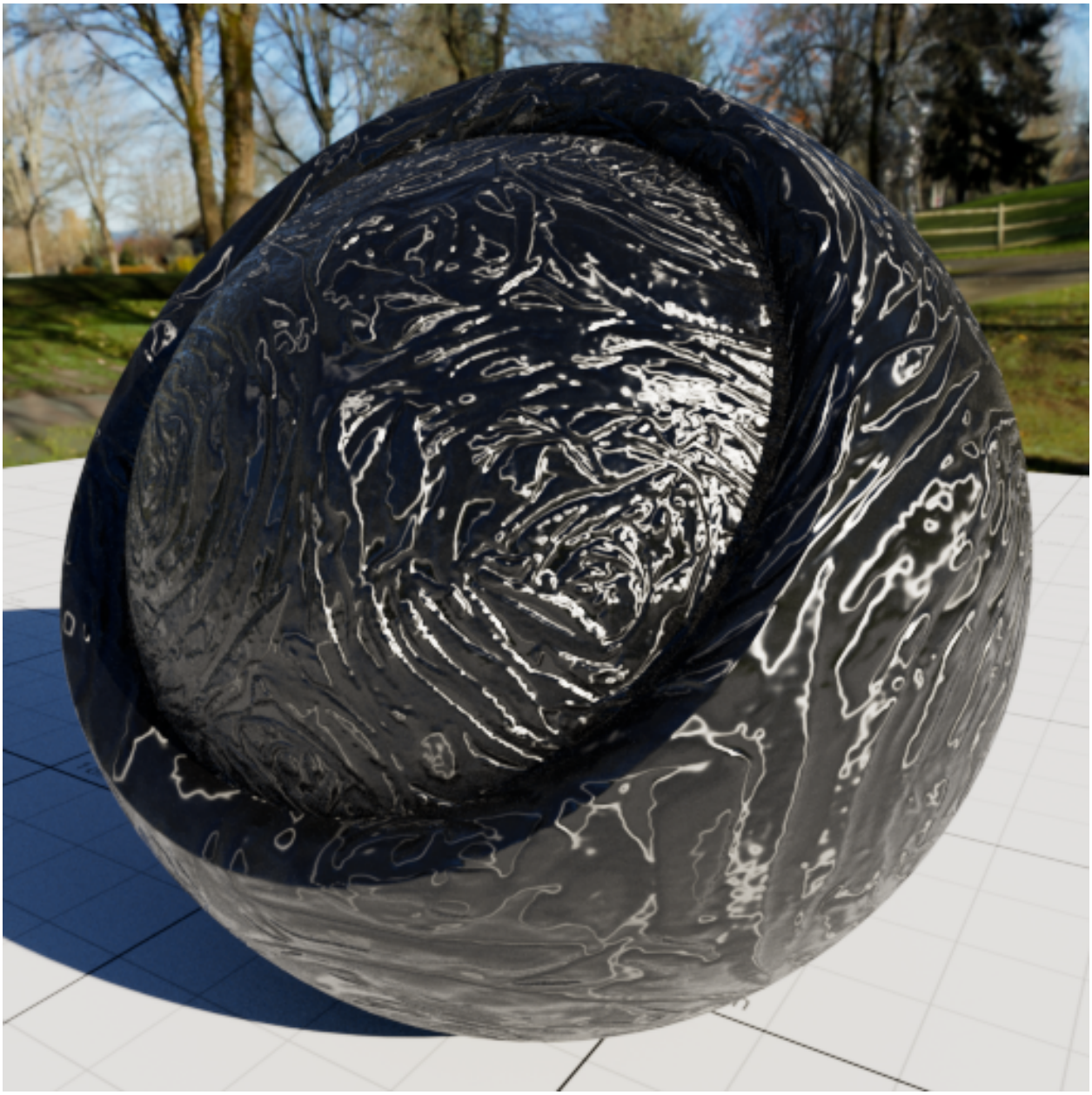
Output Color The final result color.

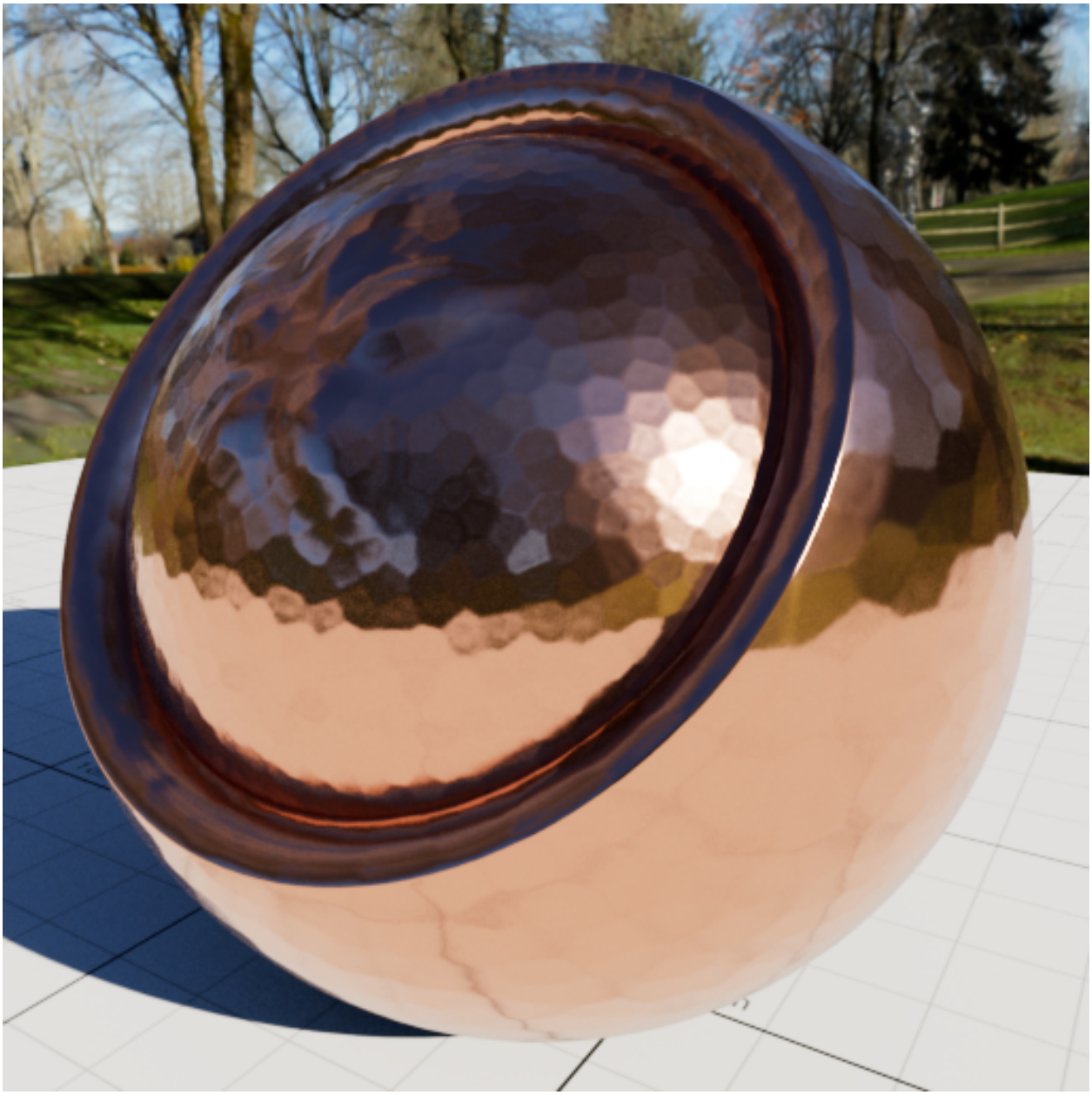
Output Transparency The resulting transparency (unused presently).

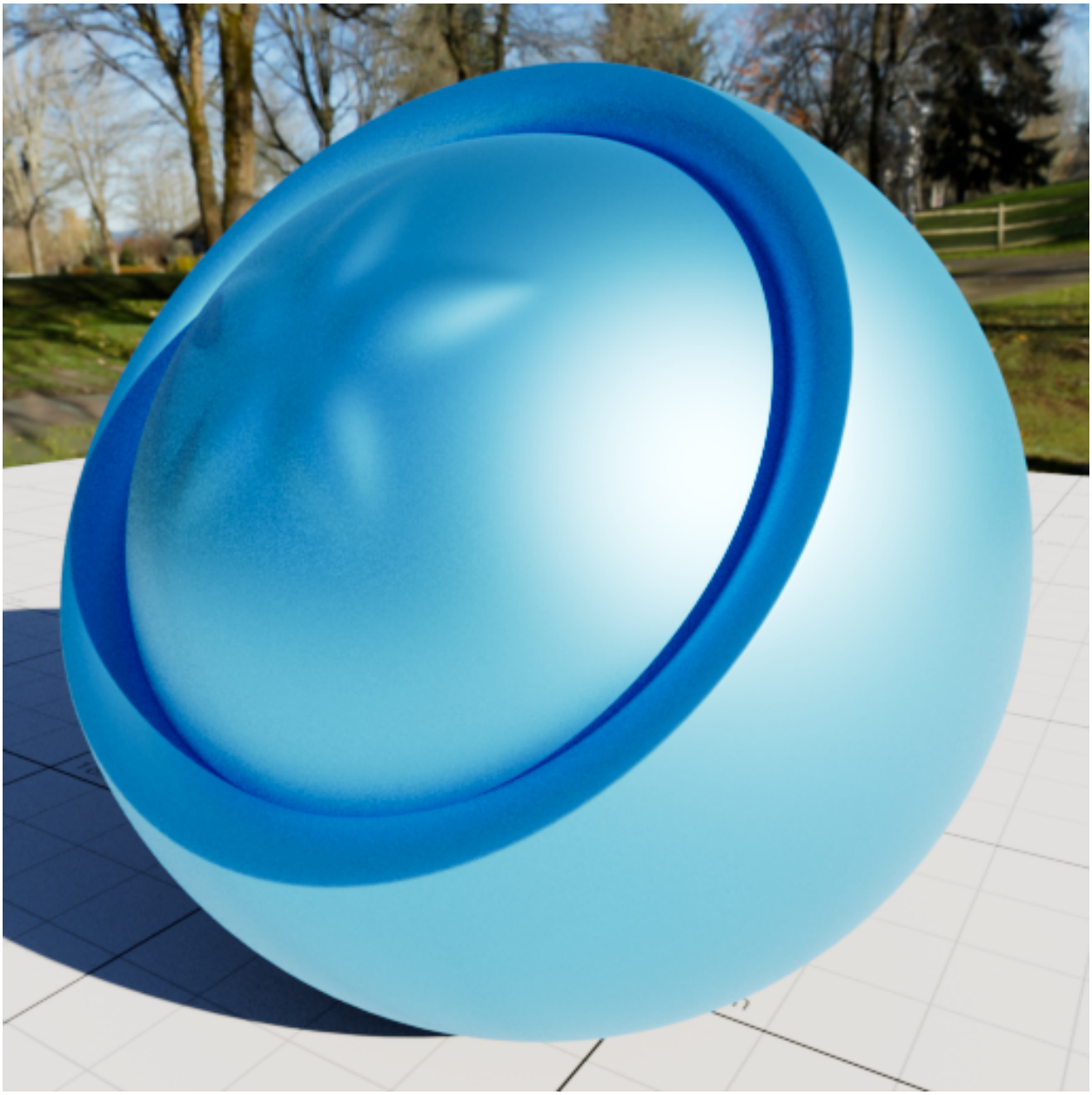
Output Matte Opacity The resulting matte opacity.

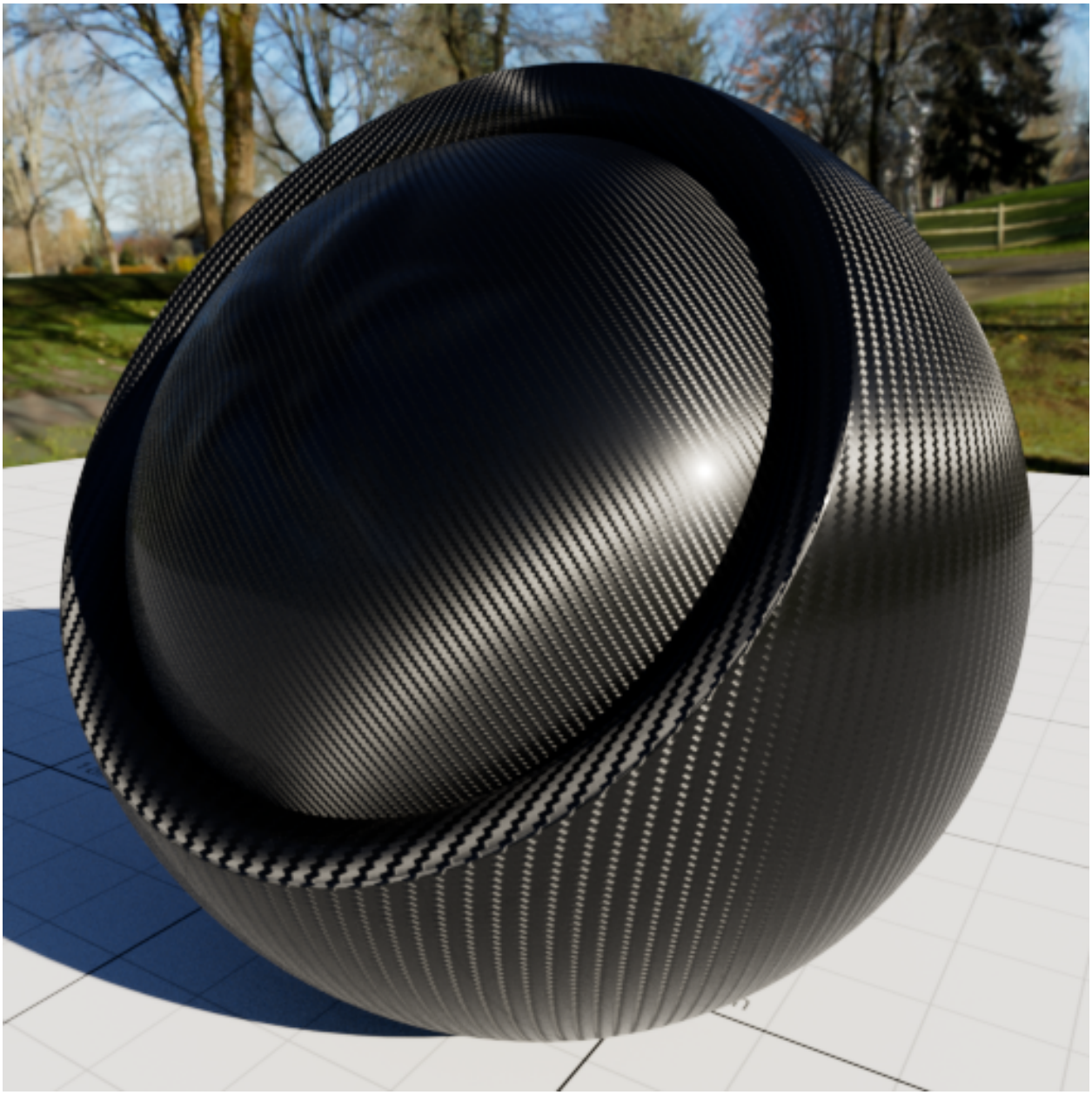
1.6.3 Screenshots

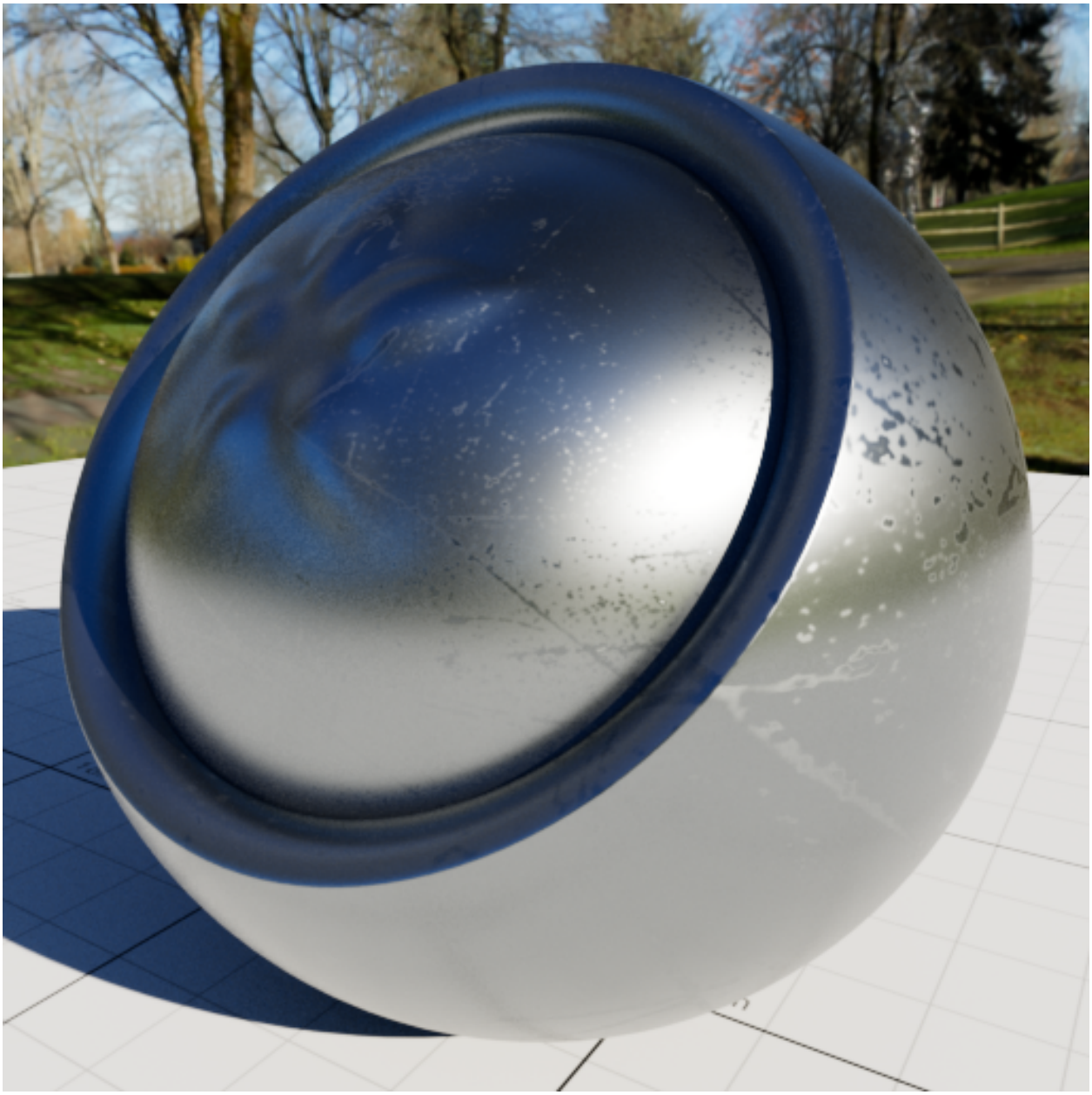


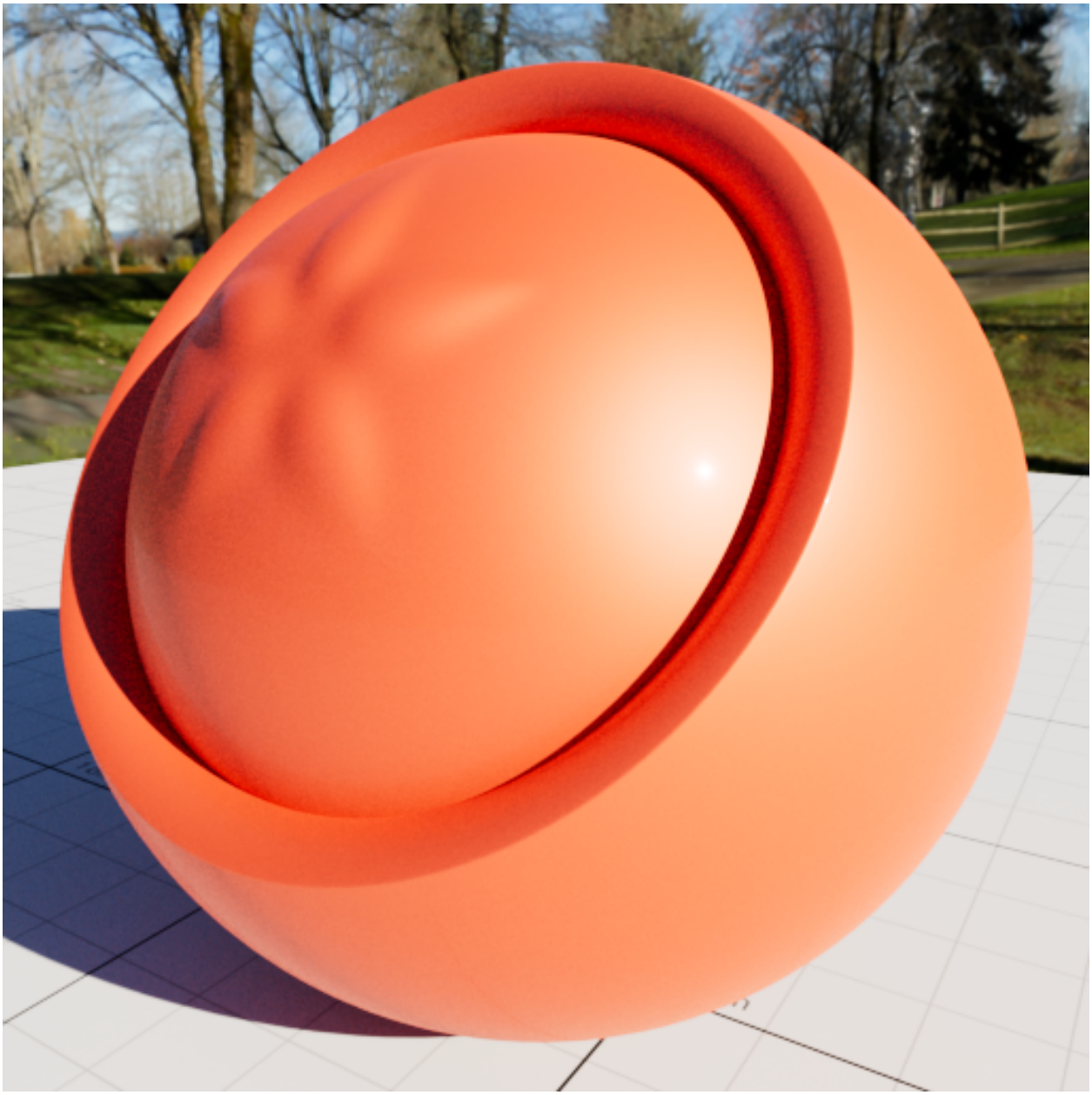


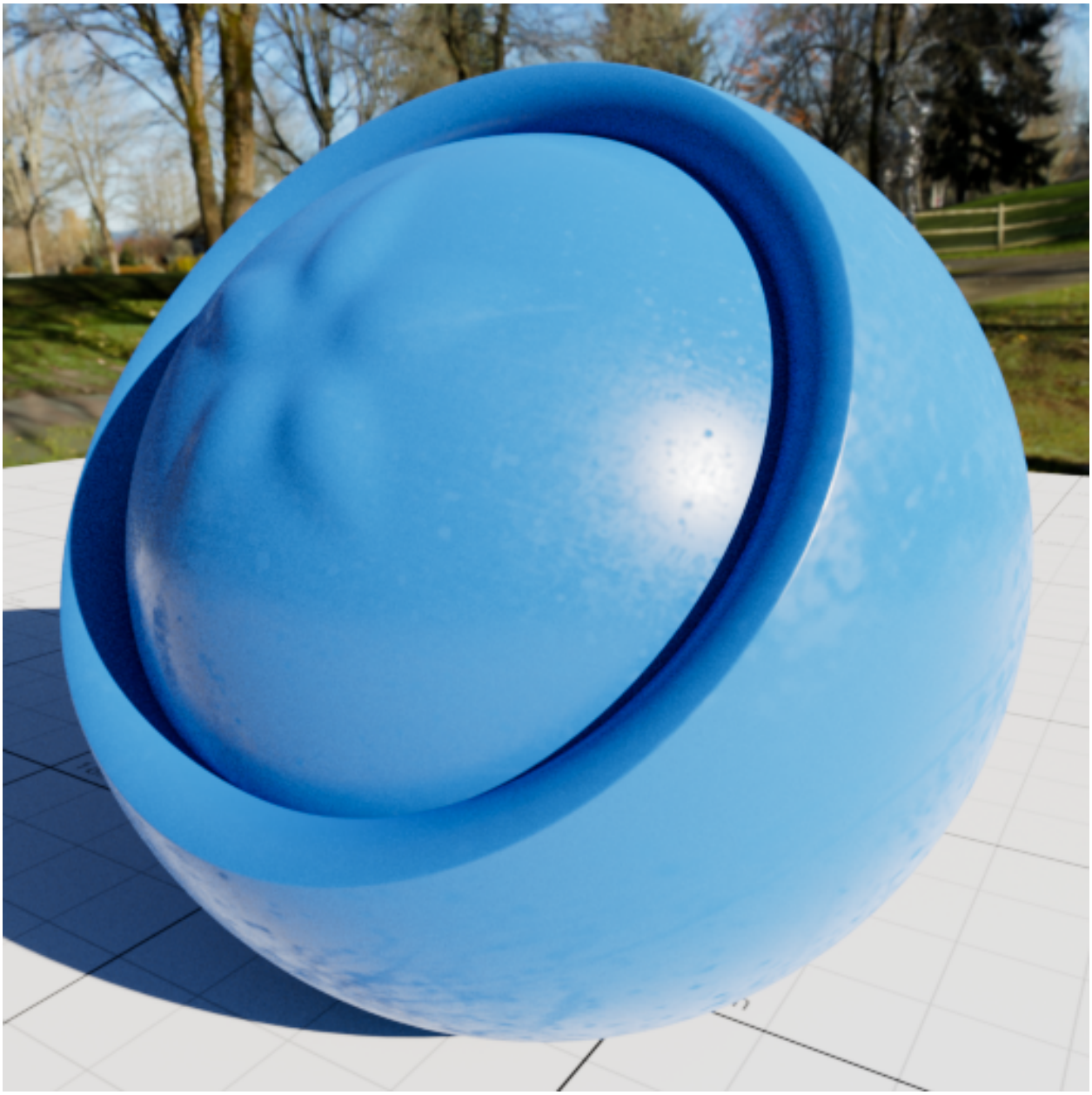


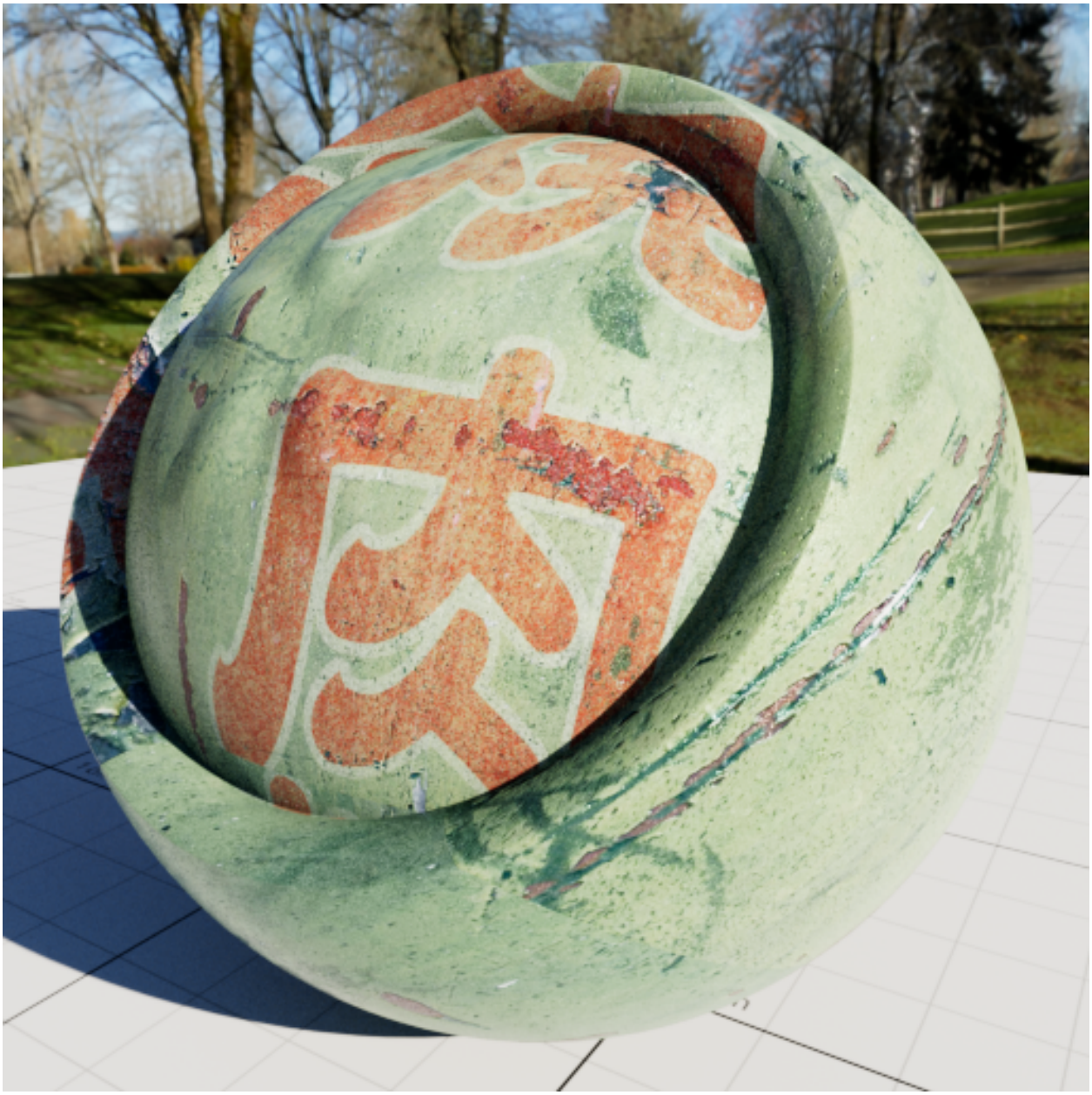


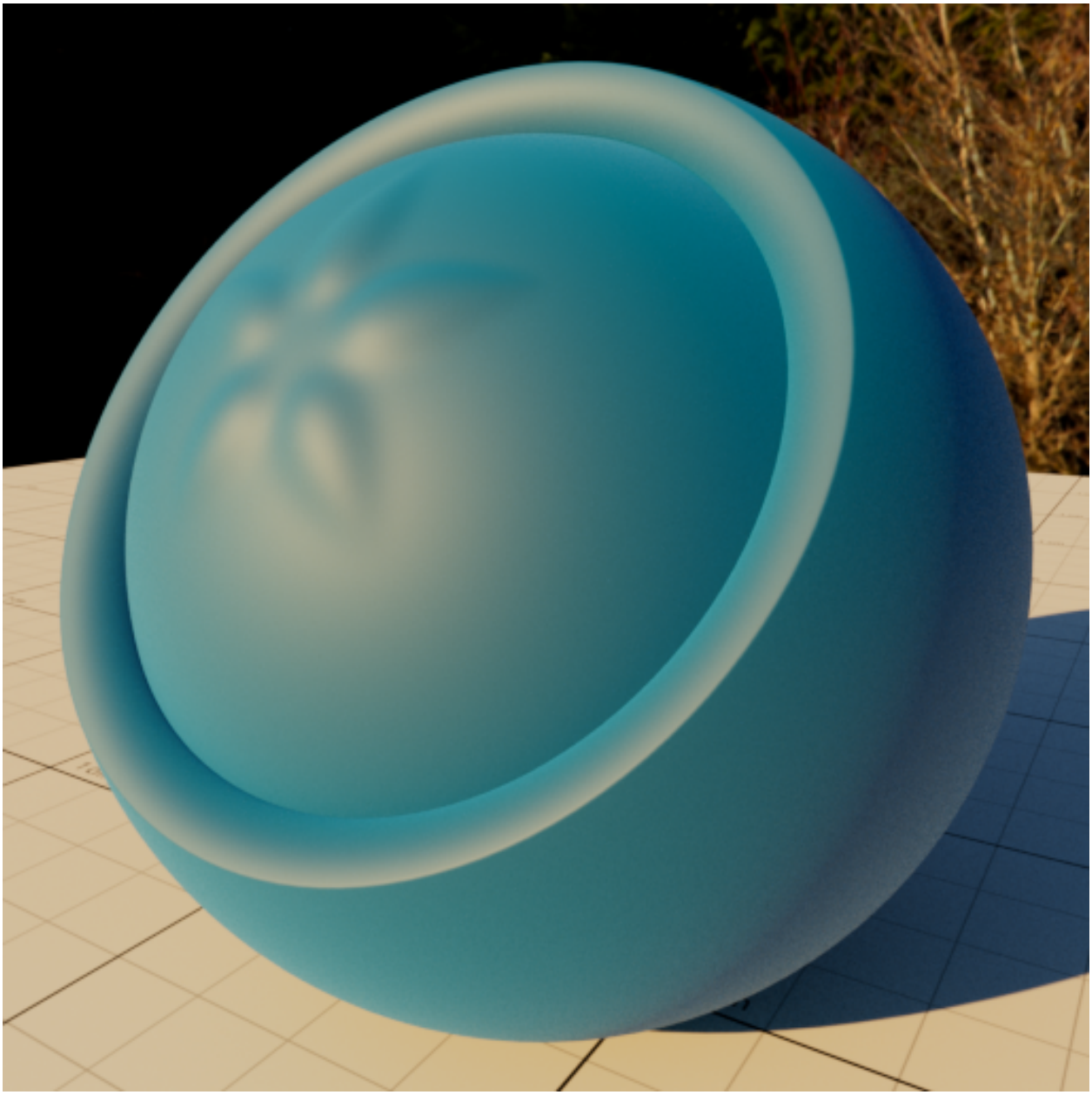


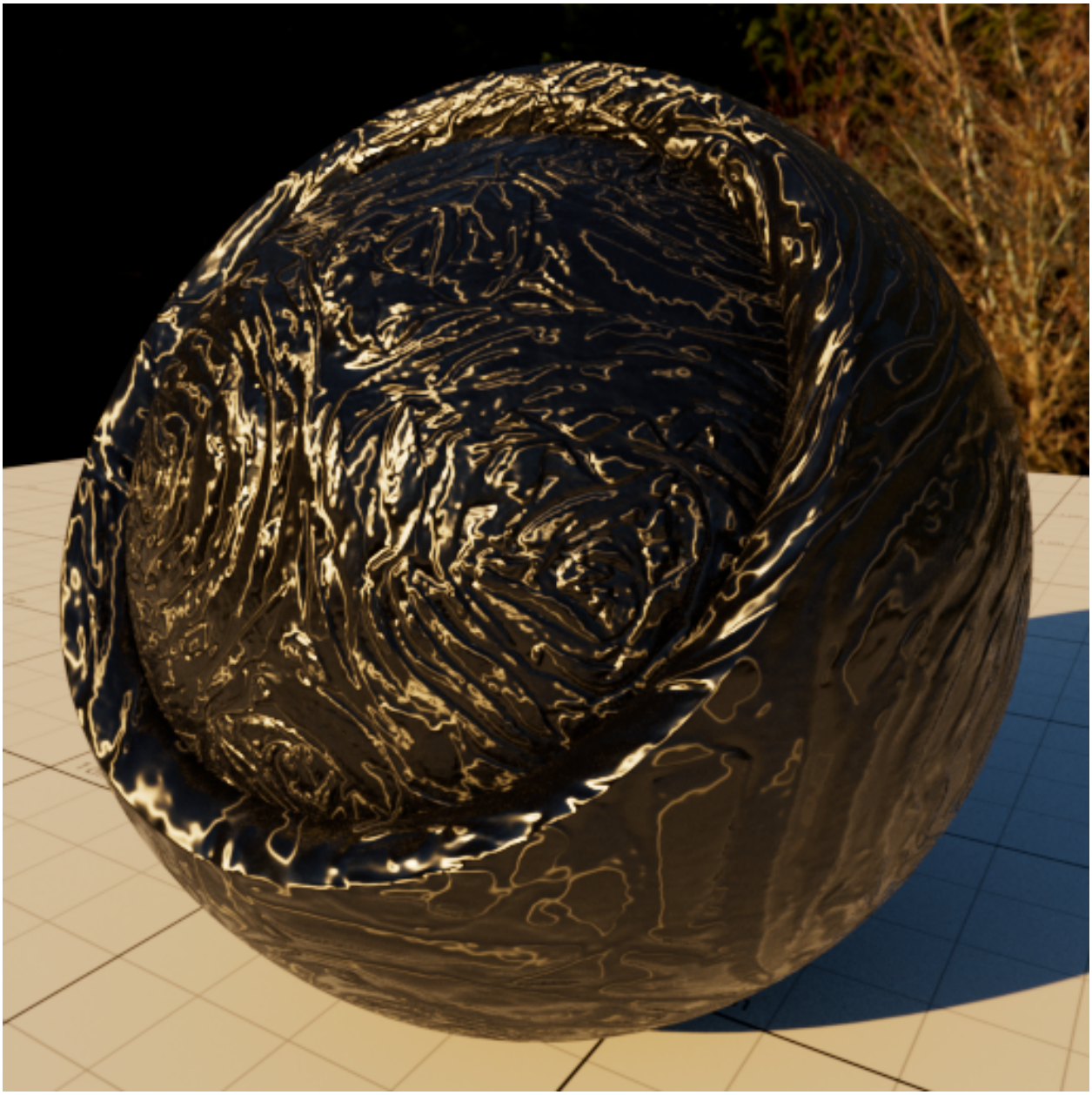


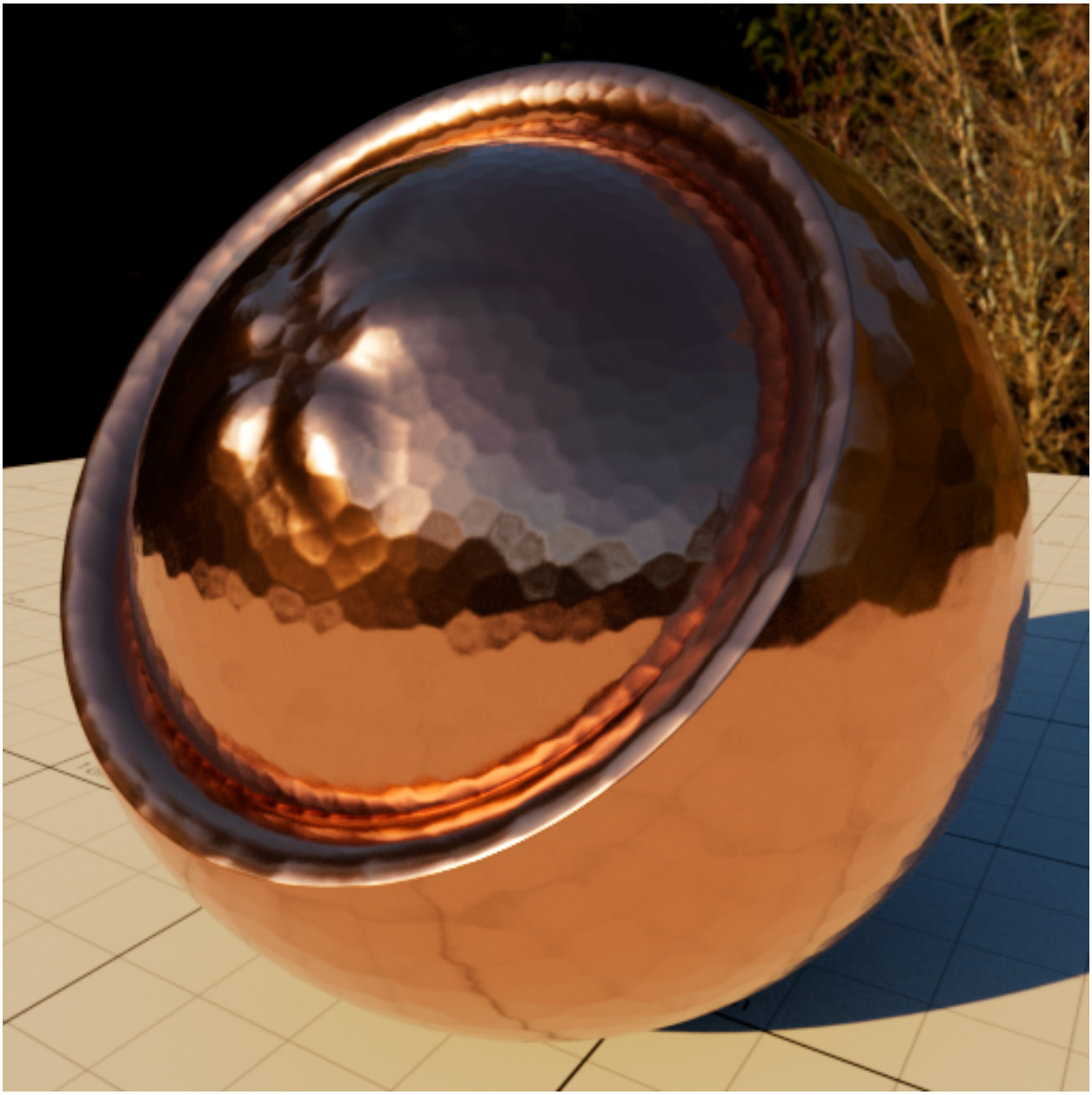


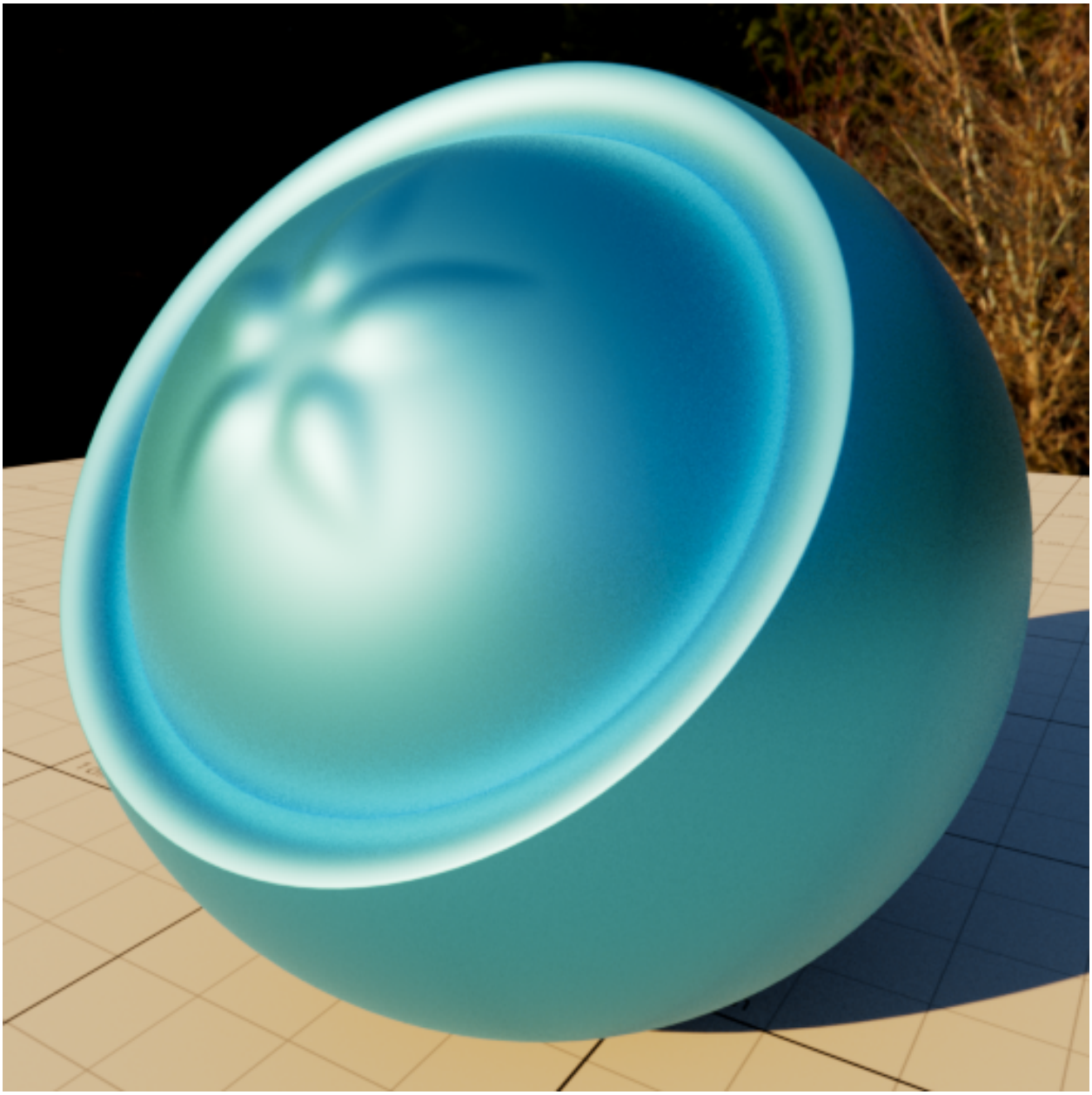


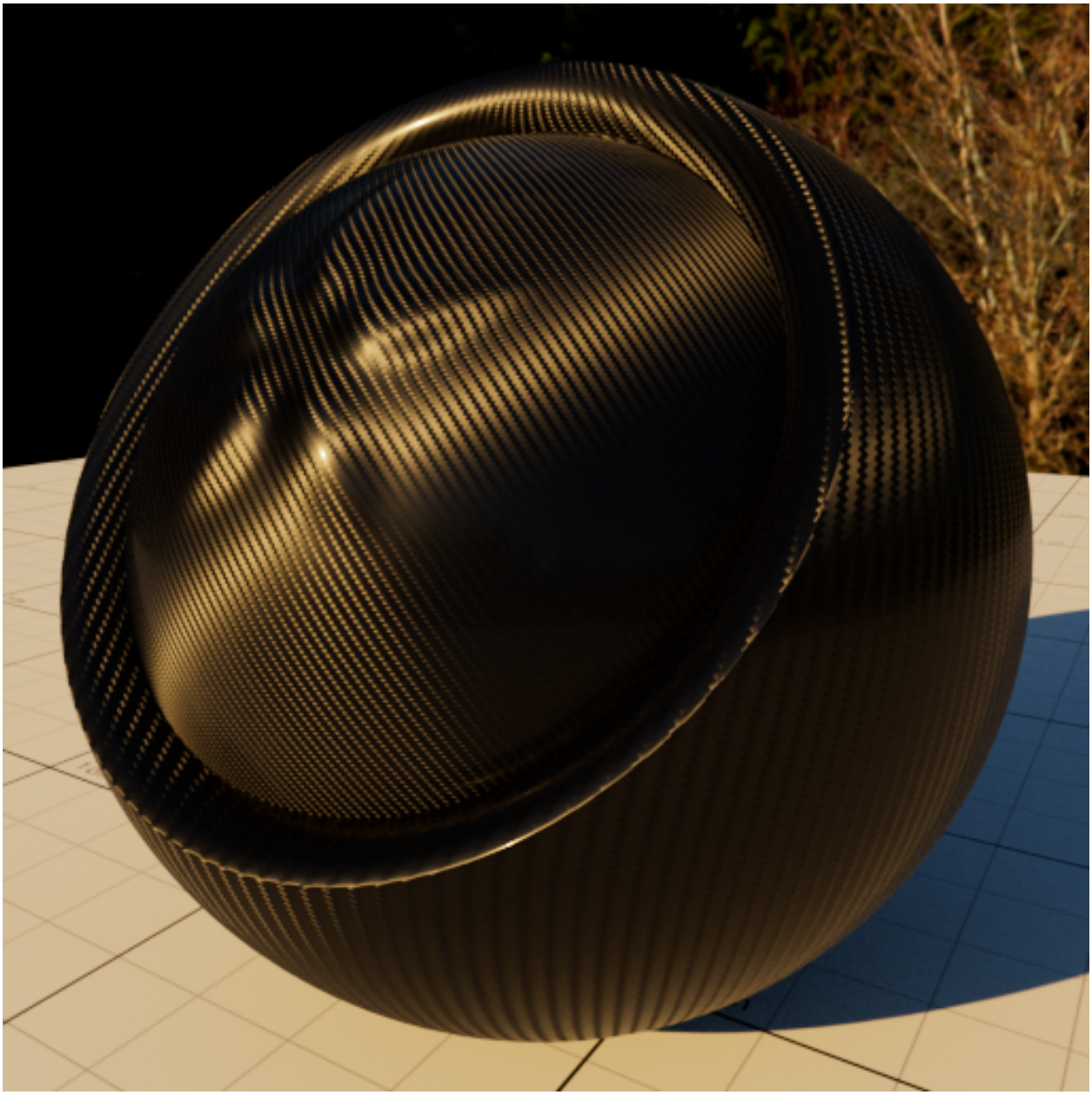


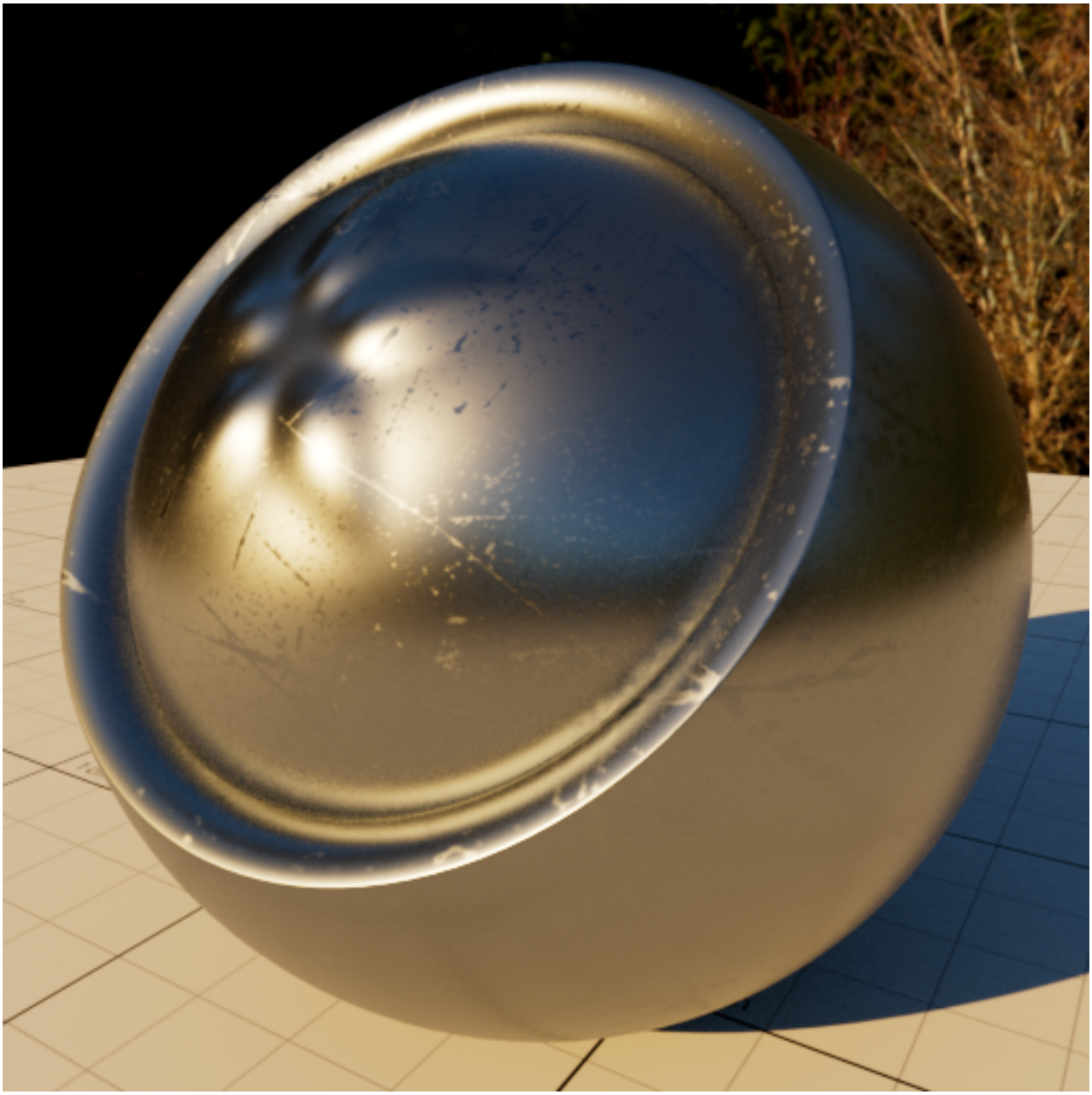


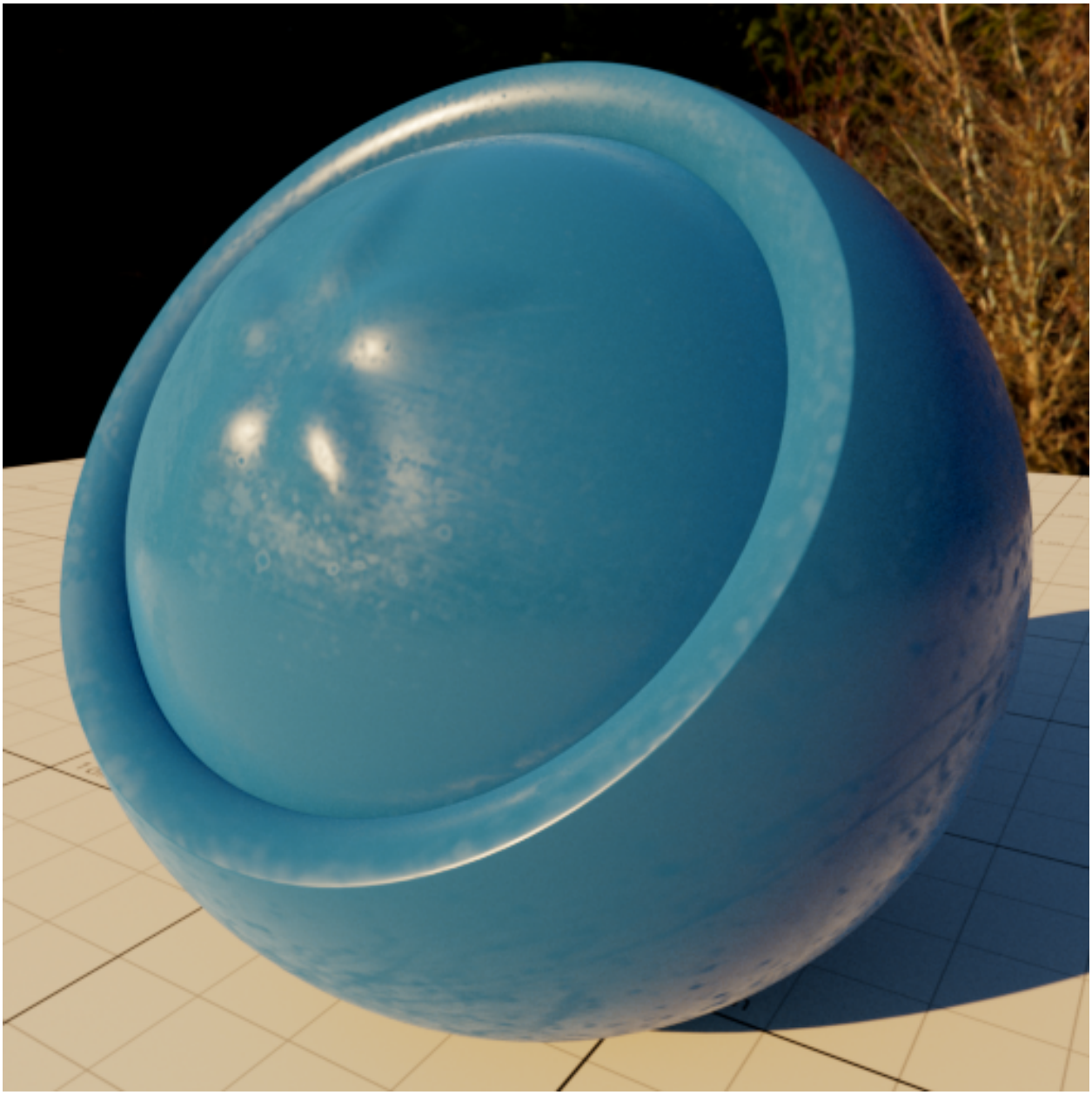












References

1.7 asGlass

A Glass BSDF *[WMLT07]*, with volumetric absorption.

1.7.1 Parameters

Surface Transmittance

Transmittance Color The color that is transmitted to the underlying glass medium. A color of 0 will mean no transmittance will take place, a color of one will mean full transmittance. This parameter is weighted by the next parameter, *transmittance amount*.

Transmittance Amount Transmittance amount, with a value of 0 meaning no light reaches the glass medium, and a value of 1.0 meaning a full amount reaching the glass medium. This is useful for compositing the glass BSDF with other BxDFs.

Specular Parameters

Reflection Tint Overall tint for the reflection (BRDF) component of the glass BSDF.

Refraction Tint Overall tint for the refraction (BTDF) component of the glass BSDF.

Index of Refraction Absolute index of refraction

Distribution Microfacet distribution function, it can be

1. *Beckmann [CT82]*
2. *GGX [HDEon14]*
3. *Student's t-distribution [RBMS17]*

Roughness The apparent surface roughness, affecting both the reflection and refraction equally.

Specular Spread Specular highlights spread. This controls the *tails* of the specular highlights, with high values producing a softer looking highlights, and lower values producing sharper looking highlights.

Note: This parameter is valid only for Student's t-distribution only. Higher values approach the GGX distribution, and lower spread values, therefore sharper highlights, approach the Beckmann distribution.

Anisotropy Amount Overall intensity of the anisotropy effect, with a value of 0.0 representing isotropic specular highlights.

Anisotropy Angle Rotation angle for the anisotropic highlights, with the value in [0,1] range mapping to a rotation from 0 to 360 degrees.

Anisotropy Vector Map Also known as tangent field, encodes the anisotropy directions along X and Y in the Red and Green or Red and Blue channels of the image. appleseed expects values encoded in the Red and Green channels.

Volume Material Parameters

Volume Transmittance Color of the volumetric absorption as the refracted ray travels within the medium.

Transmittance Distance The distance at which full absorption is supposed to occur. When this distance is set to 0, no absorption takes place. Lower values imply a stronger absorption, and higher values imply a weaker absorption effect as the ray would need to travel a greater distance for full absorption to take place.

Bump Parameters

Bump Normal The unit length world space normal of the bumped surface.

Advanced Parameters

Ray Depth The maximum ray depth a ray is allowed to bounce before being terminated.

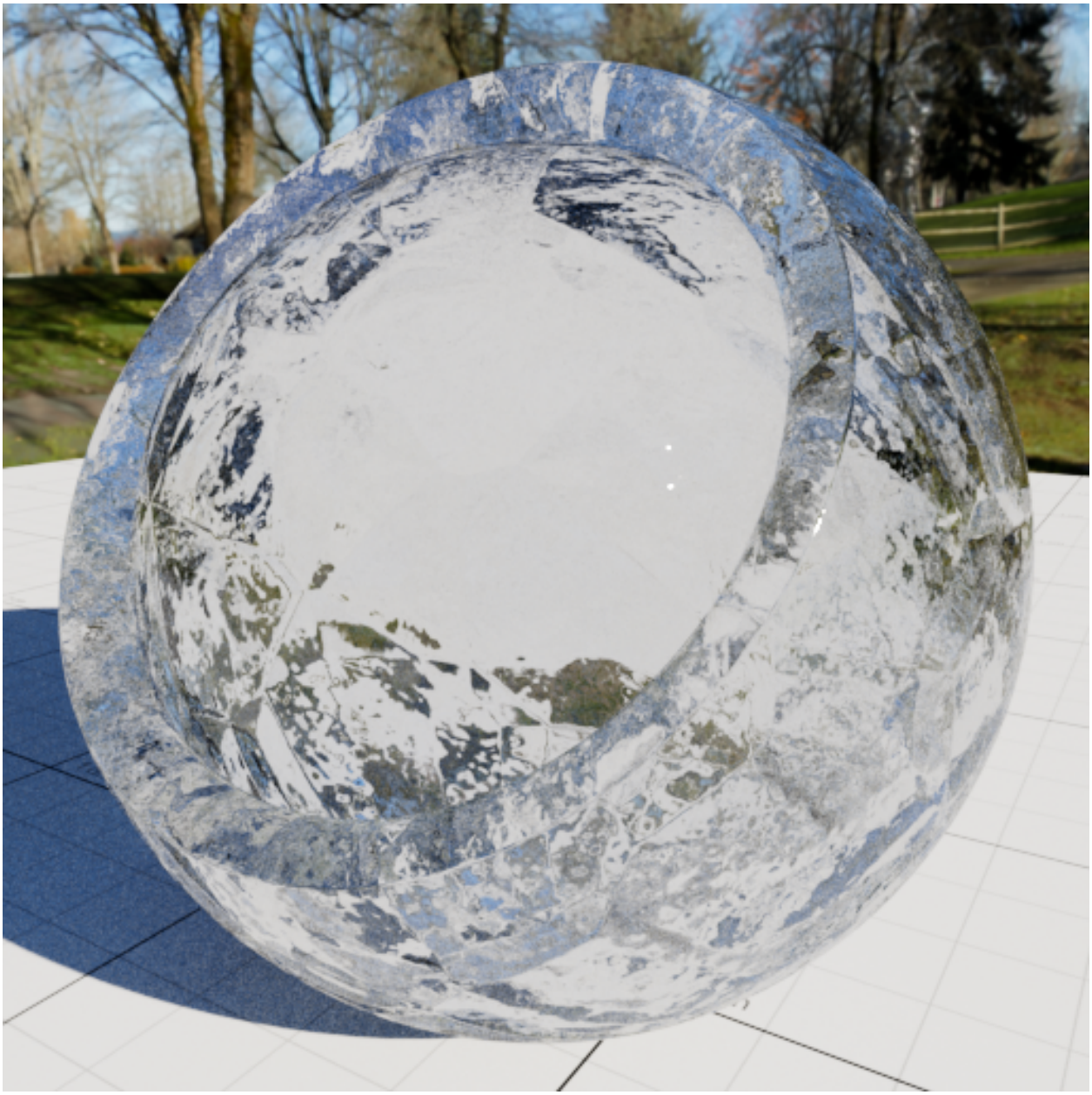
1.7.2 Outputs

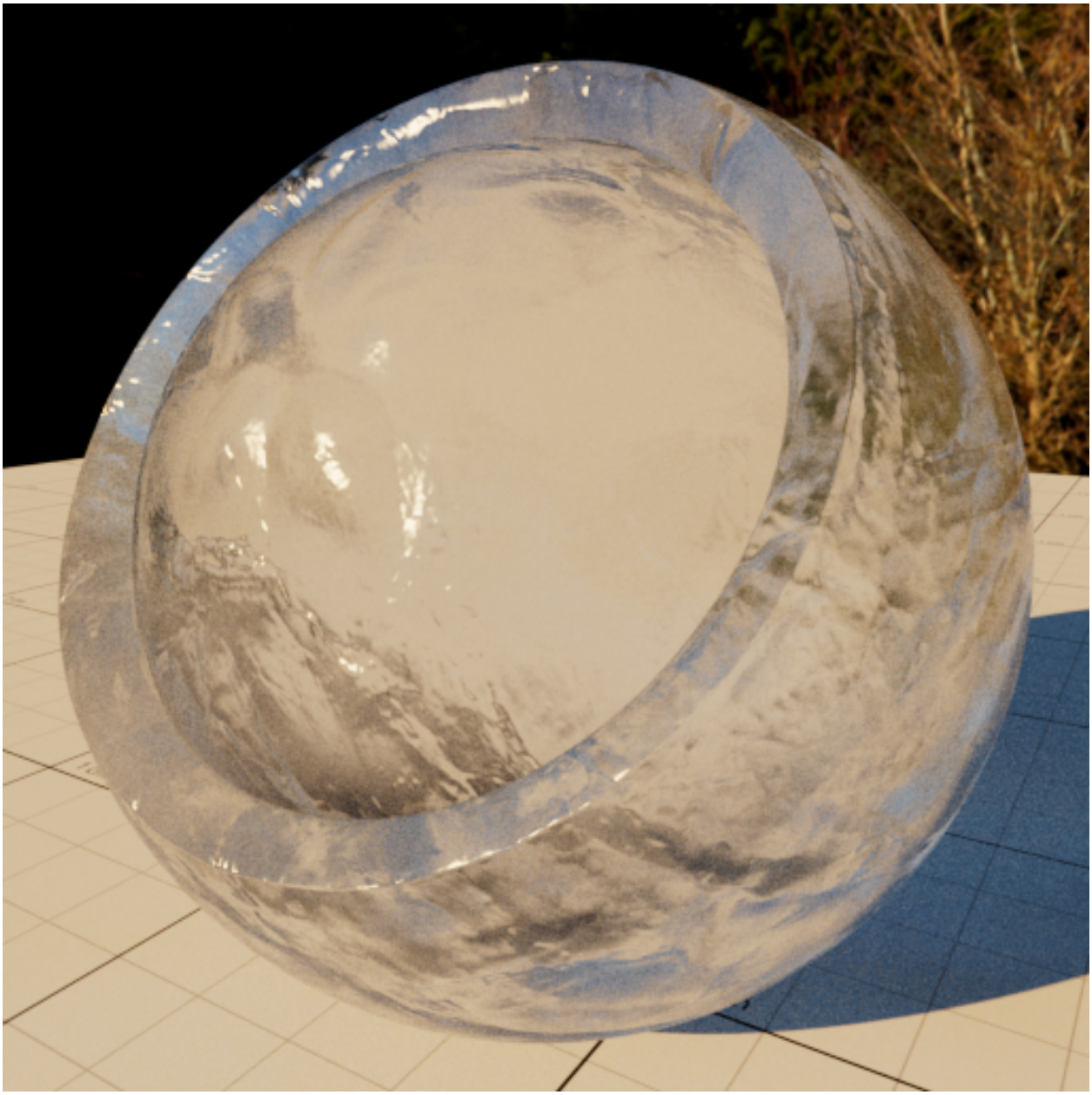
Output Color The BSDF output color.

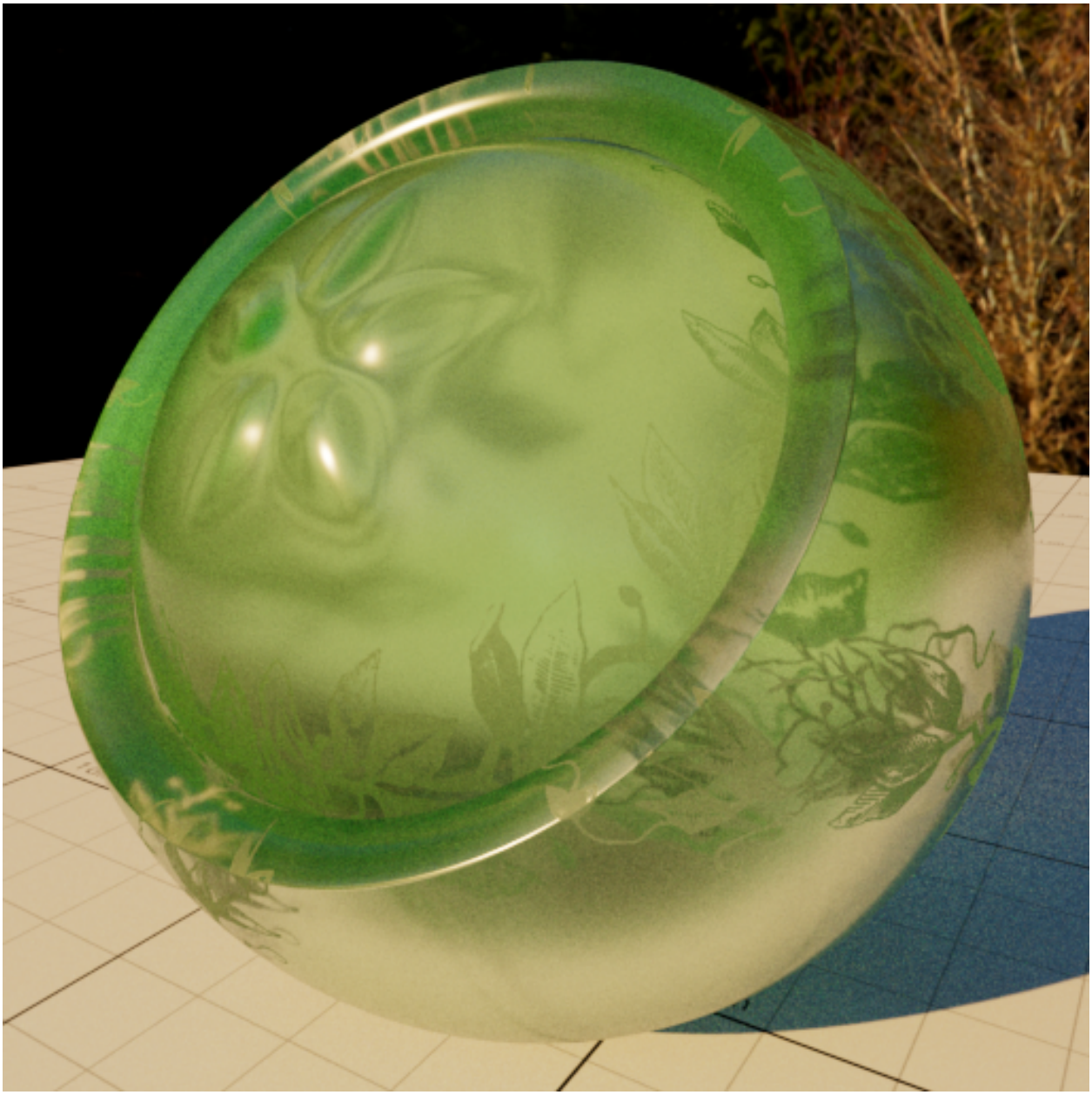
Output Transparency The output transparency.

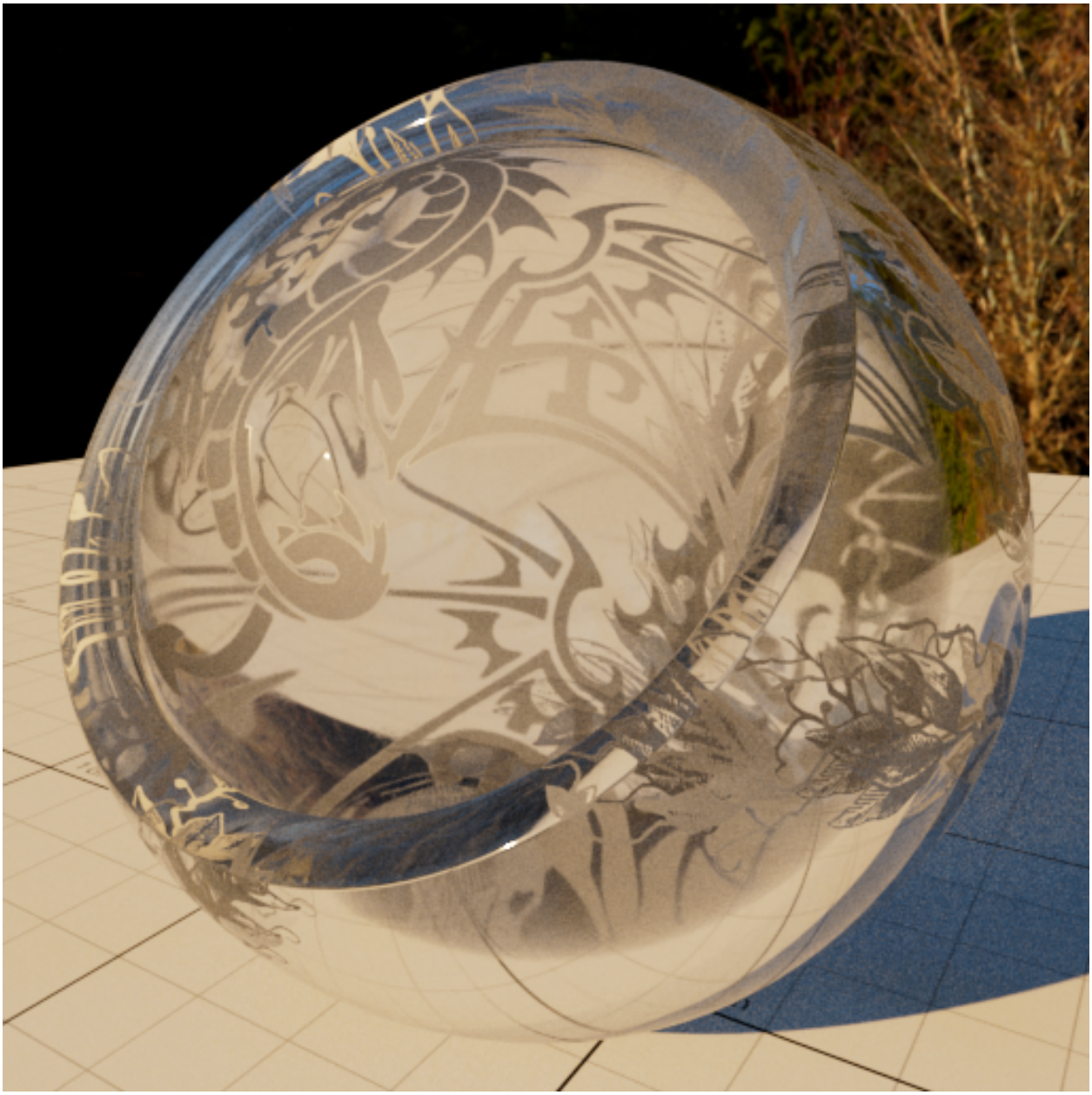
Note: The output transparency is unused at the moment.

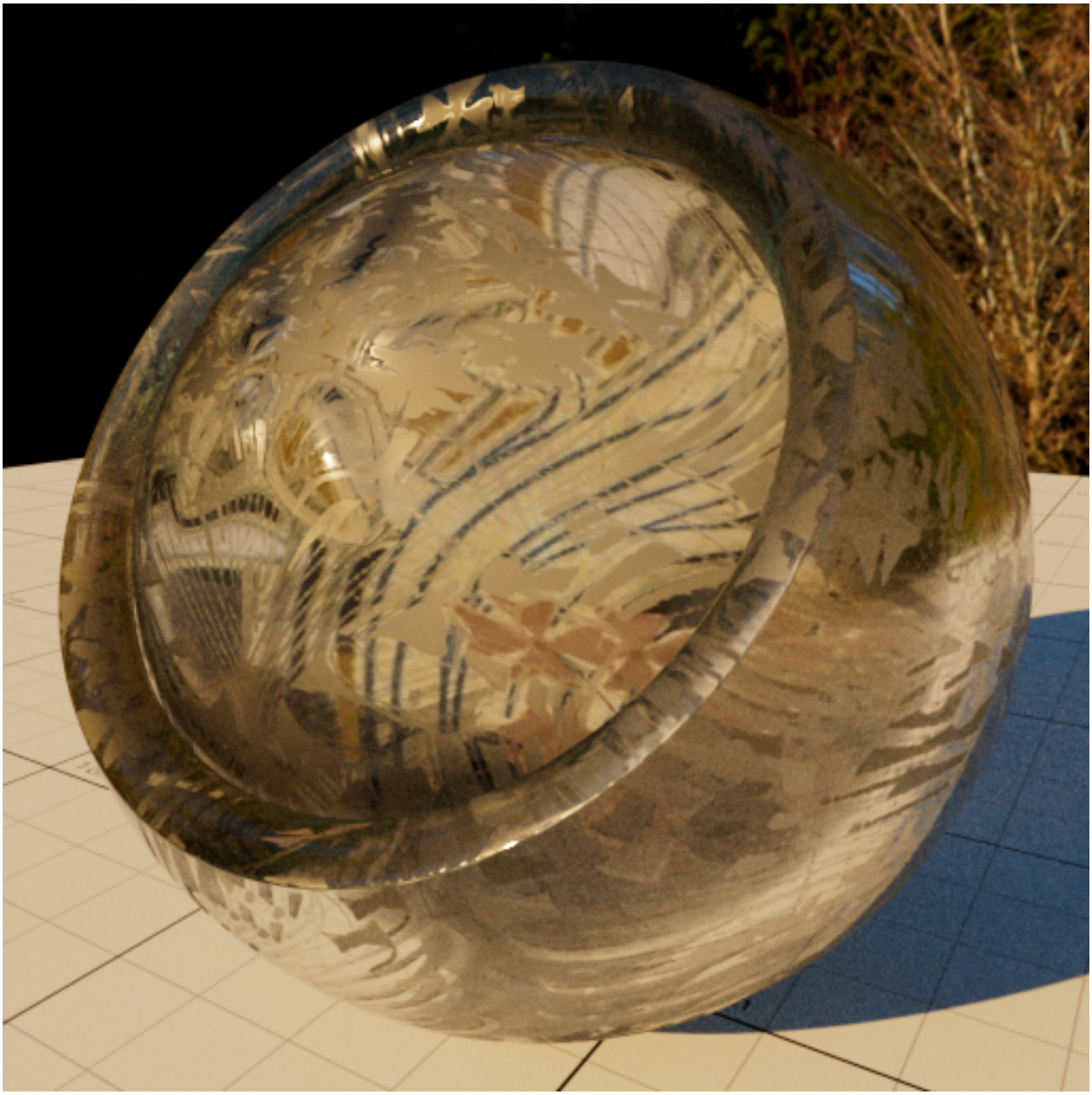
1.7.3 Screenshots

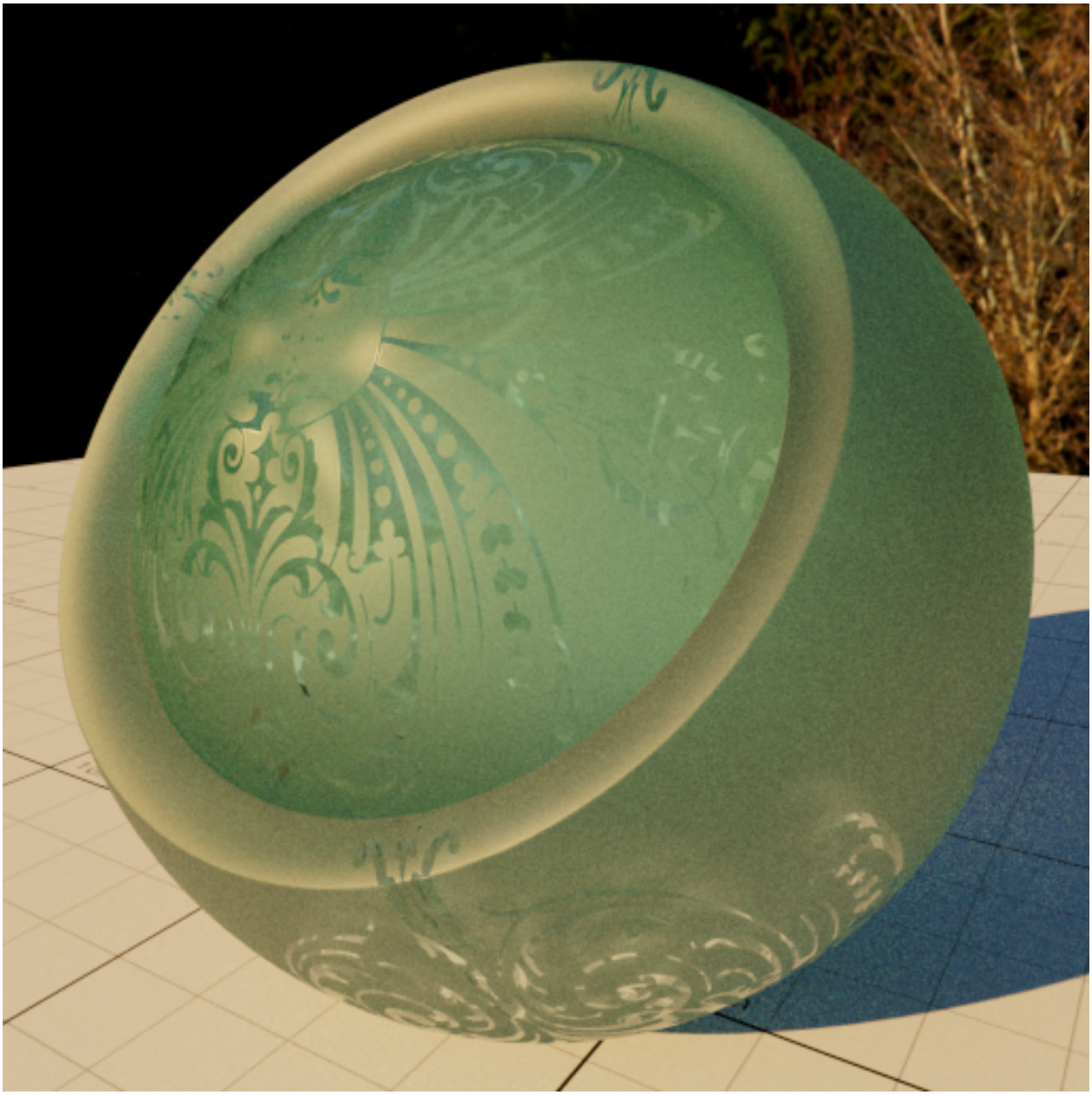


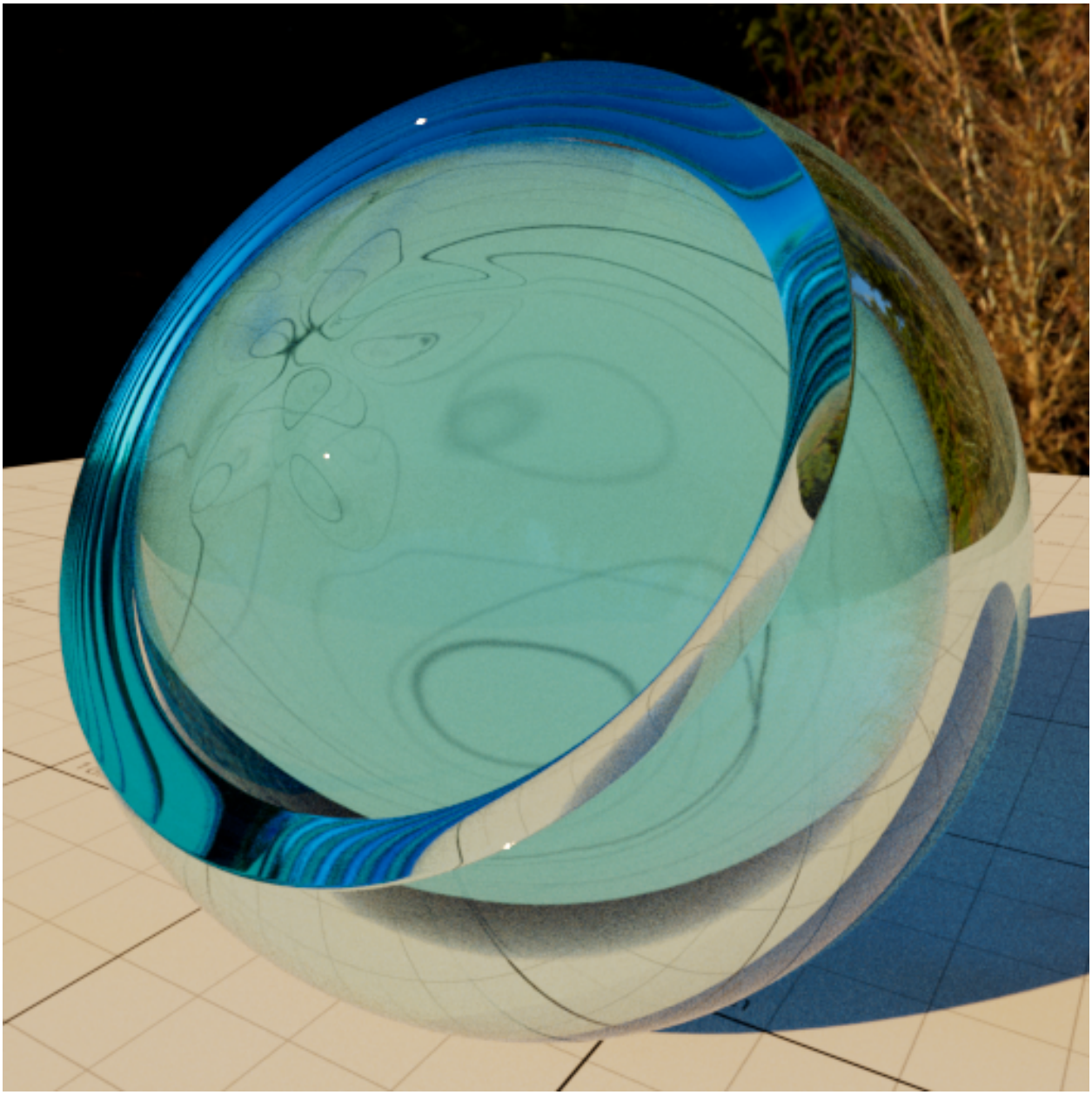












References

1.8 asMatte

A matte holdout shader allowing the user to specify an arbitrary color and matte opacity.

1.8.1 Parameters

Surface Parameters

Color The input material to pass-through for non-camera rays.

Note: Most materials have their own matte controls, however the user might wish to blend materials, with for instance, the *asBlendShader node*, and for such it might be preferable to set this matte node at the root of the shader graph.

Matte Parameters

Enable Matte Checkbox that toggles matte holdouts on or off.

Matte Color The color for the matte, only visible for *camera* rays.

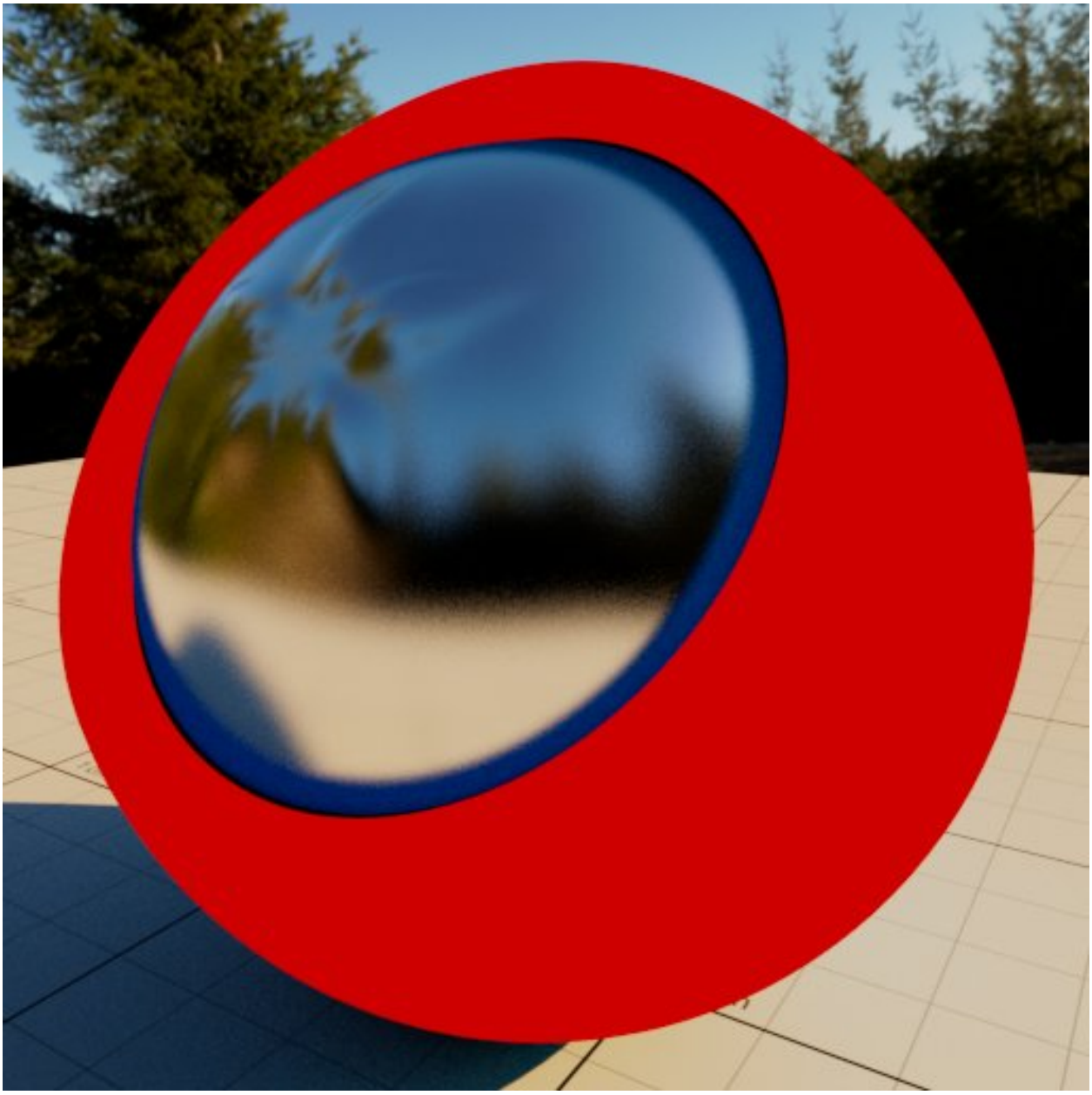
Matte Opacity The opacity value written to the alpha channel.

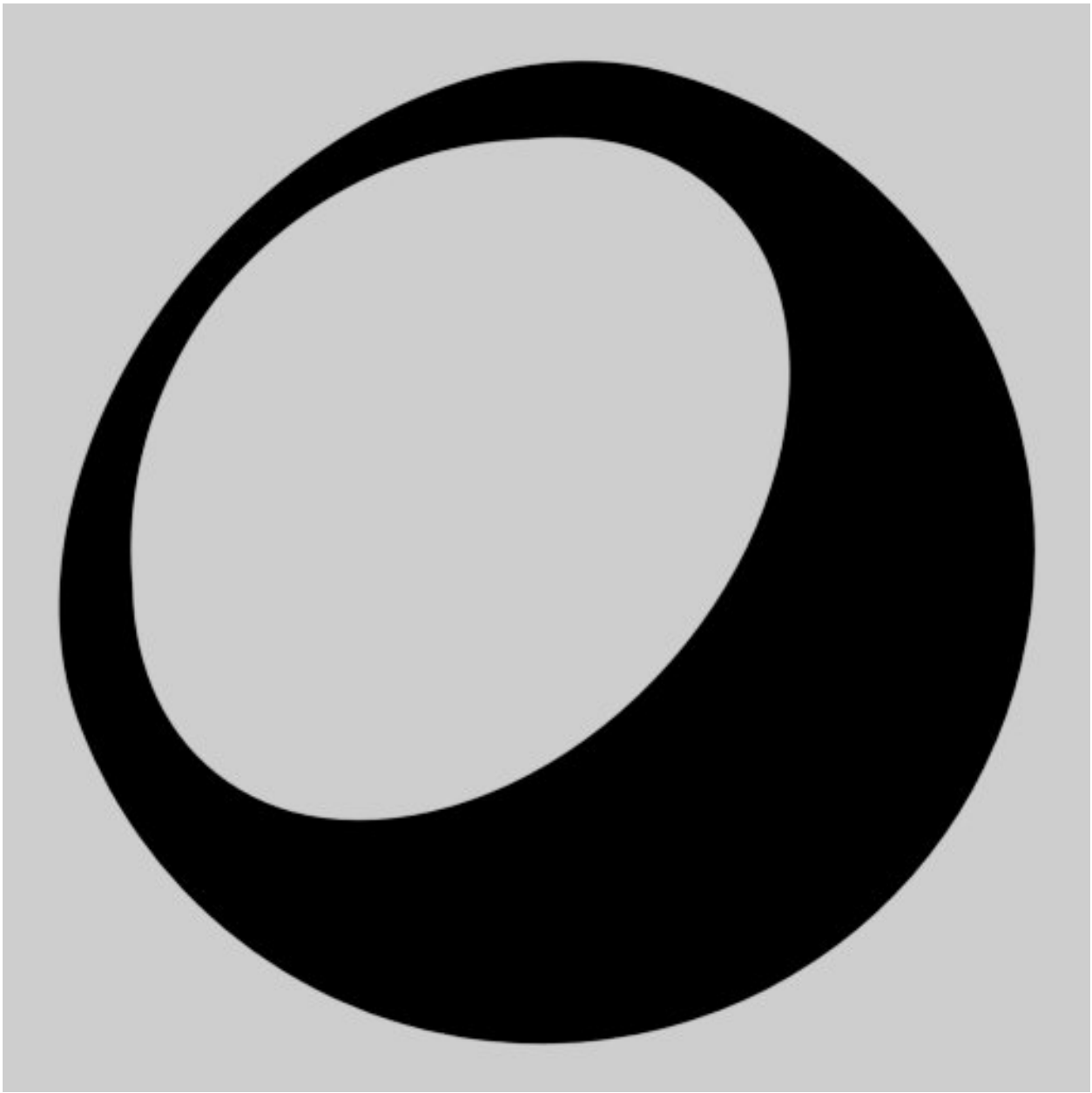
1.8.2 Outputs

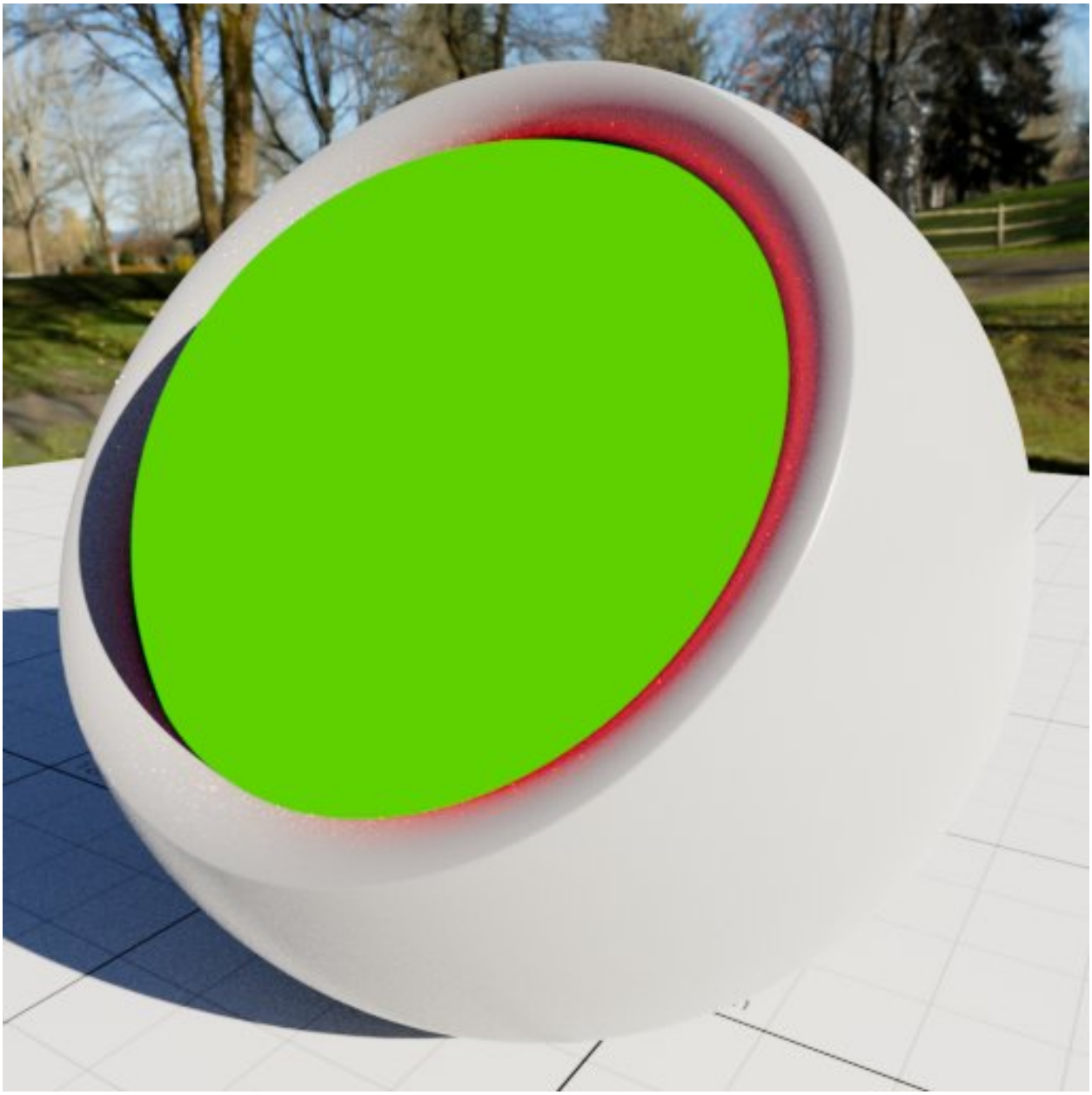
Output Color The pass-through closure output.

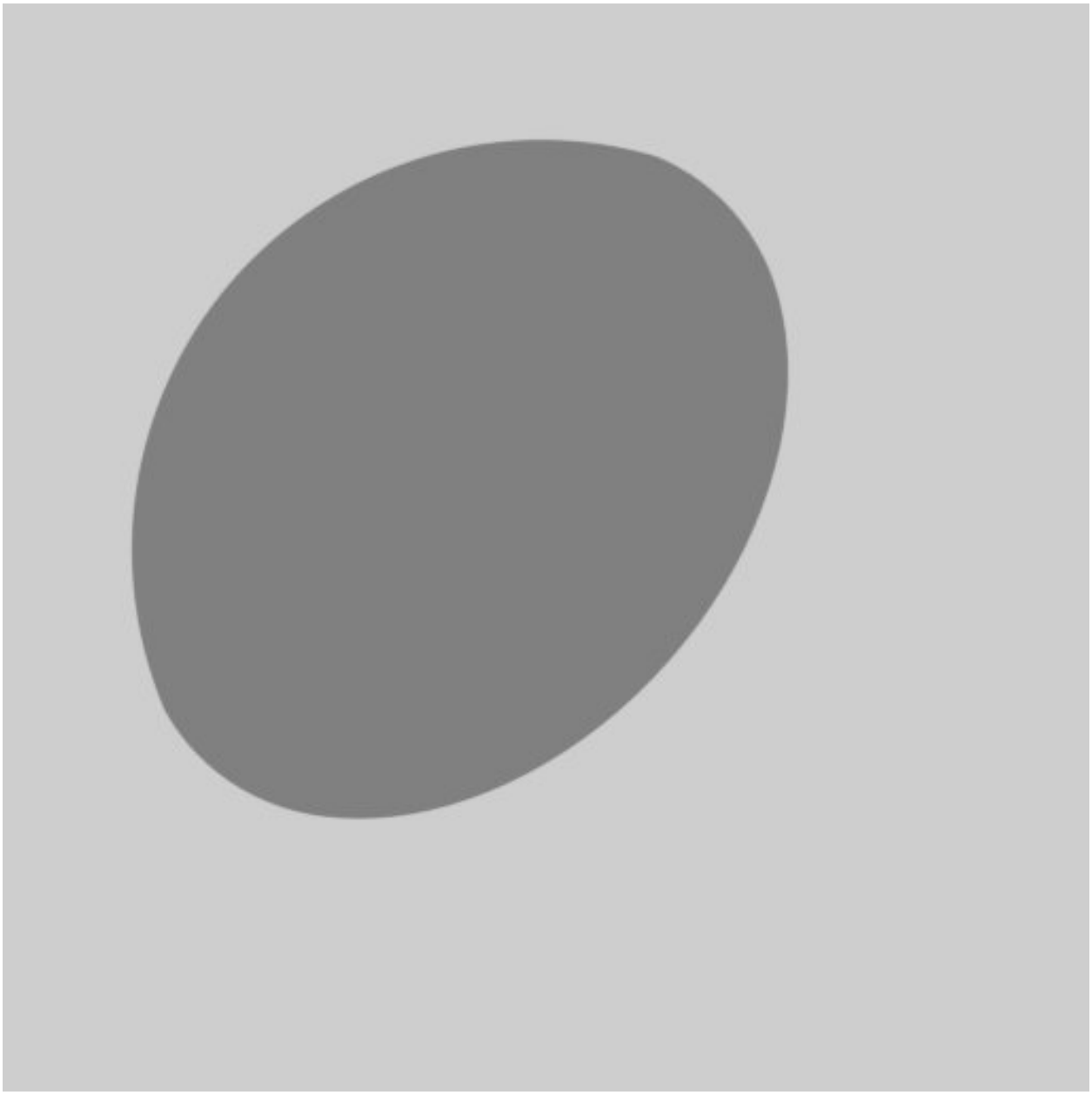
Output Matte Opacity The matte holdout.

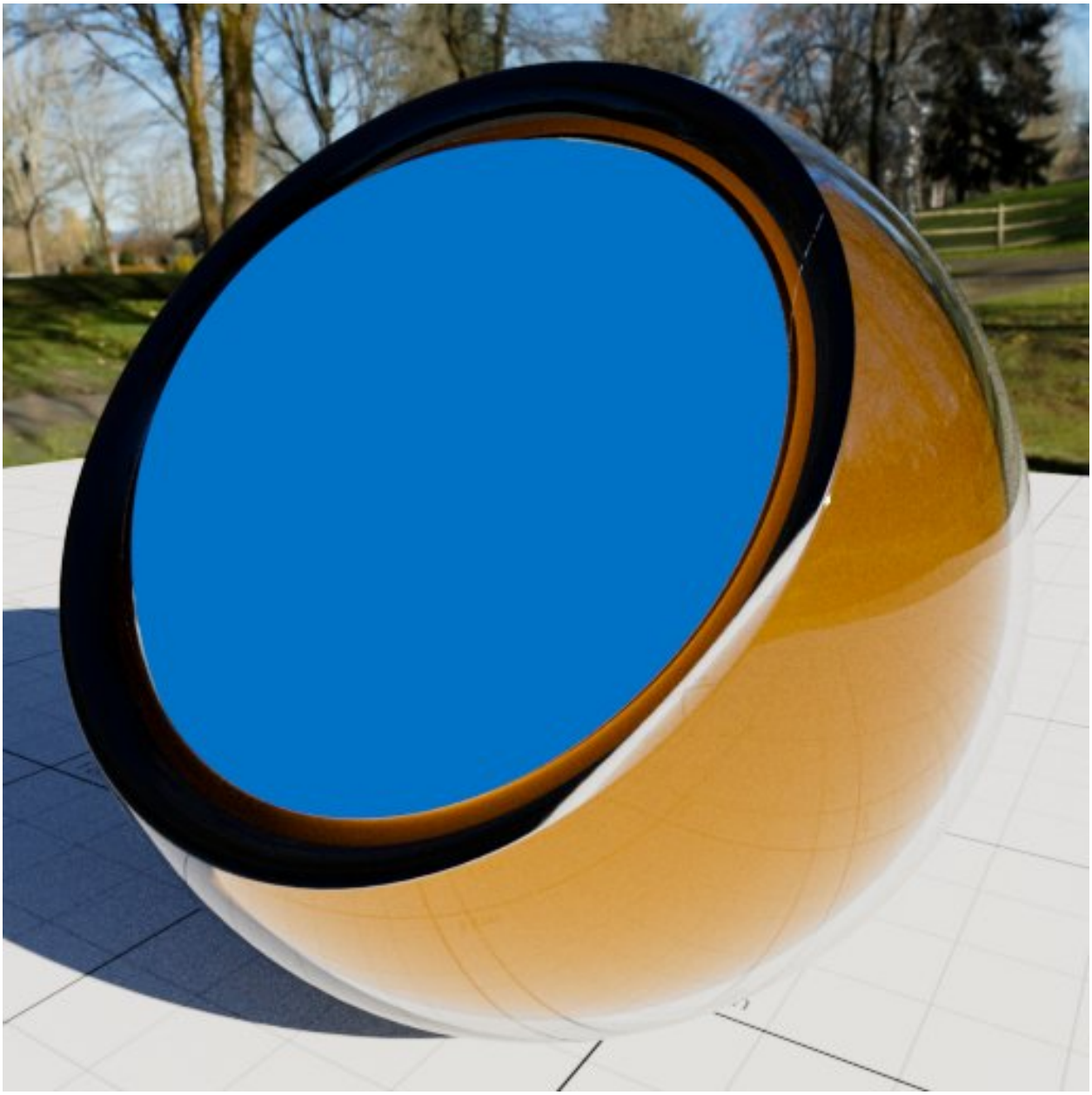
1.8.3 Screenshots











_images/screenshots/matte/as_matte_shot3_alpha.png

References

1.9 asMetal

A metal shader, with anisotropy and a user-friendly *[Gul14]* complex index of refraction Fresnel, with some presets for common metal types derived from real-world measurements[1].

1.9.1 Parameters

Fresnel Parameters

Face Reflectance RGB reflectance at normal or facing incidence.

Edge Reflectance RGB reflectance at edge or grazing incidence.

Specular Parameters

Distribution The microfacet distribution function to use, it can be one of

- GGX *[WMLT07]*
- Beckmann *[CT82]*

Roughness The apparent surface roughness.

Energy Compensation Microfacet models typically fail to take into account multiple scattering *[HHdEonD16]*, and as such, with high roughness values there is a substancial energy loss. To what extent it depends mostly on the MDF used, with *GGX* exhibiting considerable energy loss, and more so than the *Beckmann* MDF.

In order to negate the impact of this energy loss a separate compensation term is applied. This parameter scales the contribution of this compensation term, with a value of 1.0 trying to compensate for all the energy lost, and a value of 0.0 essentially disabling any compensation.

See also:

This is covered in SIGGRAPH 2017 physically based shading course Revisiting Physically Based Shading at Imageworks (Christopher Kulla and Alejandro Conty).

[1] From *.nk* files, containing the data for several metals, alloys, semi-conductors in several wavelenght ranges (not exclusively in the visible light range). Maya attribute presets are provided for aluminium, brass, chromium, copper, gold, iron, lead, mercury, nickel, osmium, platinum, aluminium-gold intermetallic (*purple plague*), silver, titanium, titanium nitride, tungsten and zinc. See the LuxPop database and refractive index database for more measured complex ior data.

Anisotropy

Anisotropy Amount Overall intensity of the anisotropy effect, with a value of 0.0 representing a isotropic specular highlight.

Anisotropy Angle Rotation angle for the anisotropic highlight in [0,1], mapping a rotation from 0 to 360 degrees.

Anisotropy Mode Toggles between accepting a direct texture map in the form of an anisotropy vector map, or between an explicit vector (or a connection to a node that generates such a vector[2]). It can take the values

- Anisotropy Map
- Explicit Vector

Anisotropy Map Also known as tangent field, encodes the anisotropy directions along X and Y in the Red and Green or Red and Blue channels of the image. Appleseed expects values encoded in the Red and Green channels. Valid when the *Anisotropy Mode* is set to *Anisotropy Map* only.

Anisotropy Direction The explicit vector passed as the anisotropy direction. Valid when the *Anisotropy Mode* is set to *Explicit Vector* only.

Bump Parameters

Bump Normal The unit length world space normal of the bumped surface.

Matte Opacity Parameters

Enable Matte Opacity Parameter that toggles matte holdouts.

Matte Opacity Matte opacity scaling factor.

Matte Opacity Color Holdout color.

Advanced Parameters

Ray Depth The maximum ray depth a ray is allowed to bounce before being terminated.

1.9.2 Outputs

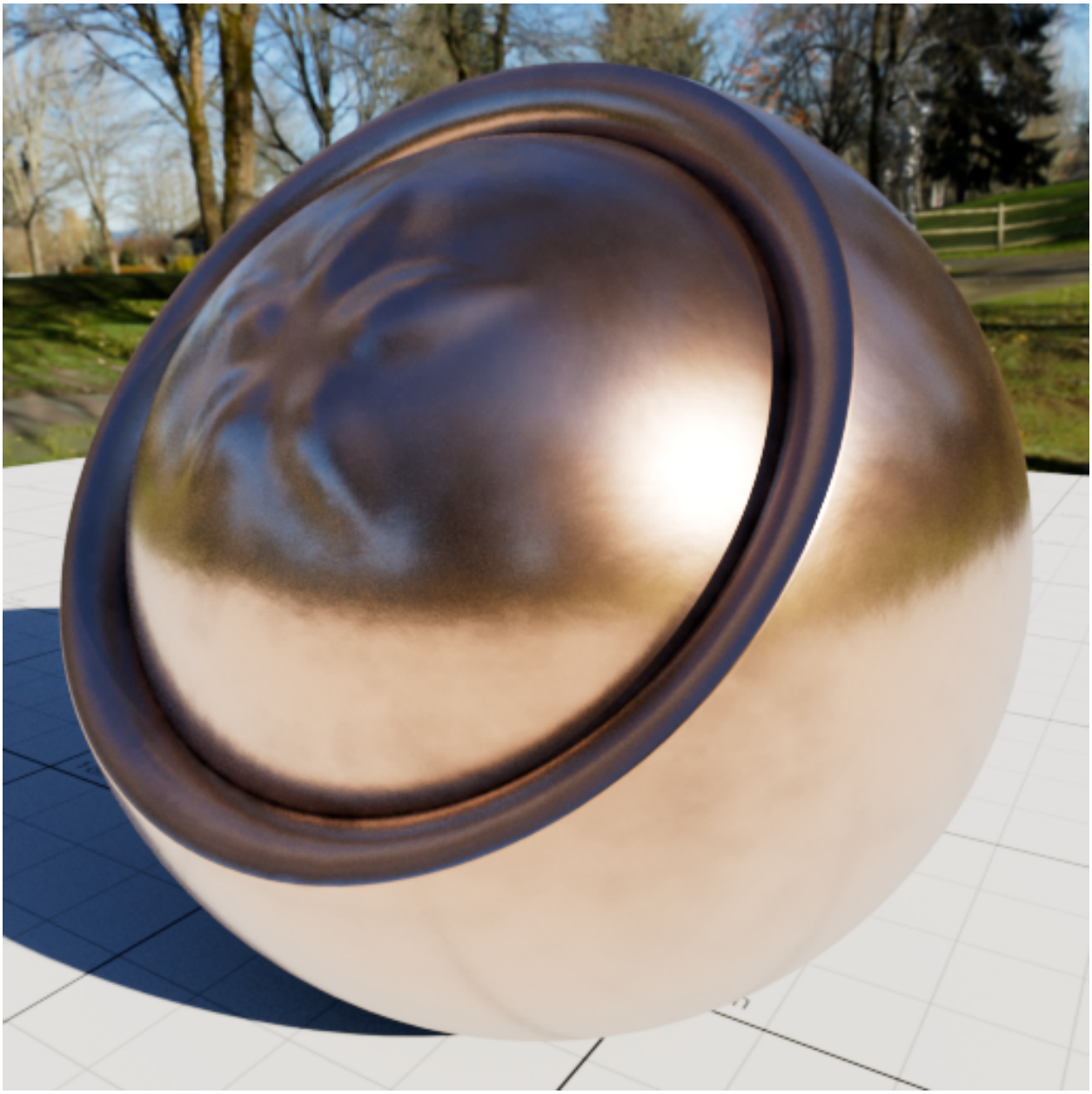
Output Color The metal BRDF output color.

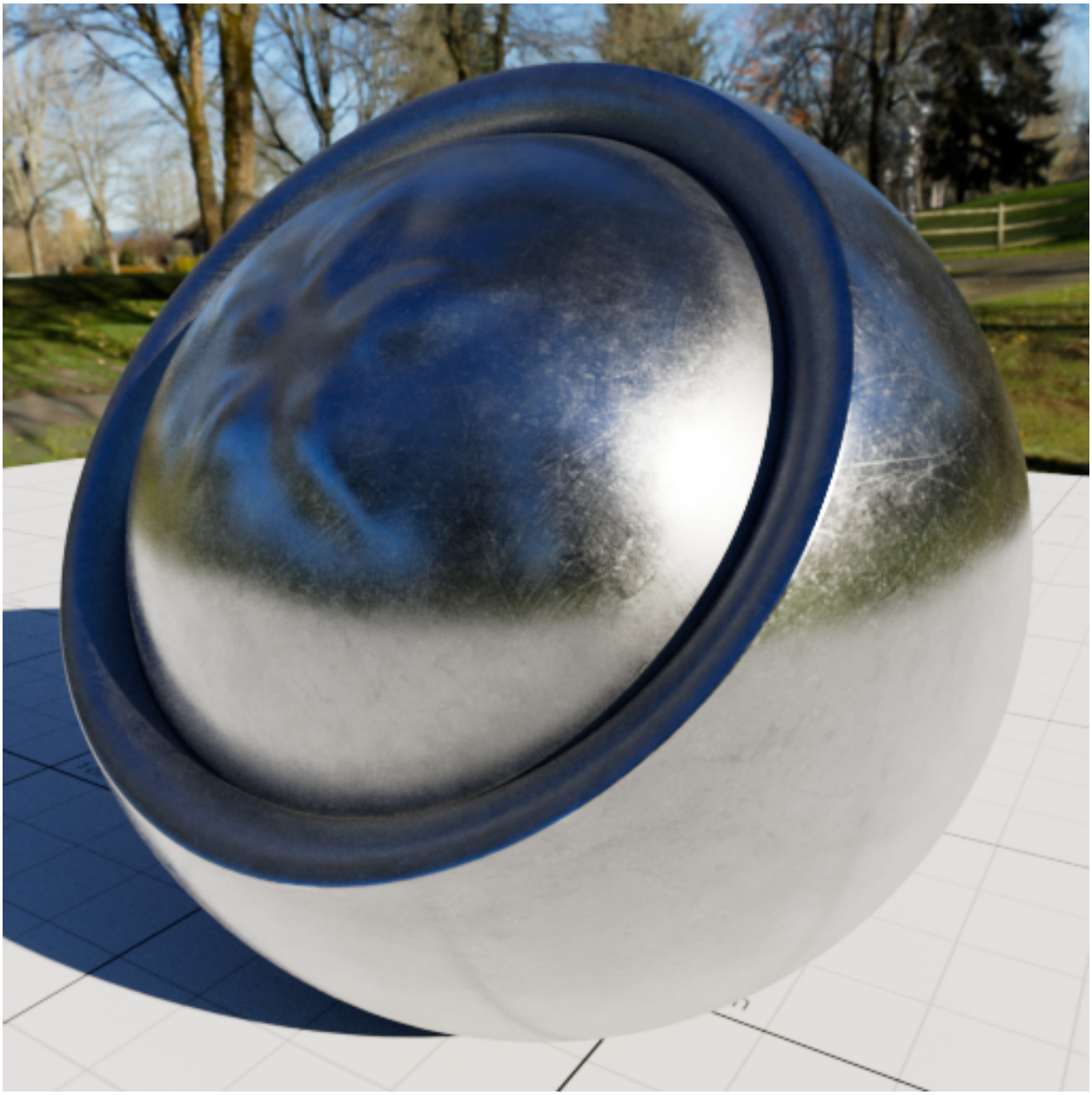
Output Matte Opacity The matte holdout.

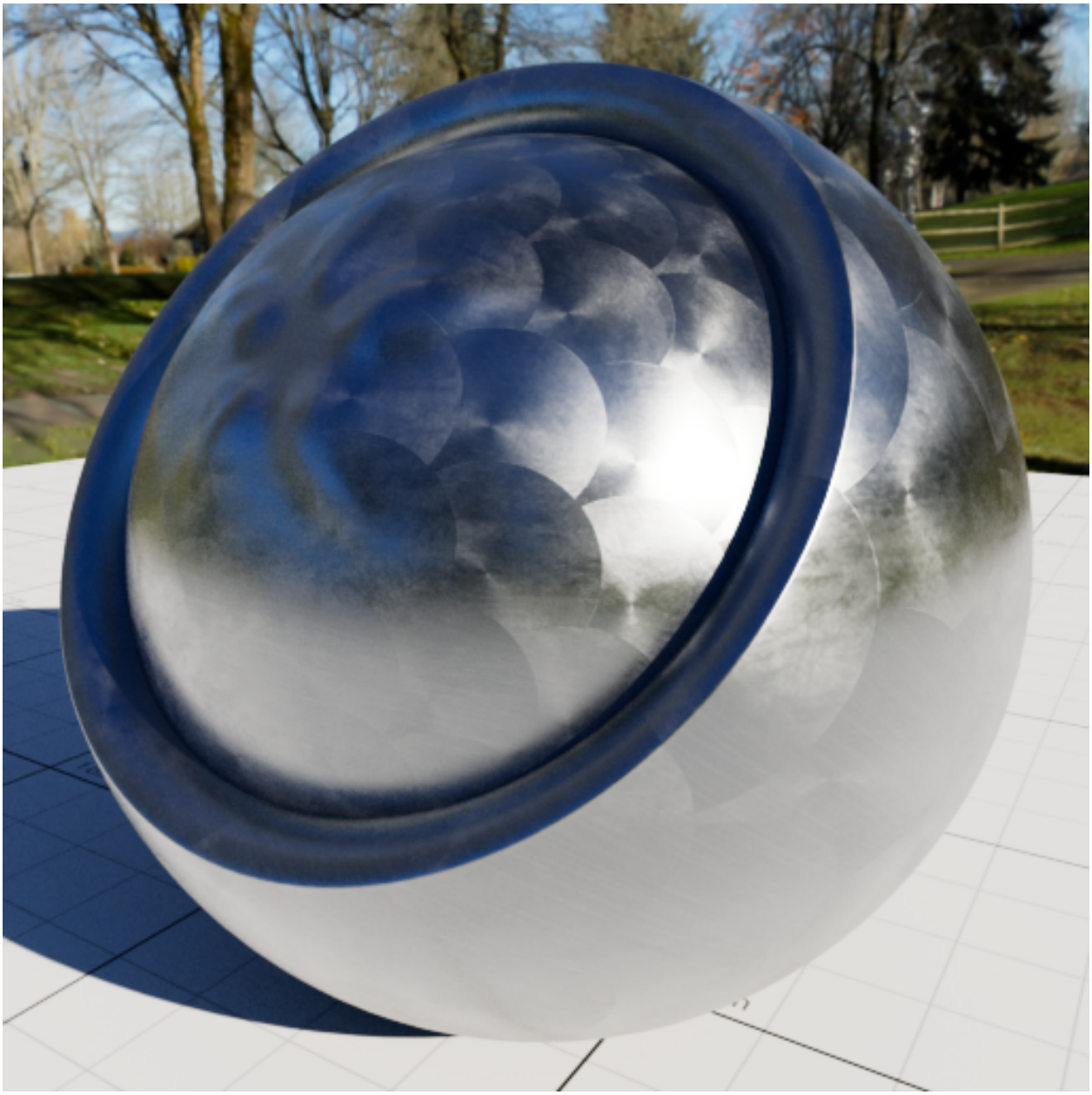
---

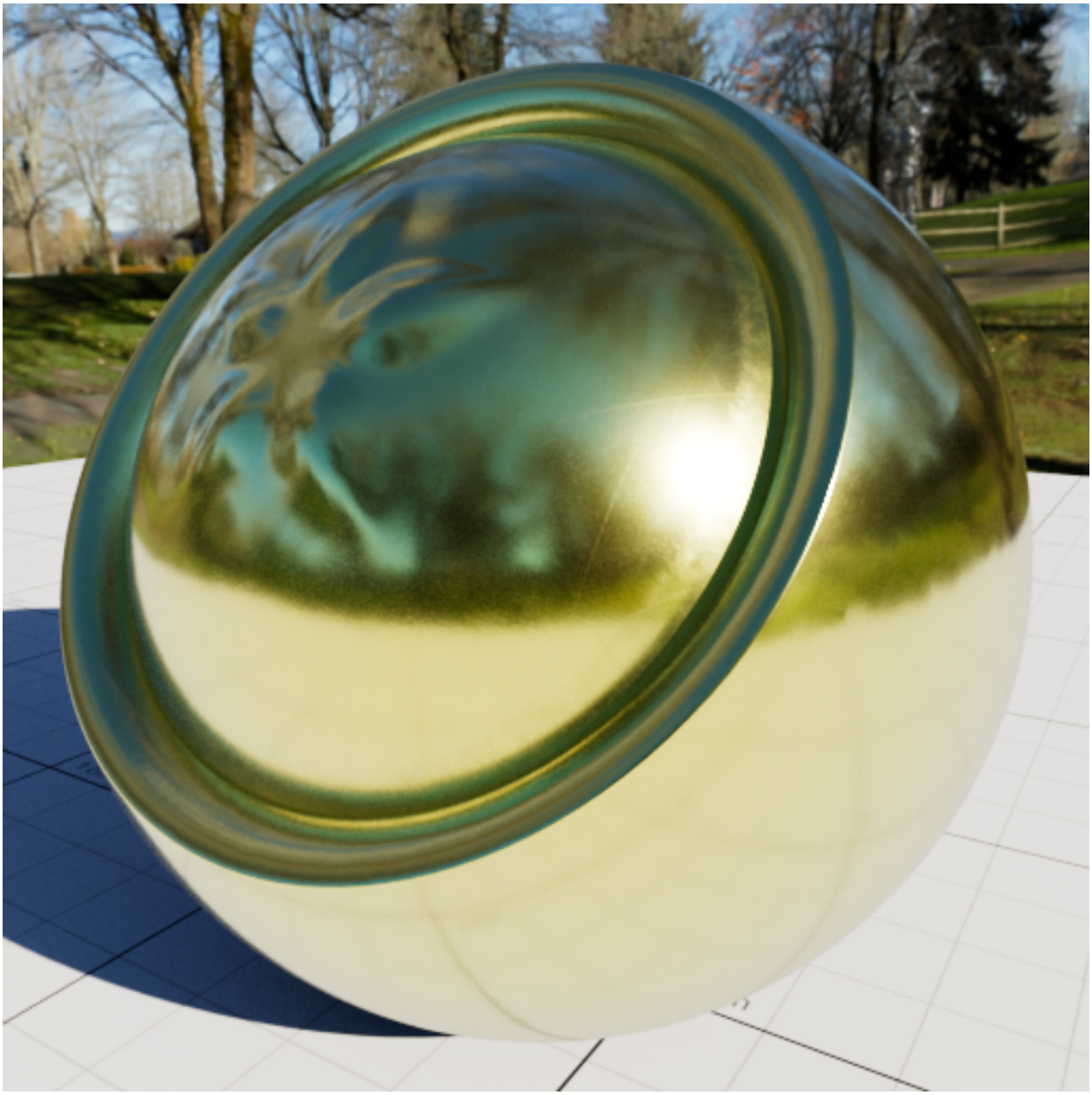
[2] Such as the *anisotropy vector field node*.

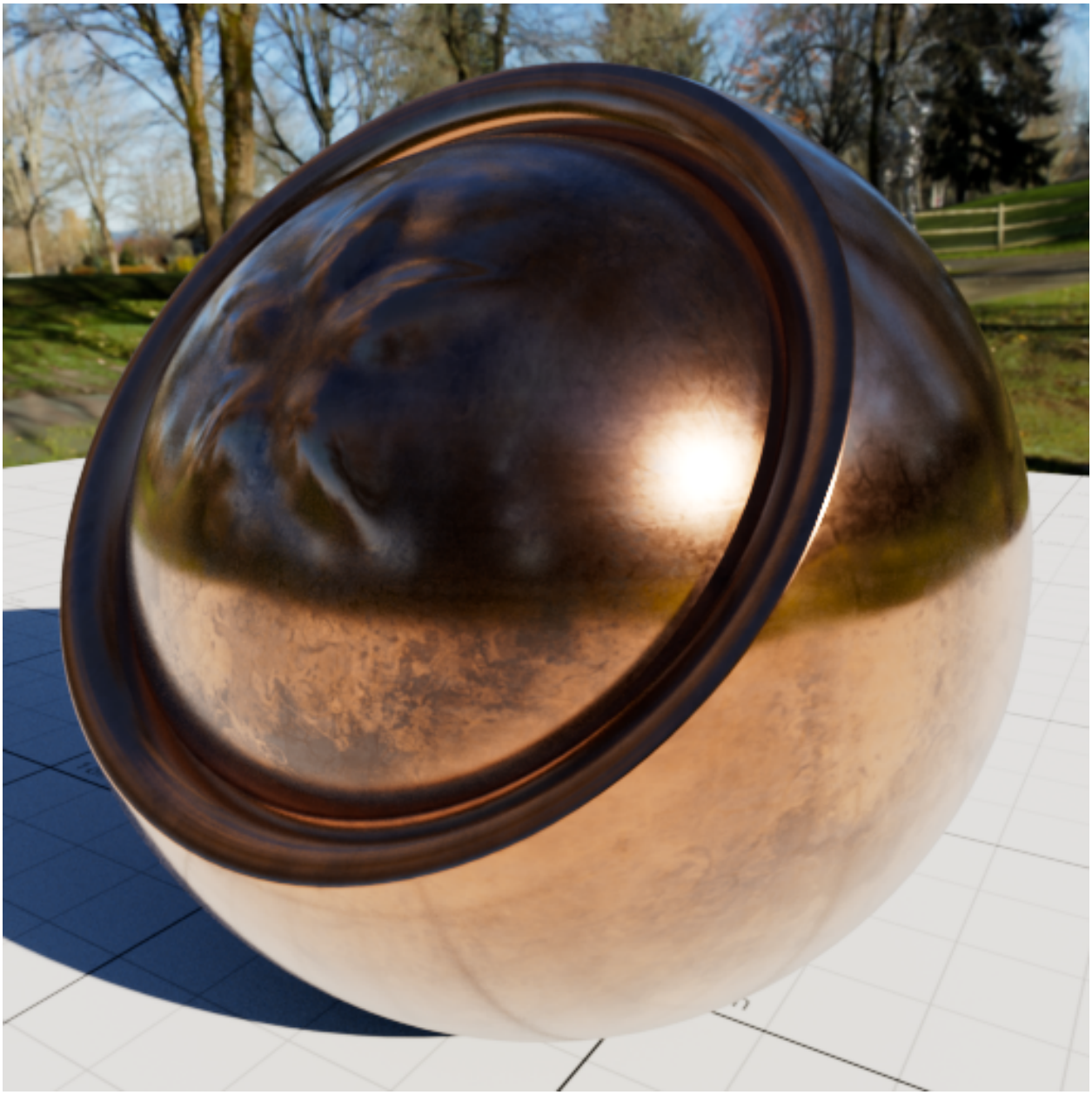
1.9.3 Screenshots

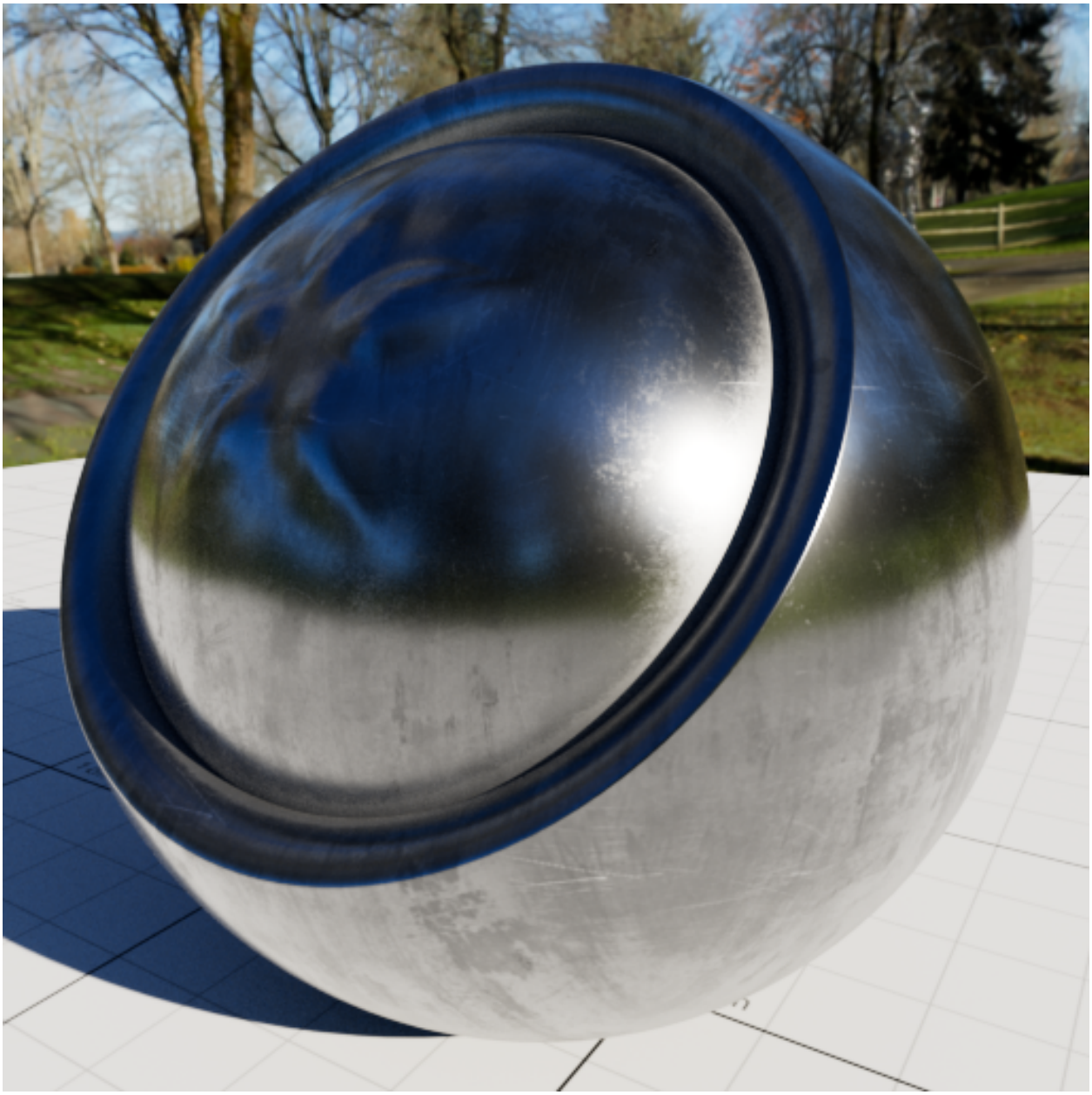


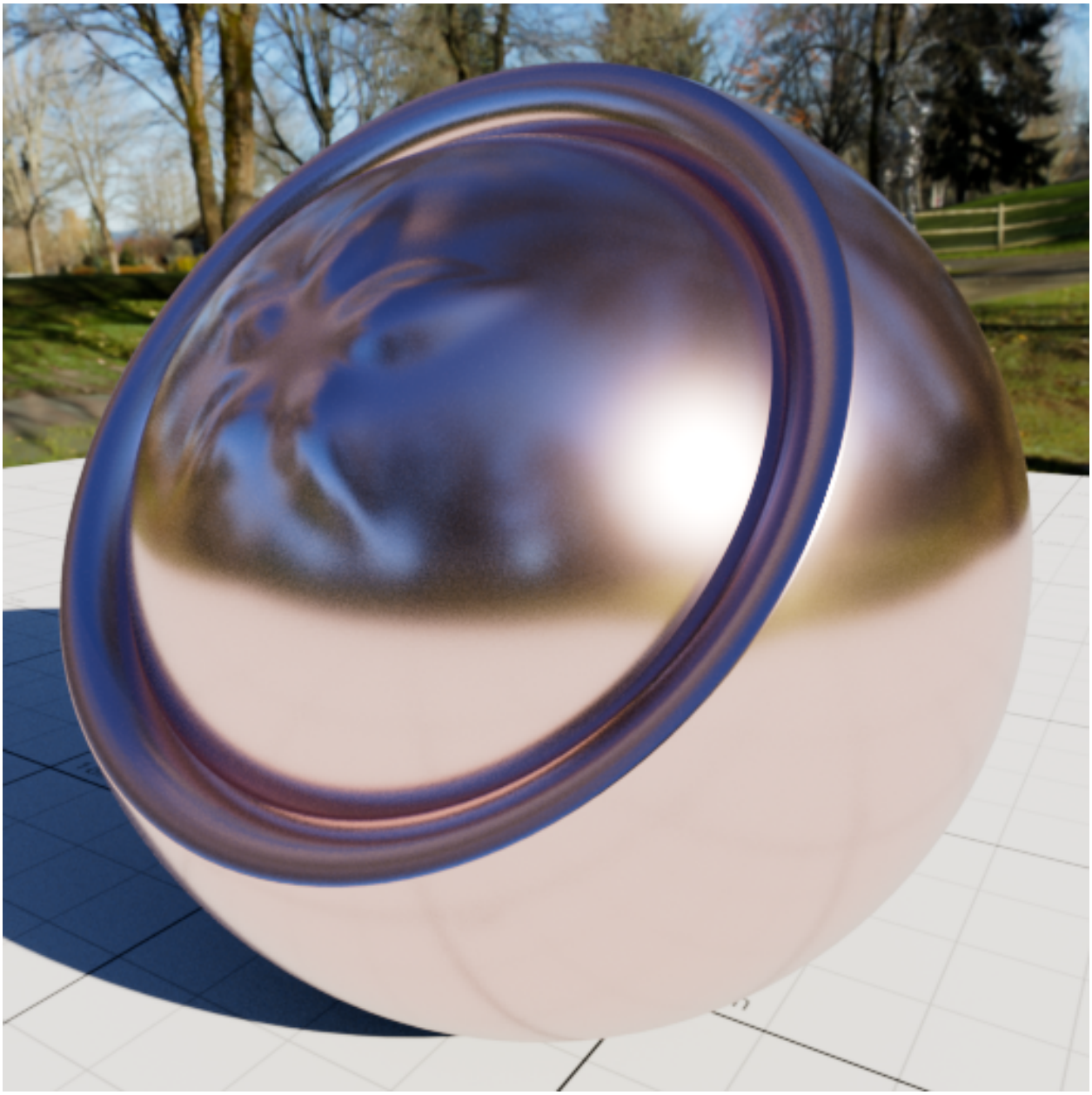


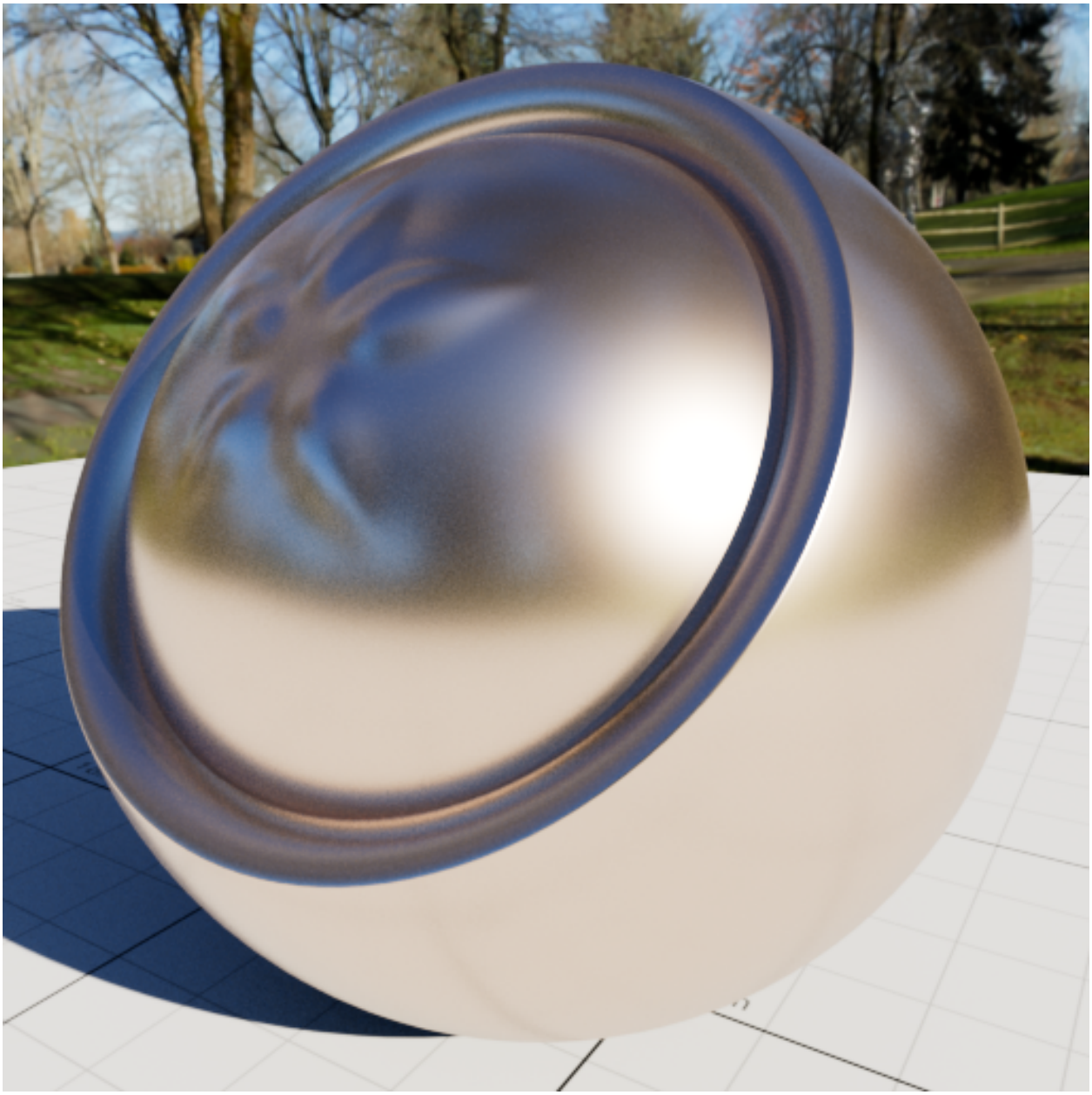


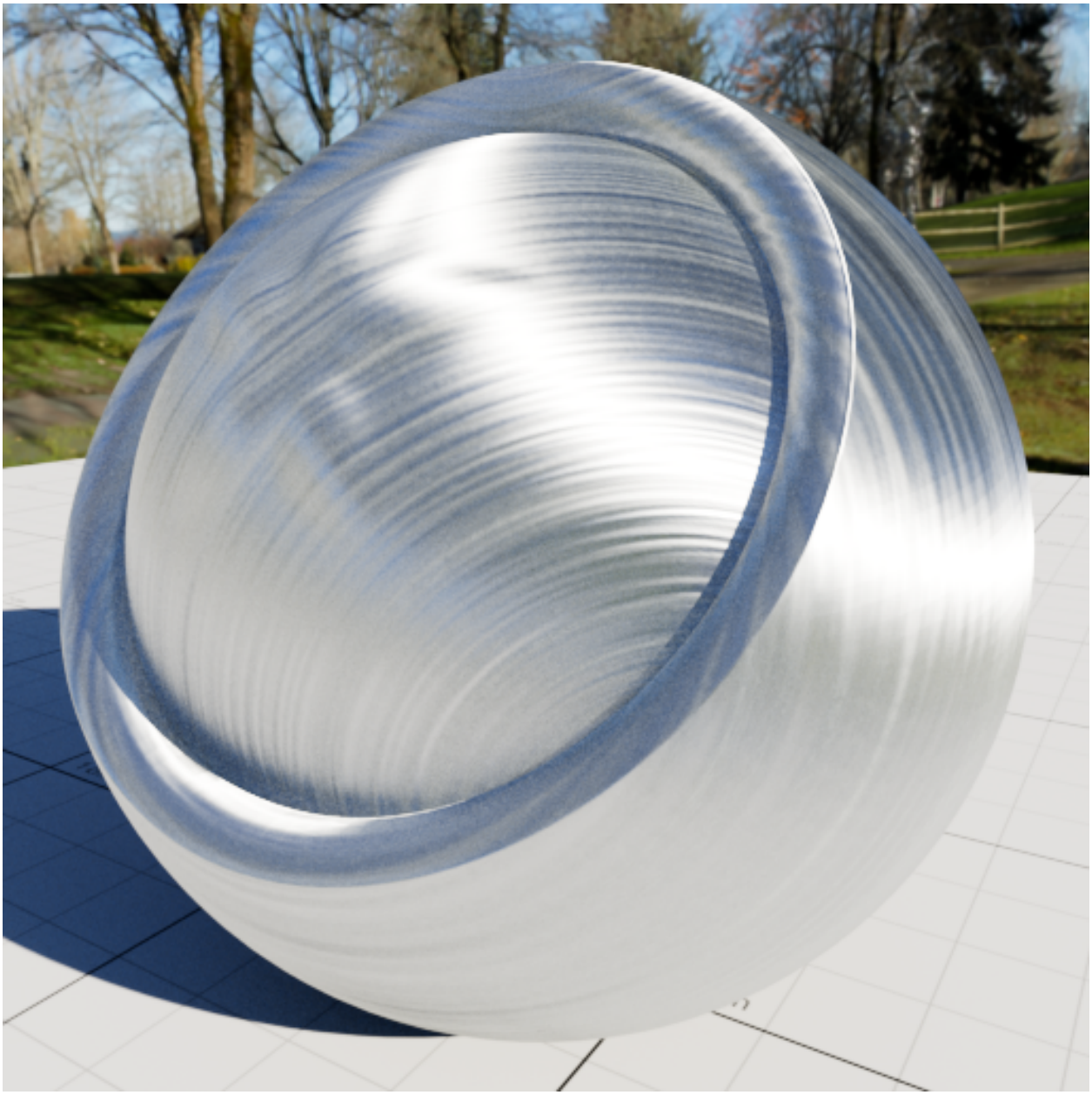


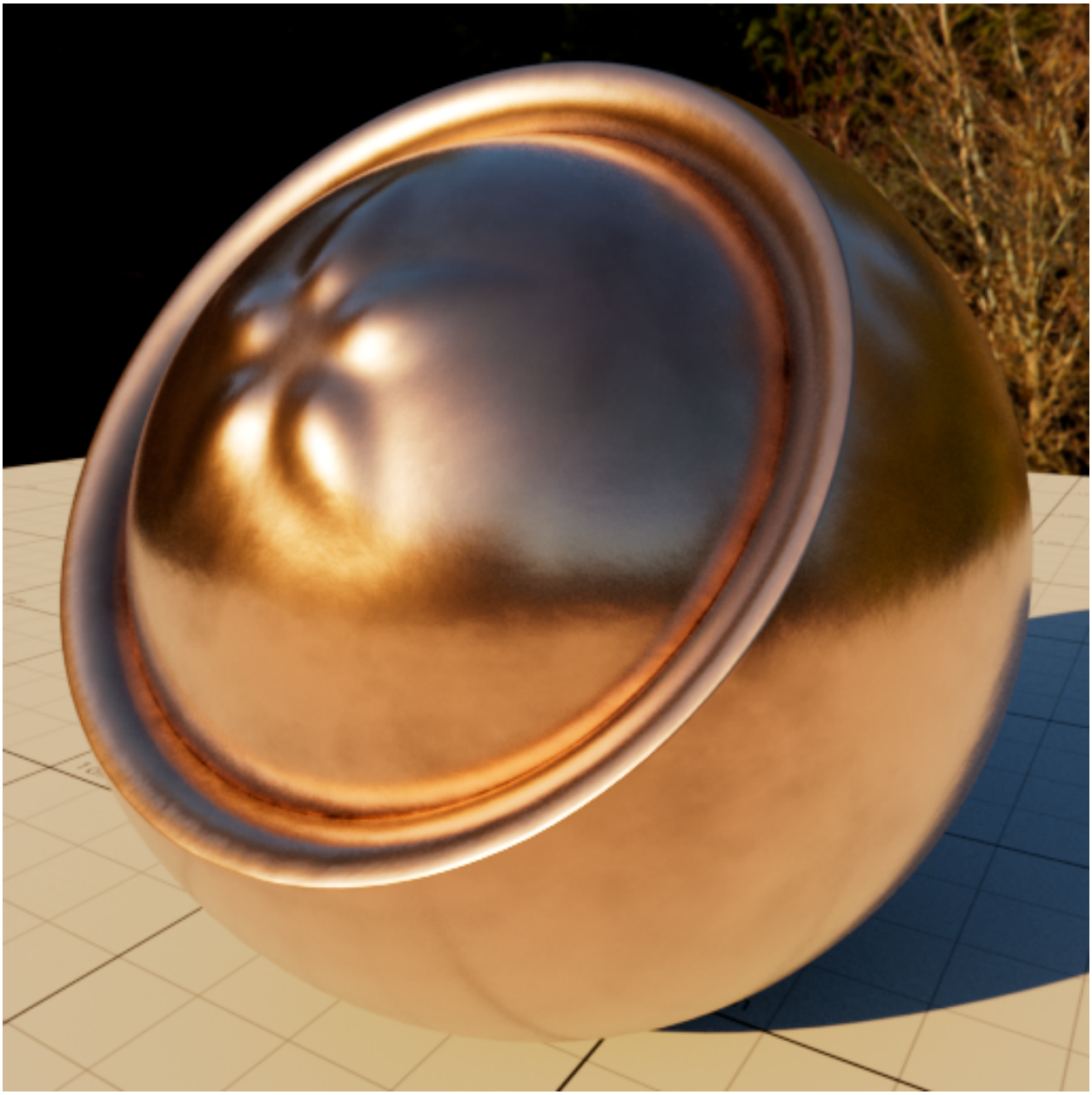


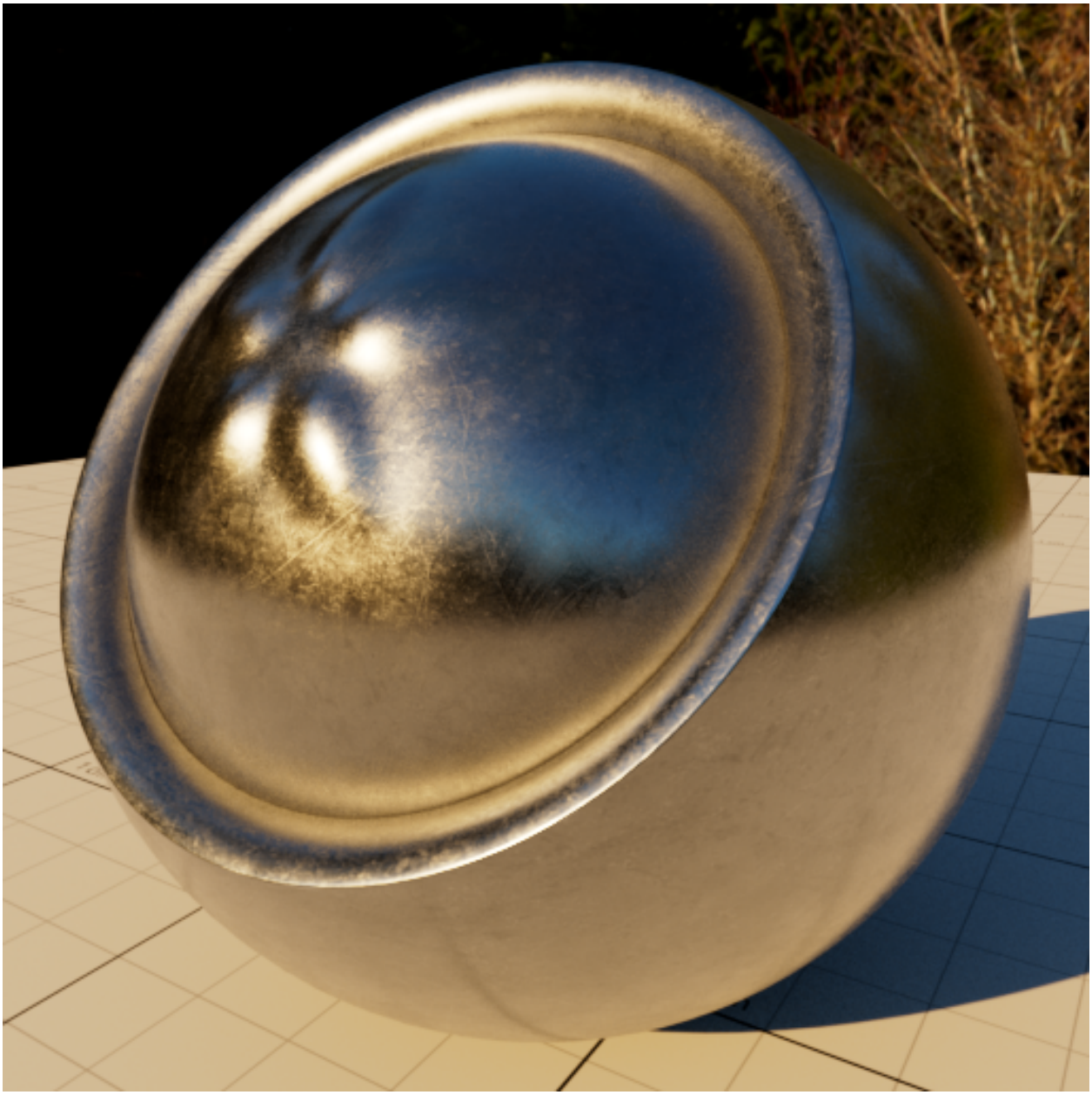


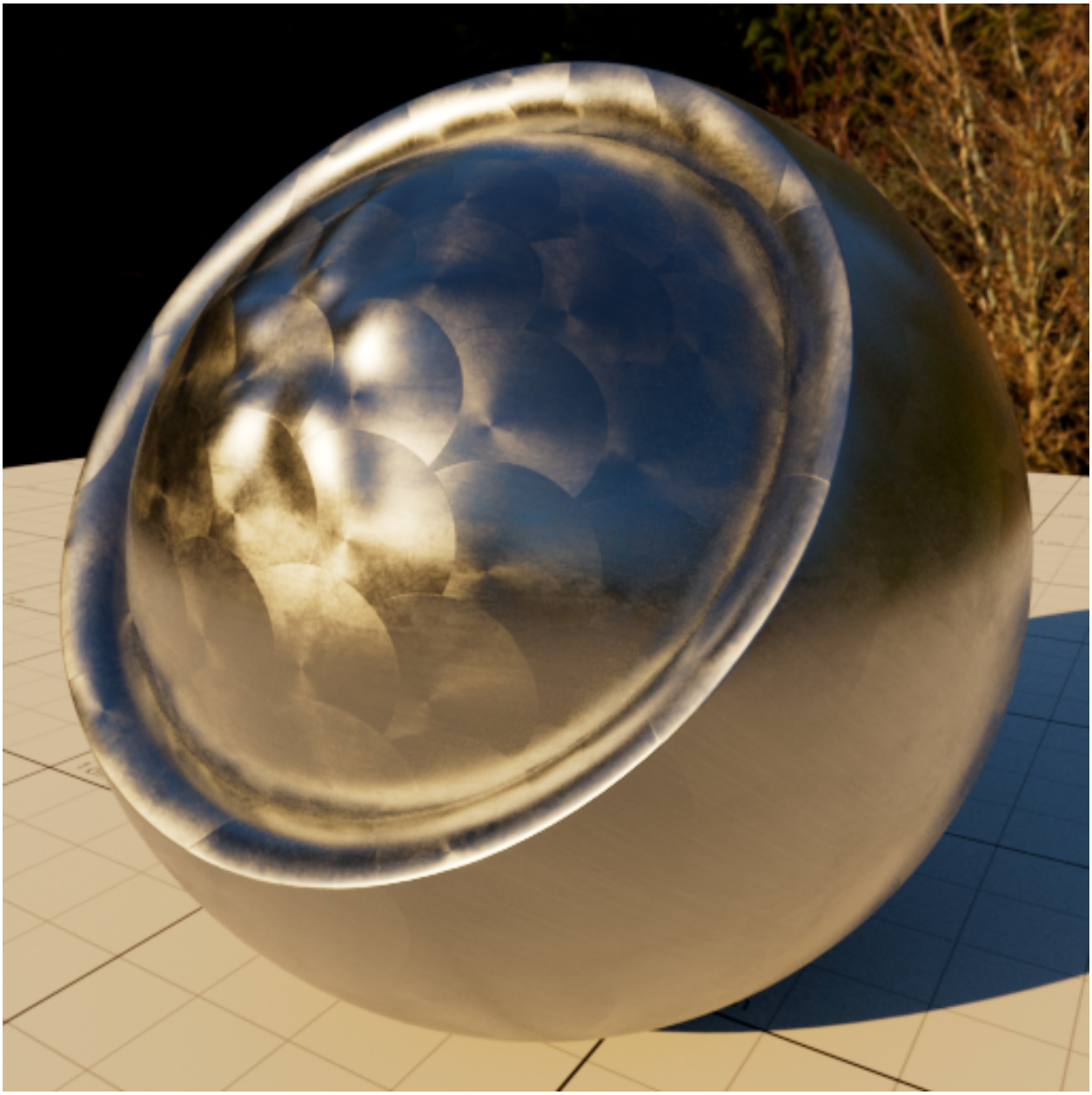


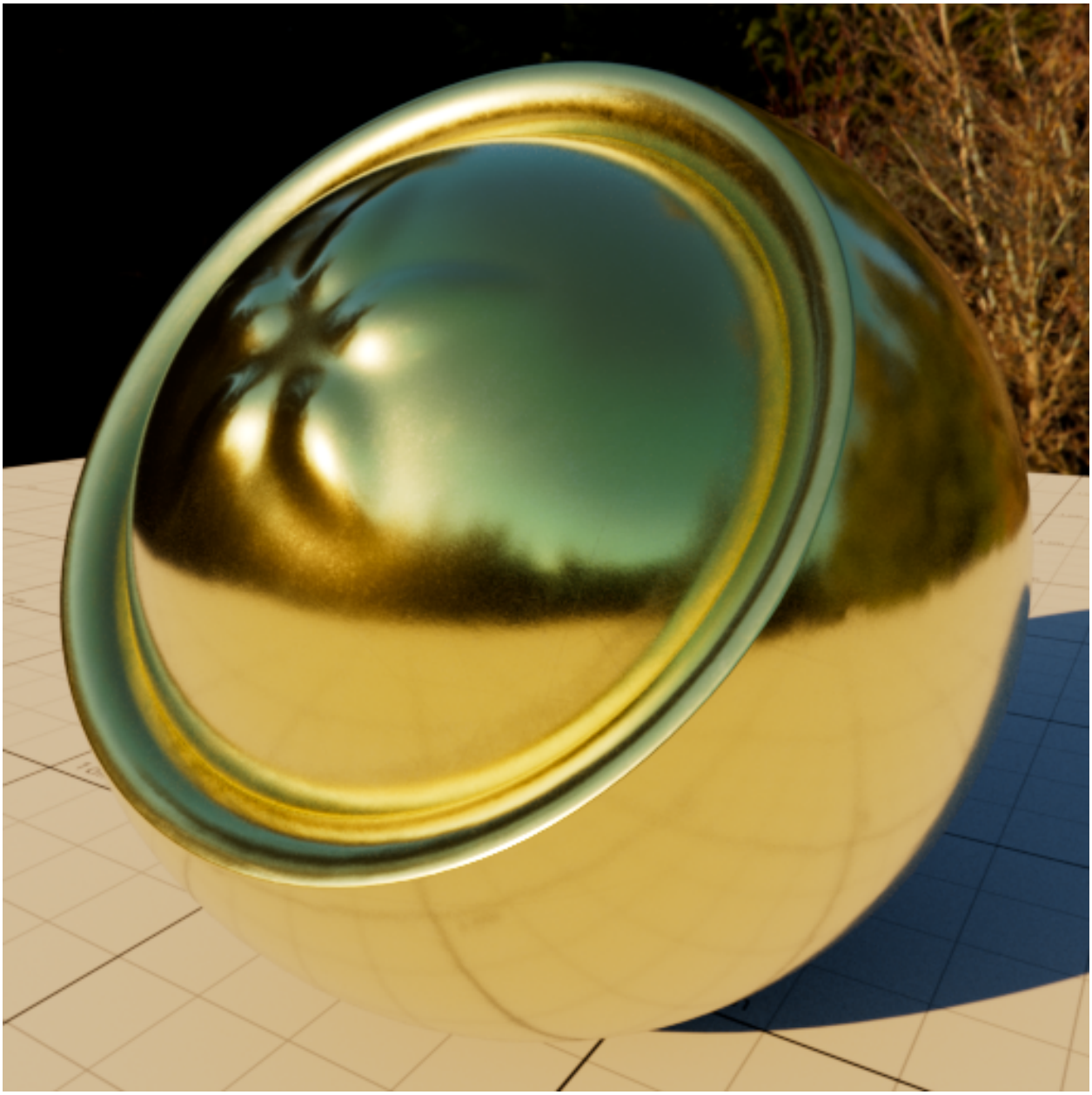


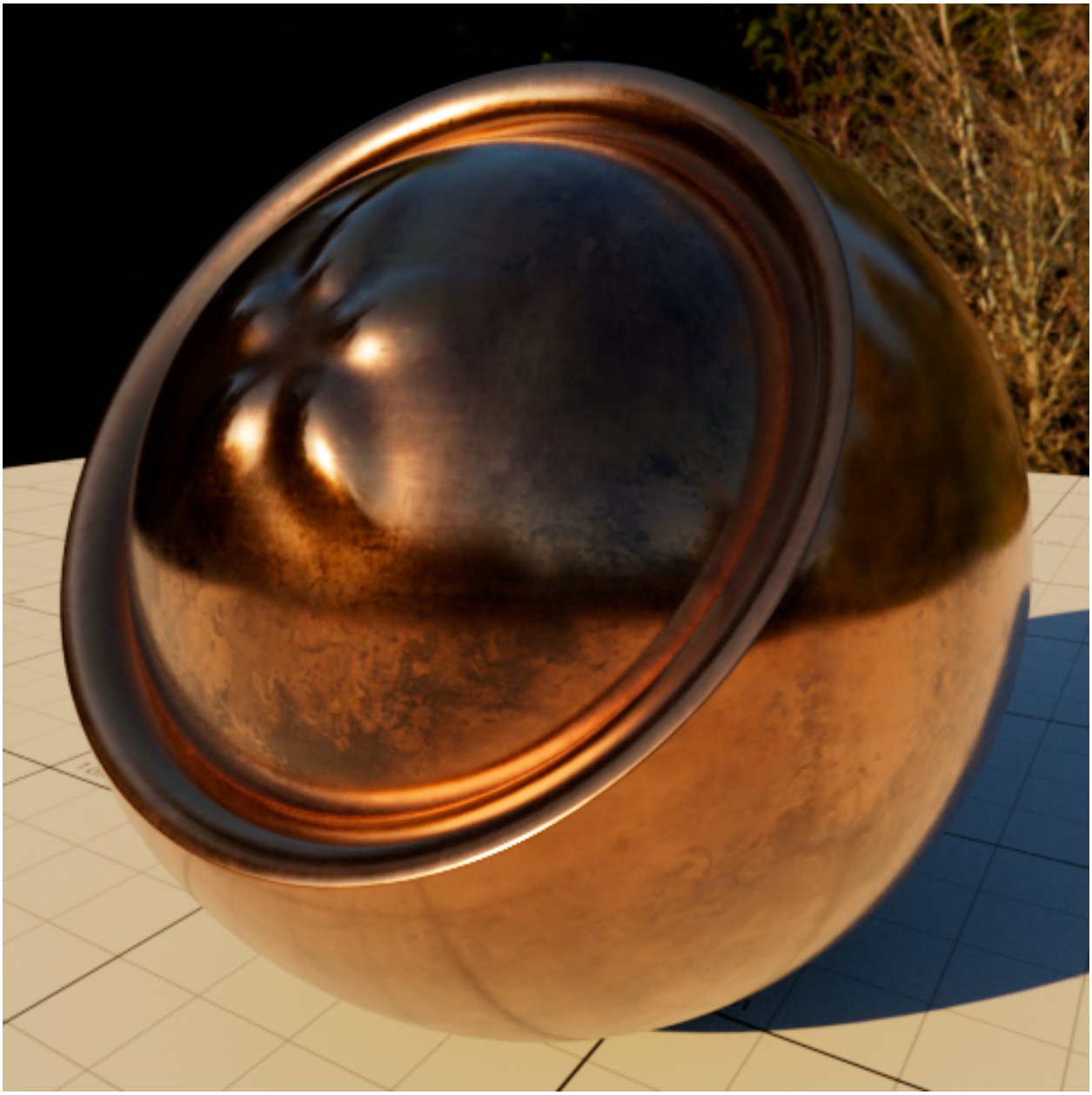


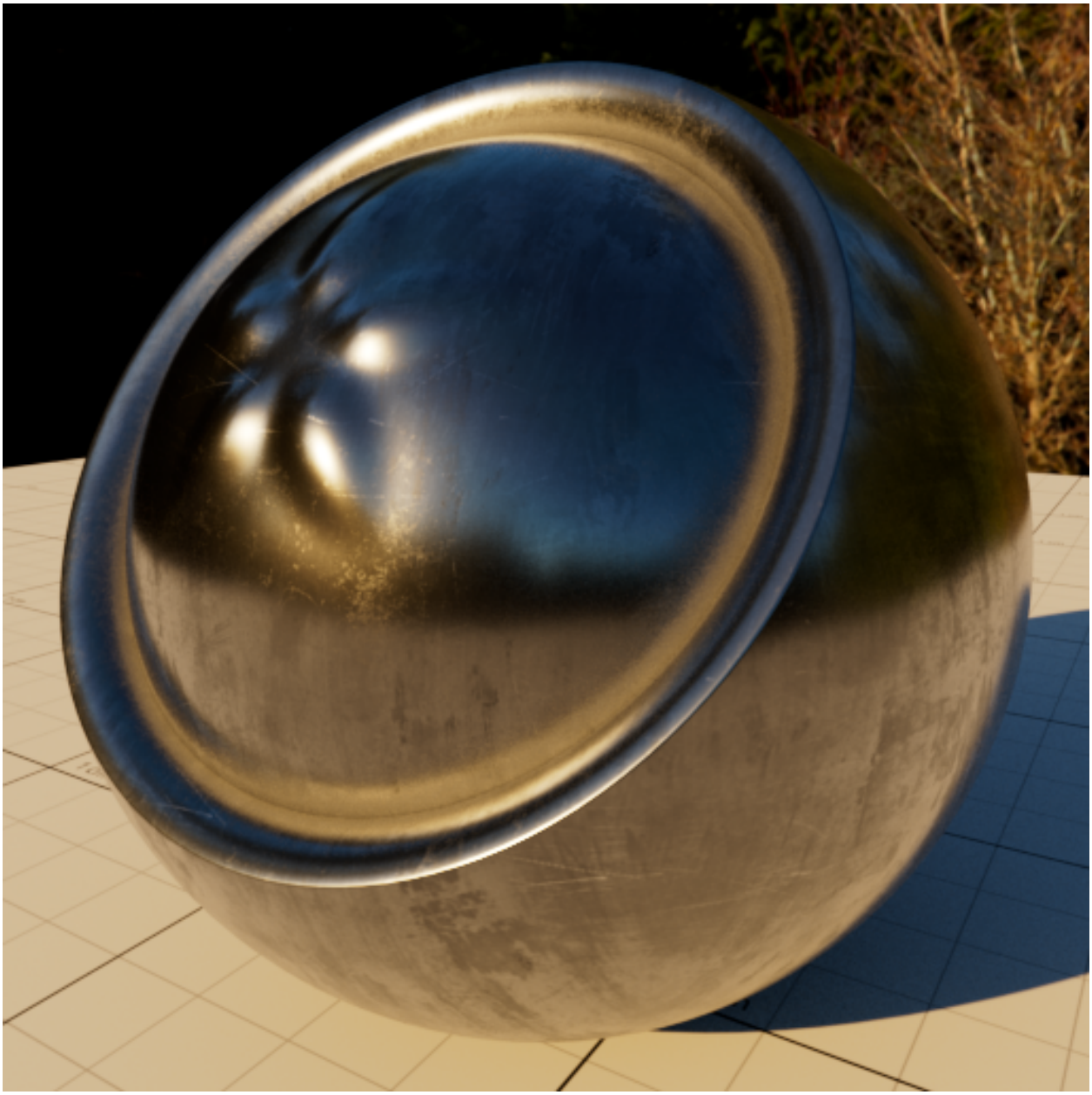


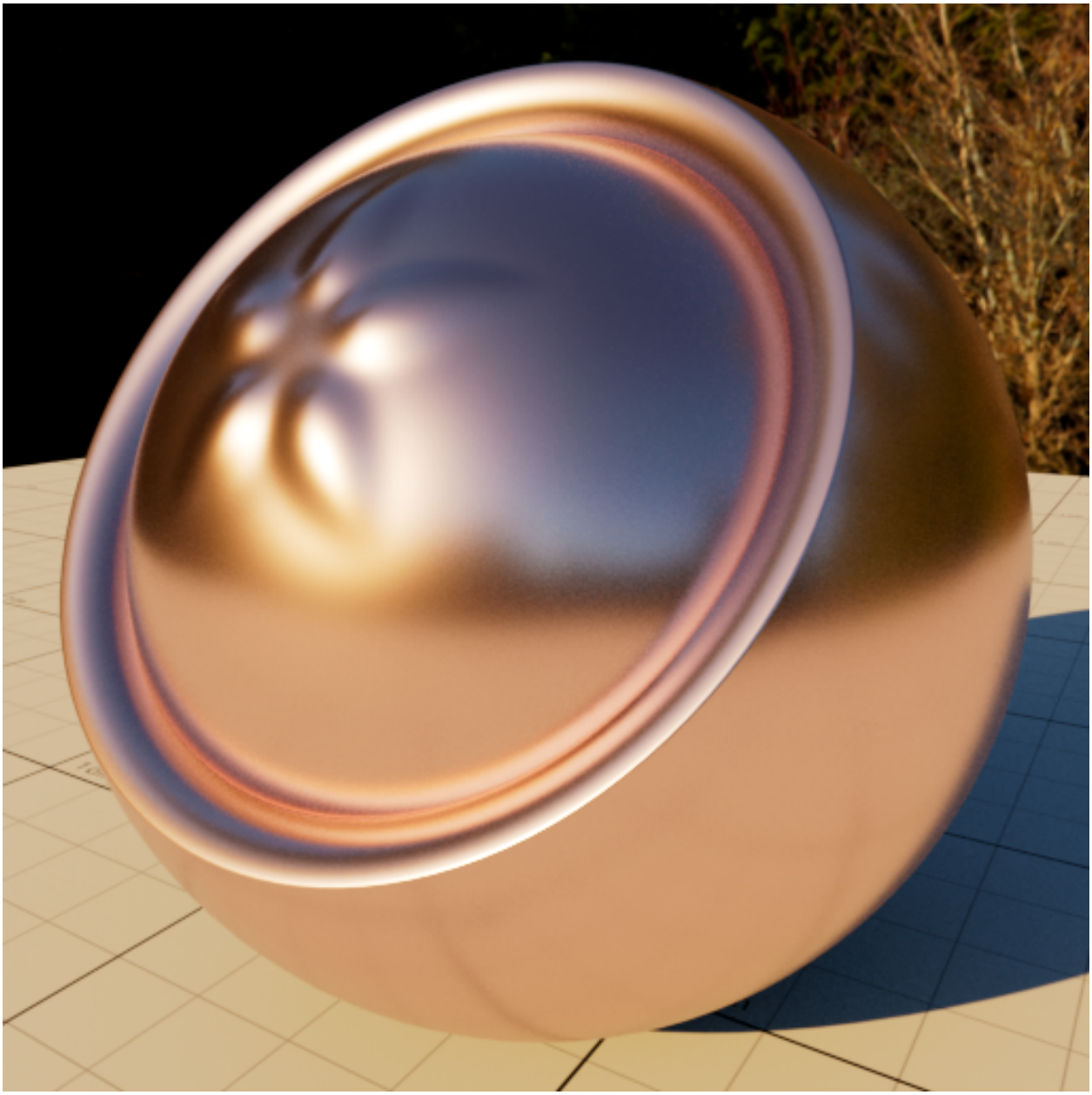


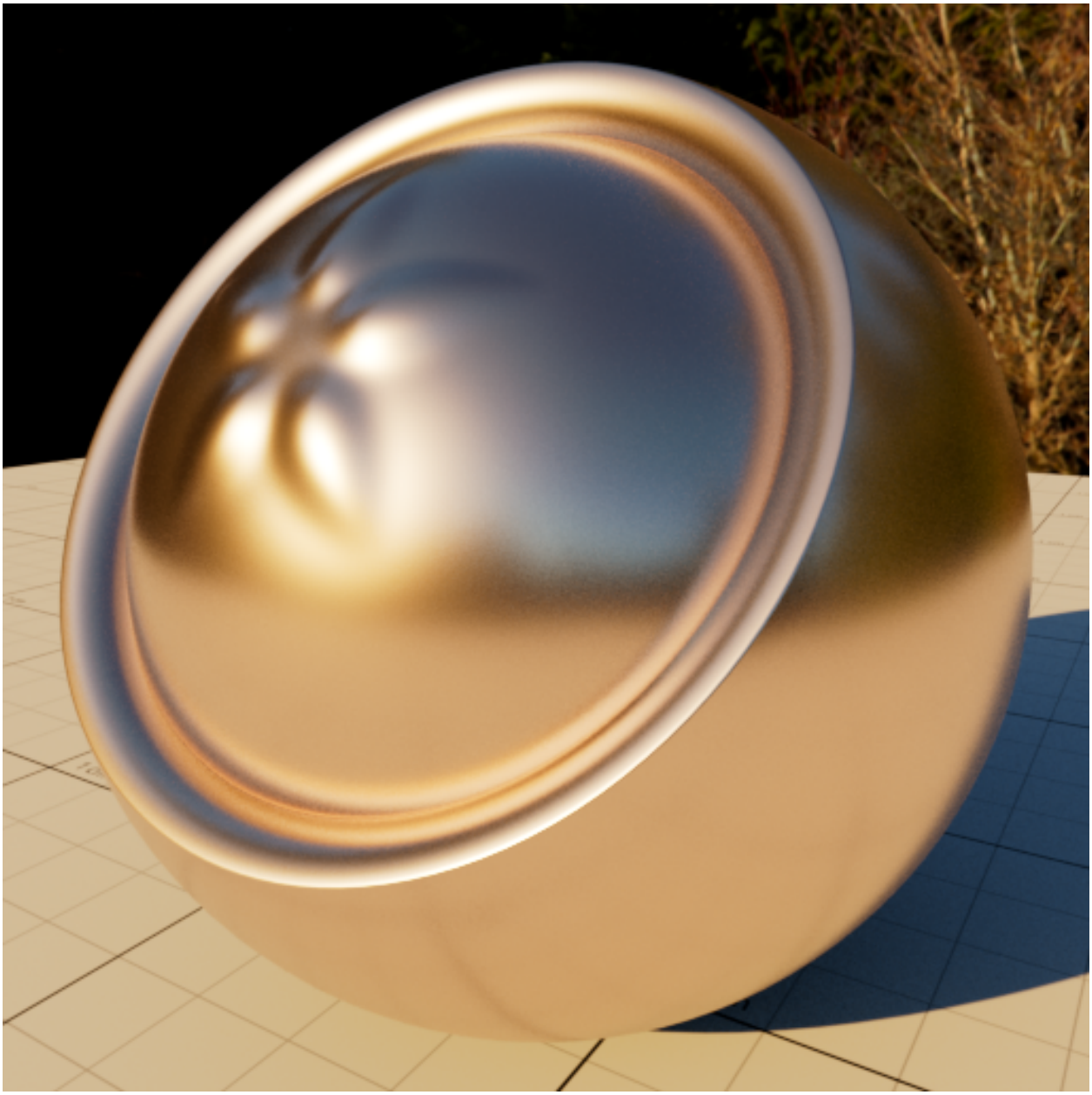


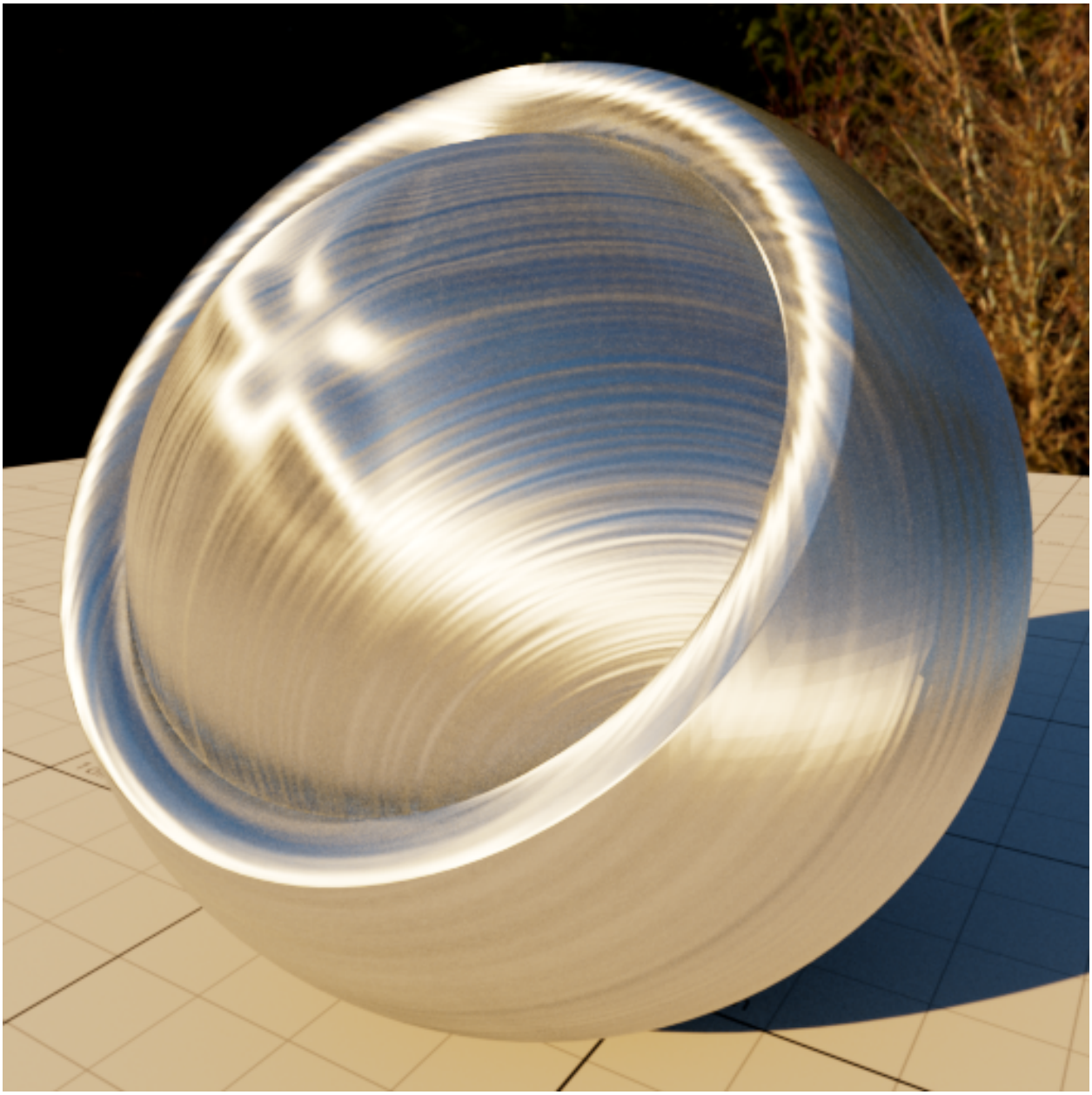




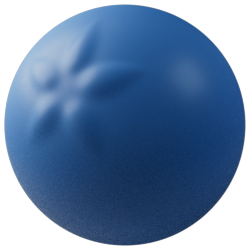








References



1.10 asPlastic

A physically correct plastic shader, stacking a specular term on top of a diffuse substrate.

1.10.1 Parameters

Diffuse Parameters

Diffuse Color The diffuse color.

Diffuse Weight A scaling factor for the diffuse BRDF.

Scattering The scattering amount for the diffuse term. Full scattering will take into account all the events in the specular term, while a value of 0 will disable them.

Specular Parameters

Specular Color The specular color.

Specular Weight A scaling factor for the specular BRDF.

Index of Refraction The index of refraction for the (dielectric[1]) Fresnel term.

Distribution The microfacet distribution function[2] to use on the specular BRDF. The Beckmann *[BS63] [CT82]* distribution produces the sharpest highlights. The GGX *[WMLT07]* distribution produces a slight veil like effect that softens the specular highlights a bit, and the GTR *[MHH+12]* distribution accentuates this even further. The Student's t-MDF *[RBMS17]* allows the user to control this softening - the *tails* of the specular highlights - via a second parameter *Specular Spread*, which as the name suggests, controls the spreading of the specular highlights. The *Distribution* parameter can then take the following values

- Beckmann
- GGX
- GTR

[1] Dielectric is a material which is an electric insulator, the opposite of *conductors* which as the name says, conducts electricity. See this page on dielectric materials for more details. In terms of look development an accepted simplification is that dielectrics have white or non-tinted specular highlights, while conductors have tinted or coloured specular highlights.

[2] The microfacet distribution function is a function that describes statistically the microscopic shape of the surface's as a distribution of microfacet orientations. See the this page on the normal distribution function (NDF), and this page on specular highlights for more details.

- Student's t-MDF

Specular Roughness The apparent surface roughness, with a value near 0 producing sharp highlights, and a value of 1.0 producing soft diffuse like highlights.

Note: As specular roughness values increase, there is energy loss due to the lack of multiple scattering *[HHdEonD16]*. Appleseed however does compensate for this energy loss, but only for the *Beckmann* and *GGX* distributions. There is no compensation applied on the *GTR* nor Student's t-MDF.

Specular Spread This parameter controls the *tails* of the specular highlights of the Student's t-MDF, softening it, and creating the appearance of a slight veil or *haze*. It has no effect on the other microfacet distribution functions.

Bump Parameters

Bump Normal The unit length world space normal of the bumped surface.

Matte Opacity Parameters

Enable Matte Opacity Parameter that toggles matte holdouts.

Matte Opacity Matte opacity scaling factor.

Matte Opacity Color Holdout color.

Advanced Parameters

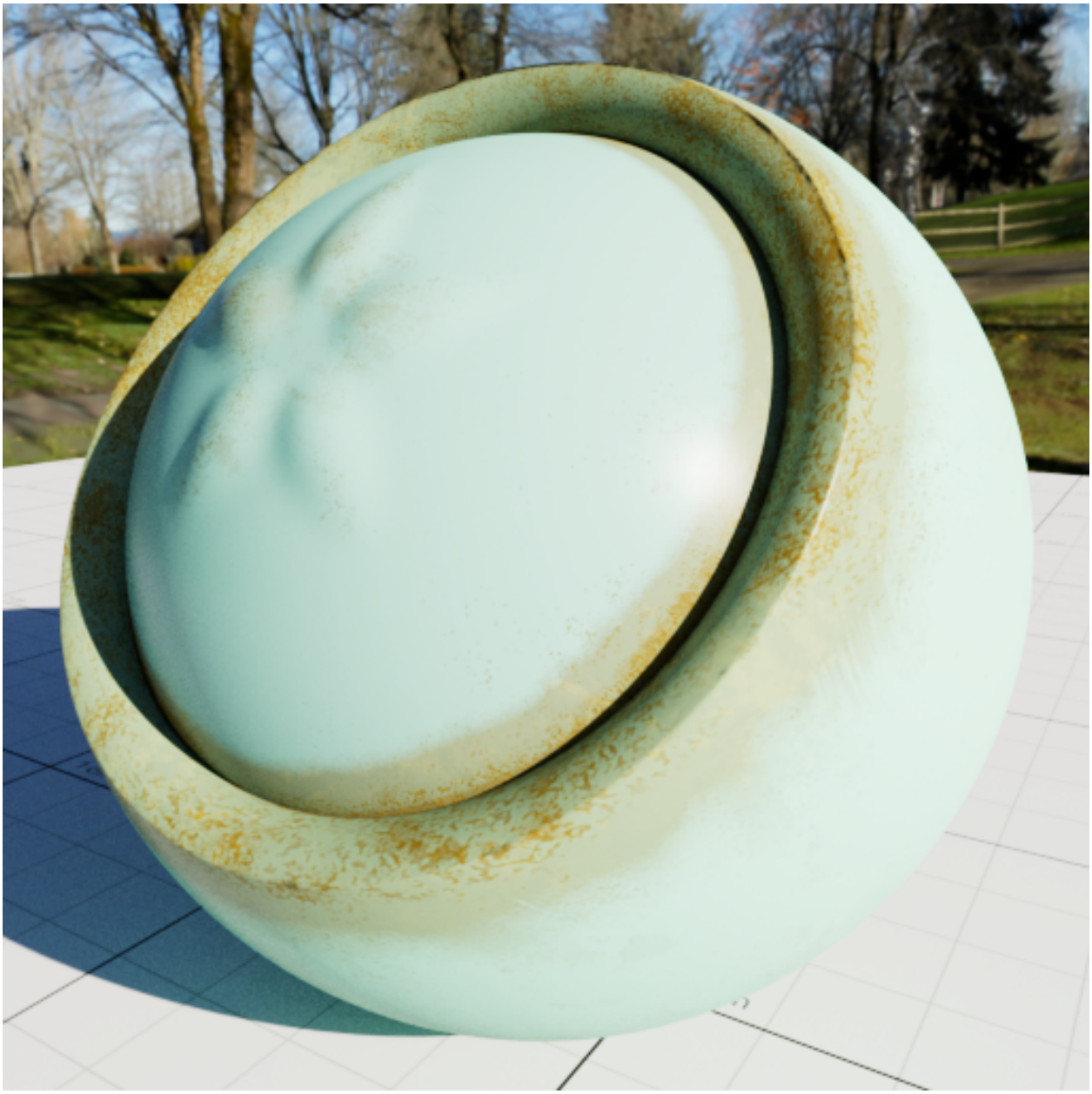
Ray Depth The maximum ray depth a ray is allowed to bounce before being terminated.

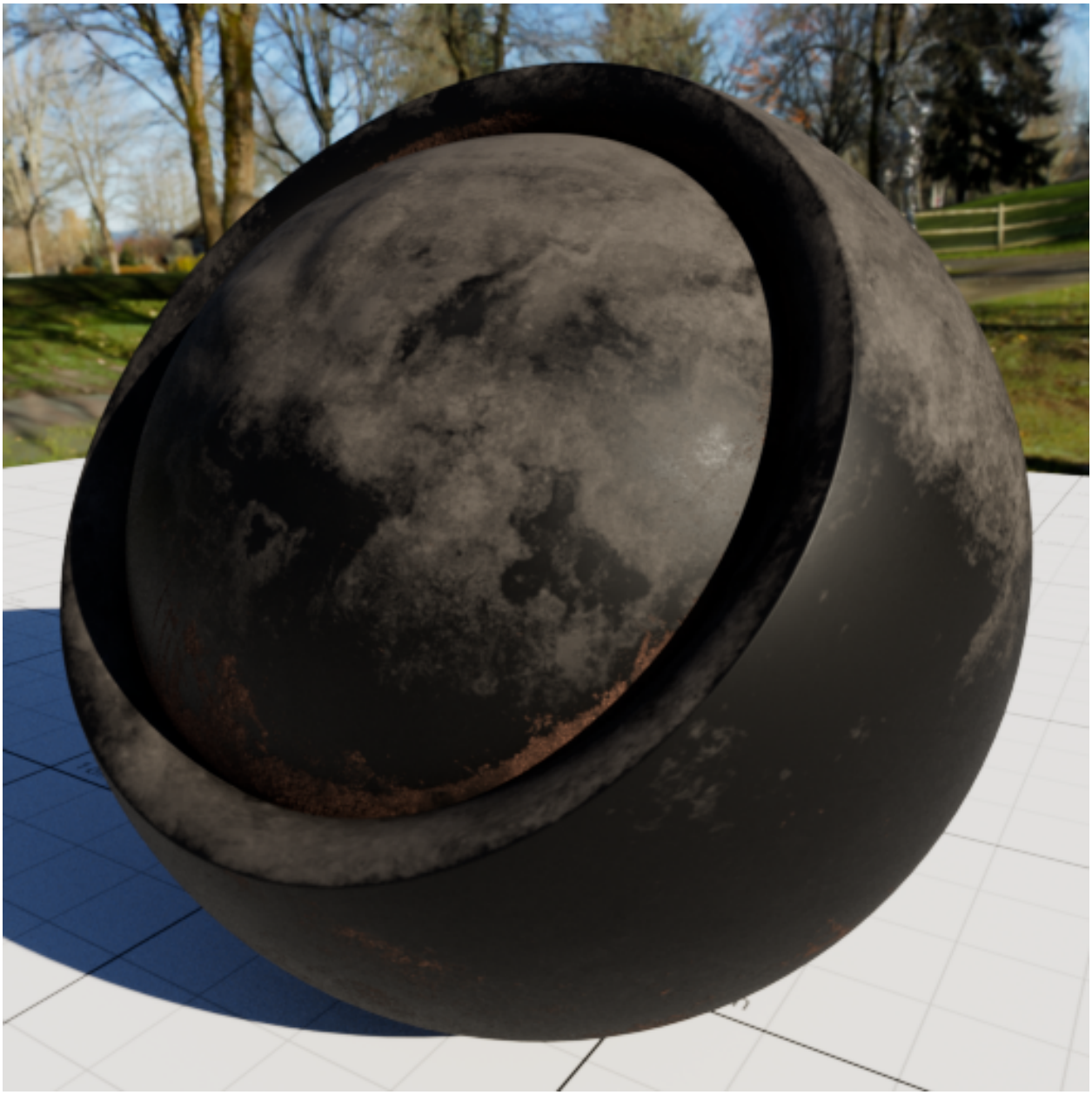
1.10.2 Outputs

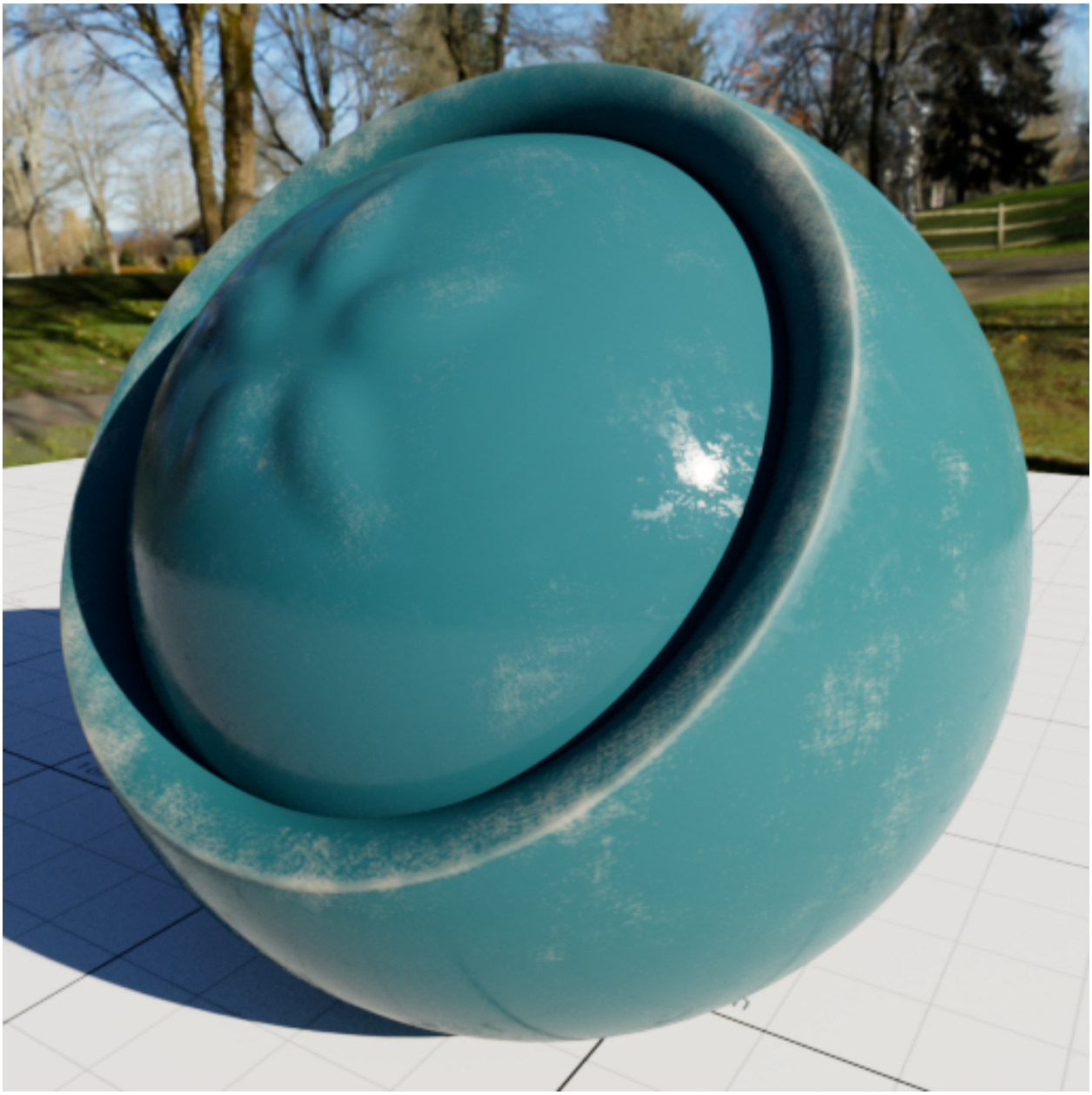
Output Color The plastic BRDF output color.

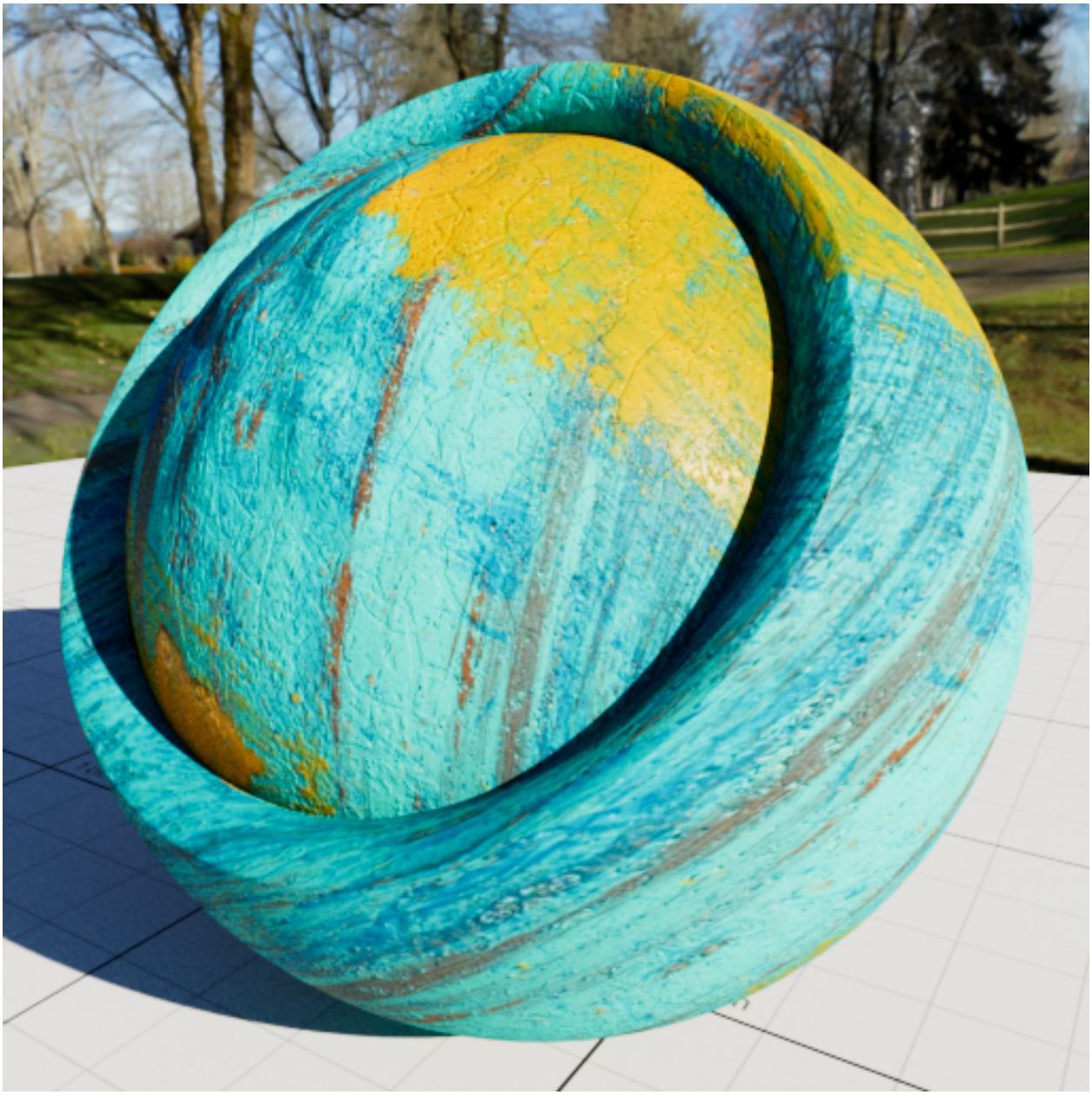
Output Matte Opacity The matte holdout.

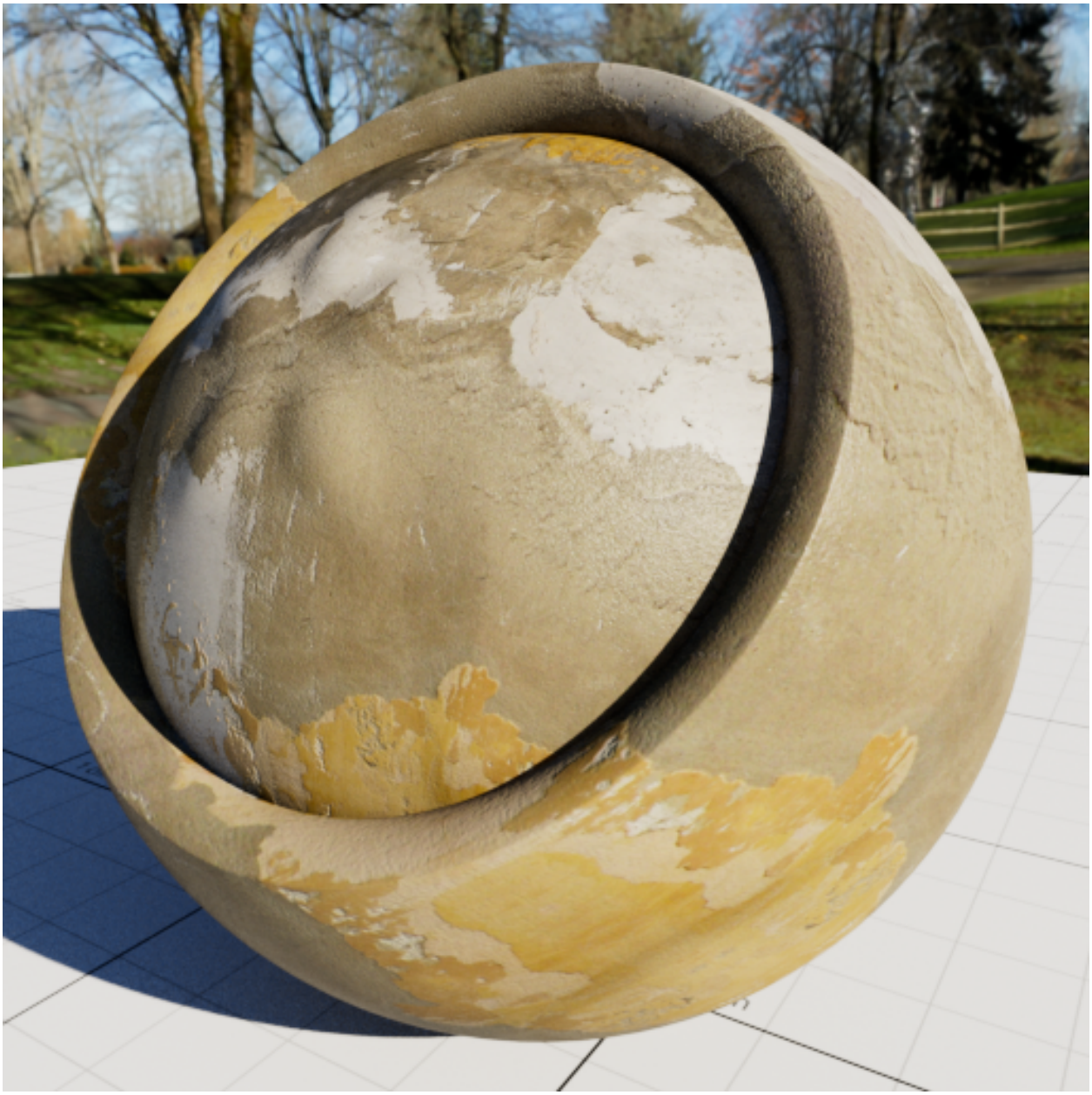
1.10.3 Screenshots

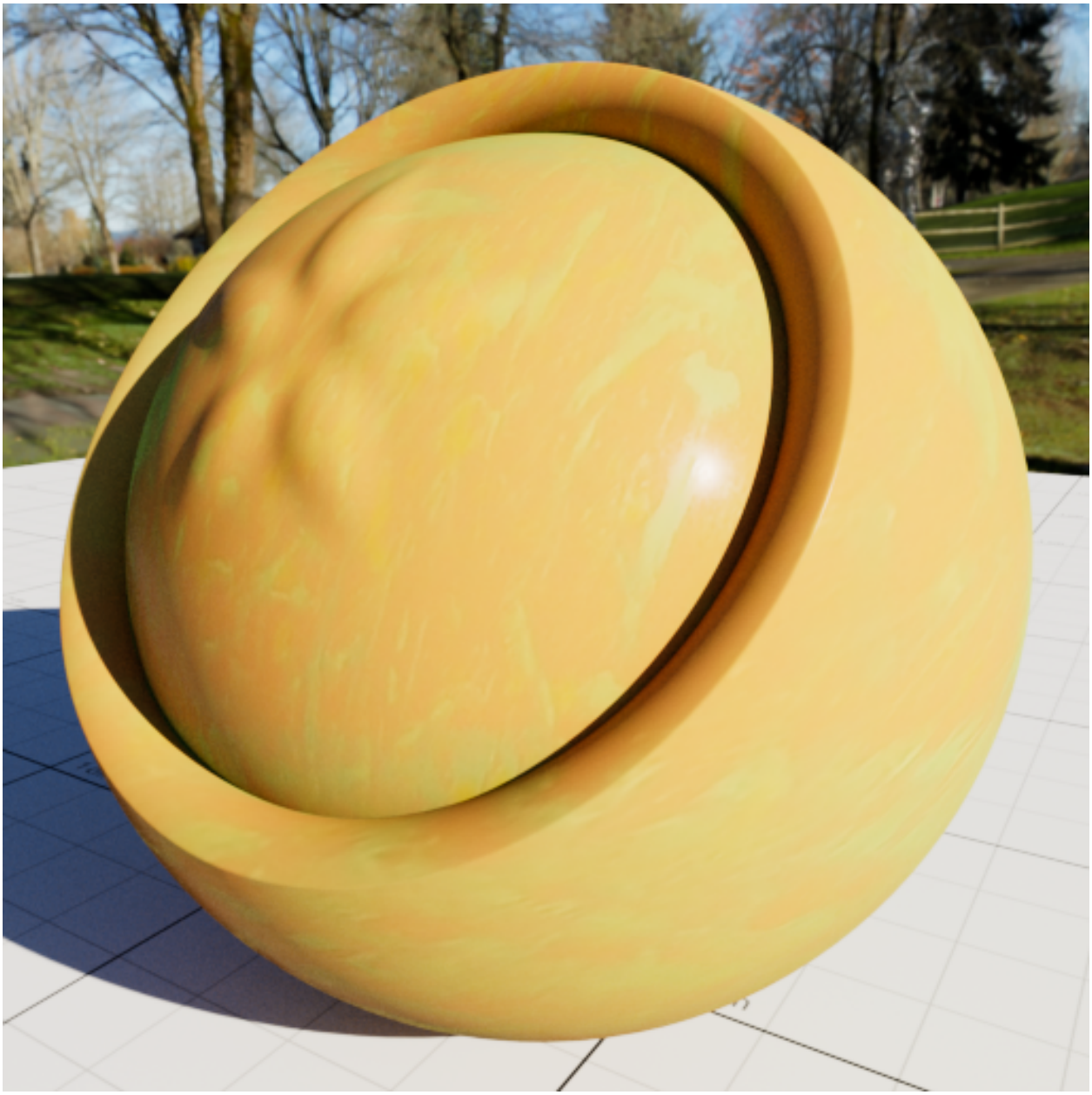


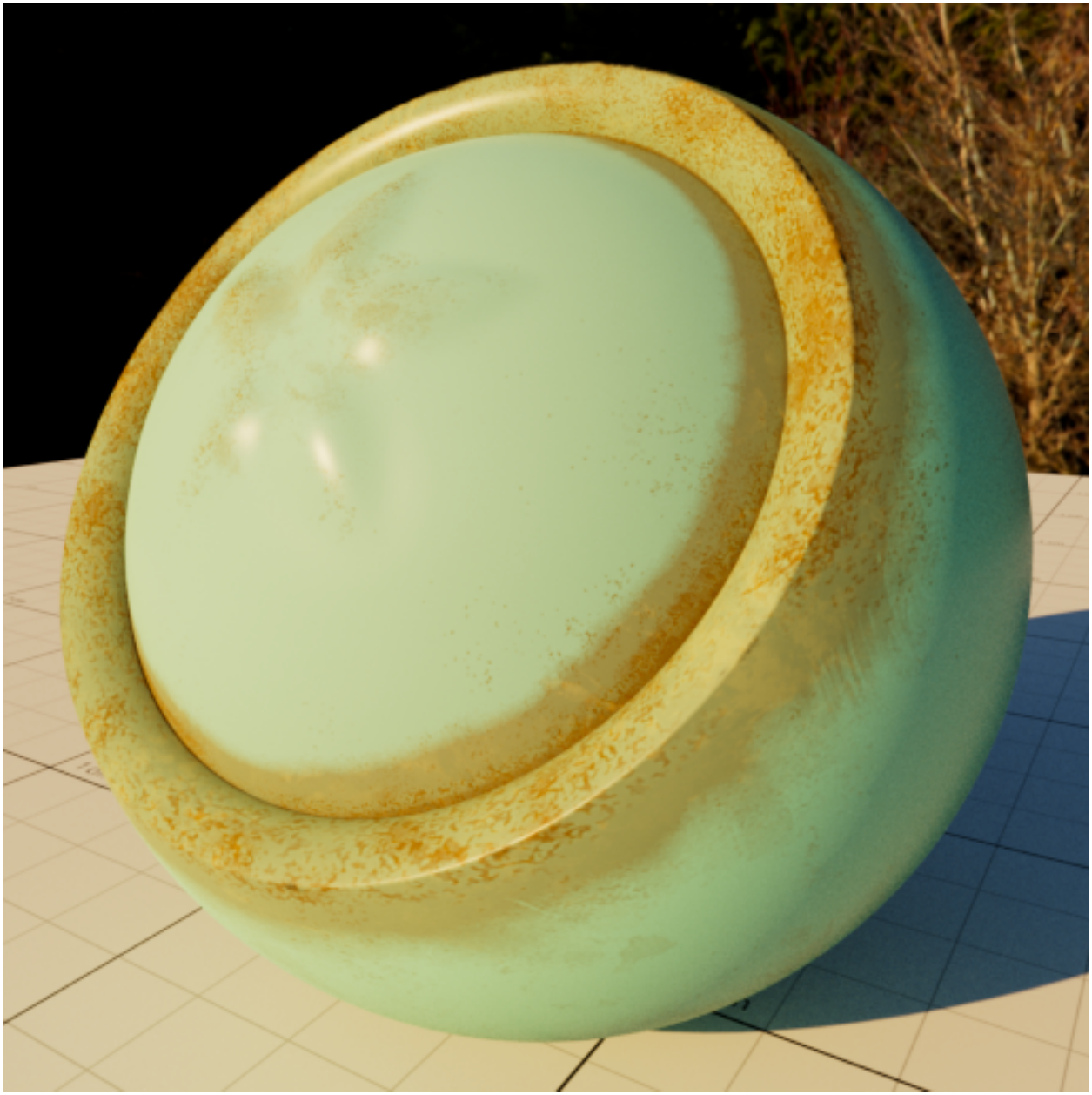


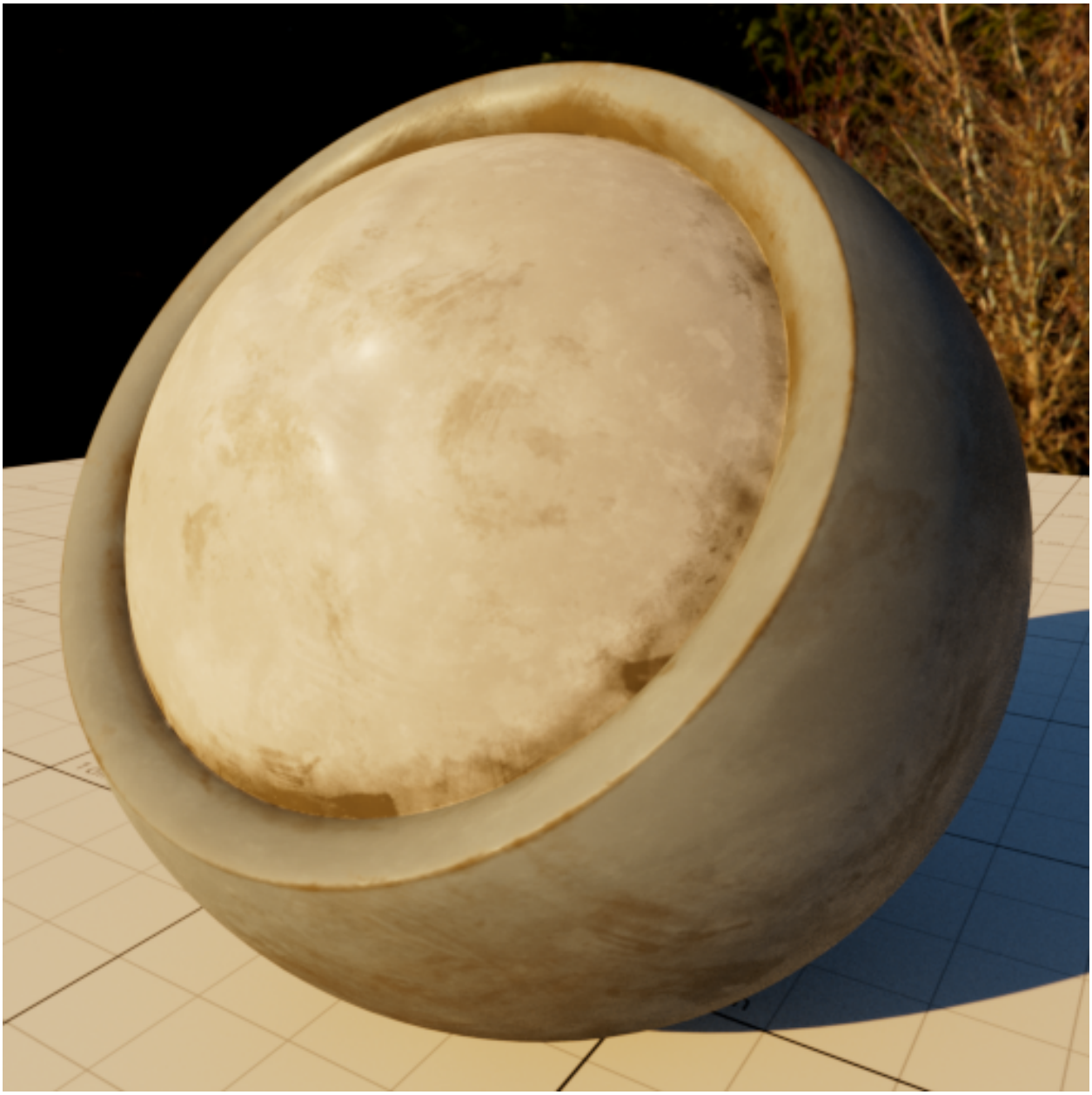


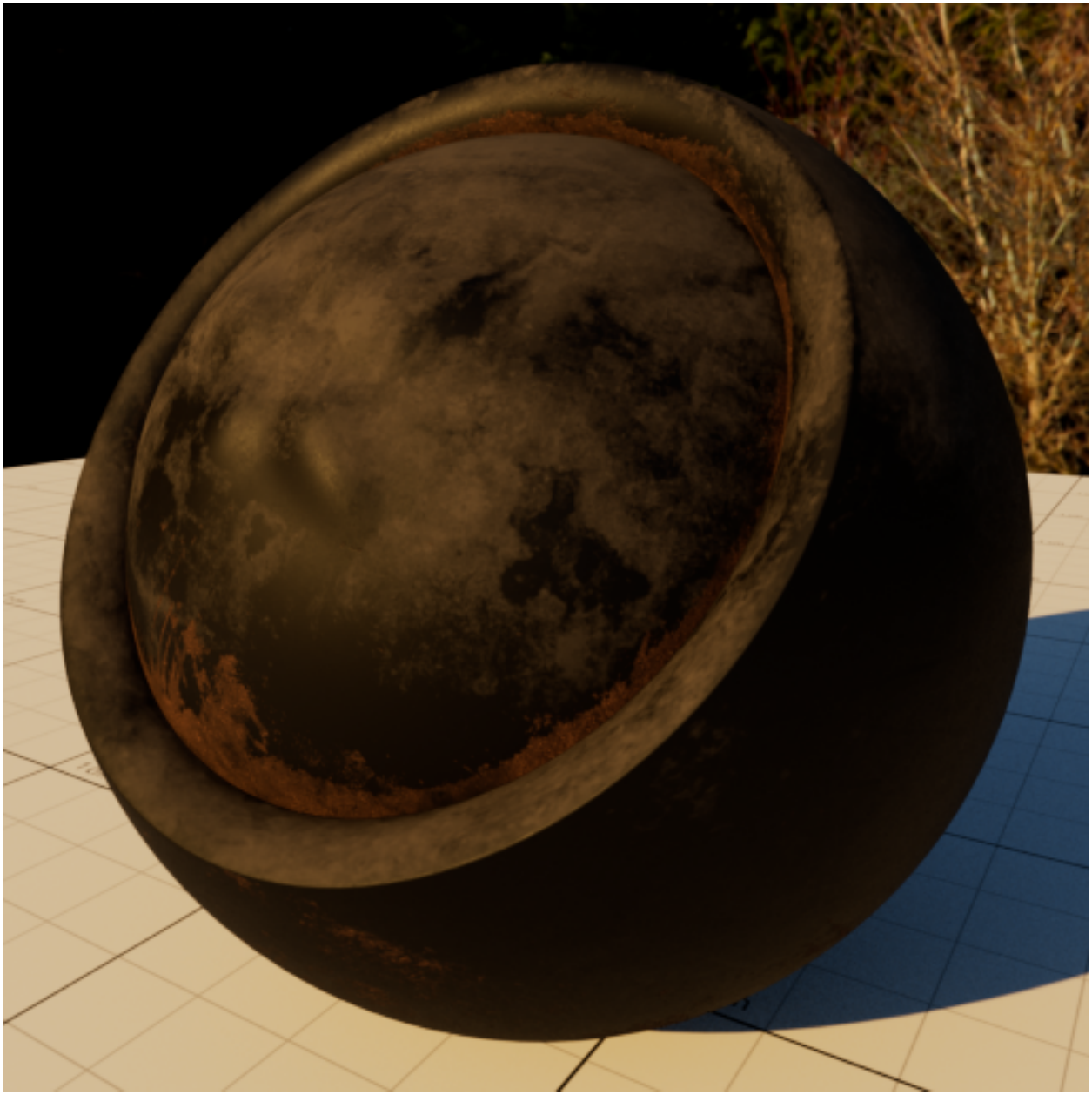


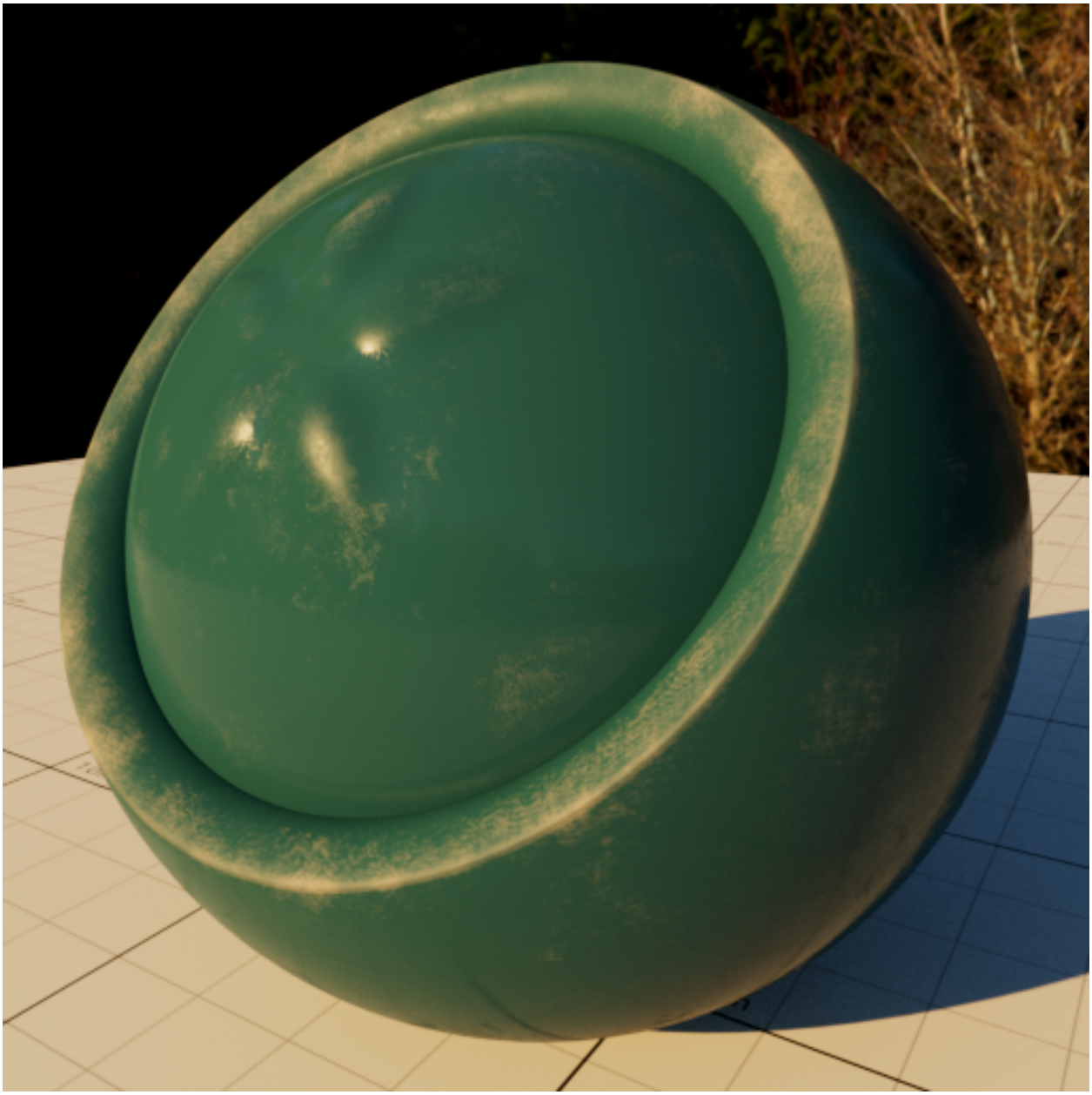












References

1.11 asSbsPBRMaterial

A shader created to give a good visual match with Allegorithmic Substance Painter's PBR material (*roughness* based workflow). It also has dedicated input slots for many of Substance Painter's exported auxiliary textures. For convenience the parameters were labelled and grouped as much as possible matching the conventions used in Substance Painter.

See also:

The Substance Painter PBR Guide.

1.11.1 Parameters

Ambient Occlusion

Ambient Occlusion This parameter does **not** compute any ambient occlusion itself. Rather it expects the exported ambient occlusion auxiliary texture from Substance Painter if such channel was exported (which is usually exported with scene-linear encoding).

Surface Color

Base Color The surface color channel, which usually is set by default to use the sRGB OETF[1].

See also:

See The importance of terminology and sRGB uncertainty for more details.

---

[1] By default, the color channel, and other color quantities are set to be encoded with the sRGB OETF. But note that you cannot specify RGB primaries or white point in Substance Painter, so you should assume these to be the sRGB/Rec.709 RGB primaries and D65 whitepoint if you intend to convert to other color spaces.

Emissive

Emissive Color The color to use for the incandescence effect, usually encoded with a sRGB OETF by default.

Emissive Intensity A scaling factor for the incandescence effect.

Height

Height The value to use for scalar bump mapping, usually a grayscale texture (and *raw* quantity).

Scale An overall scaling factor for the scalar bump mapping quantity *height*.

Metallic

Metallic The *metallicness* or *metallic* texture slot. With low values the specular appearance is that of a dielectric such as plastic or glass, while higher values produce the appearance of metal.

Normal

Normal The slot for normal mapping. In Maya this automatically creates a *bump2d* node whose *bump mode* must be set to *tangent space* bump mapping. With other DCC applications, you should create a *asBump* node instead, and set its *mode parameters* to *Normal Map* while having in mind that exporting a map in OpenGL or DirectX convention will flip the G channel of the exported map of one in regard to the other. See the *asBump documentation* for more details.

Roughness

Roughness The roughness texture channel. A *raw* quantity usually exported as a grayscale texture.

Opacity

Opacity The opacity texture channel, with a value of 0.0 denoting a fully transparent surface, and a value of 1.0 denoting a fully opaque surface. A *raw* quantity usually exported as a grayscale texture.

Specular

Specular Level The slot for the specular level texture. By default this quantity is not usually exposed to the user and is set to a fixed value.

Note: The *Specular Level* is not exposed by default, but it can be exposed. Its default value of 0.5 will be scaled internally and used to determine the reflectance at facing or normal incidence. This in turn will drive the Fresnel reflectance which is used to control the appearance of the dielectric specular term.

Anisotropy

Anisotropy Level The slot for the amount of anisotropy for the specular highlights, if the user created and exported such a channel. A *raw* quantity usually saved as a grayscale texture.

Anisotropy Angle The slot or value for an anisotropy rotation angle, where the value 1.0 maps to 360 degrees rotation. A *raw* quantity usually saved as a grayscale texture.

Refraction

Refraction The amount of refraction, where a value of 0 is a diffuse BRDF[2], and a value of 1 is a pure smooth specular BTDF[3], and in-between values blending between the diffuse and refractive term.

Refraction IOR The index of refraction[4] of the material. This parameter affects **only** the refraction term and has no effect whatsoever on the specular term.

Scattering Controls how much light is scattered through the surface. Unused at the moment.

Absorption Controls how much light is absorbed through the surface, with a value of 1.0 leading to the light being fully absorbed by the medium.

Absorption Color Controls how light shifts when light traverses the medium's volume.

Matte Parameters

Enable Matte Flag toggling matte holdouts on or off.

Matte Opacity Overall scaling factor for the matte, from solid black to normal.

Matte Opacity Color Color for the matte.

---

[2] Which in this case is the Lambert BRDF.

[3] See also the definition of BSDF for more details.

[4] The *real* and absolute index of refraction of the material. Since dielectrics have a very small extinction coefficient, this is assumed to be 0, and monochromatic. Absolute since it's assumed to be the ratio of the wave speed in the vacuum and wave speed in the medium. Appleseed is **not** querying the exterior medium with *nested dielectrics [SB02]* to compute the correct index of refraction. The priority was given to have as much as possible a matching appearance with Substance Painter's PBR material.

Advanced Parameters

Maximum Ray Depth The maximum number of bounces a ray is allowed to travel.

1.11.2 Outputs

Output Color The final result color.

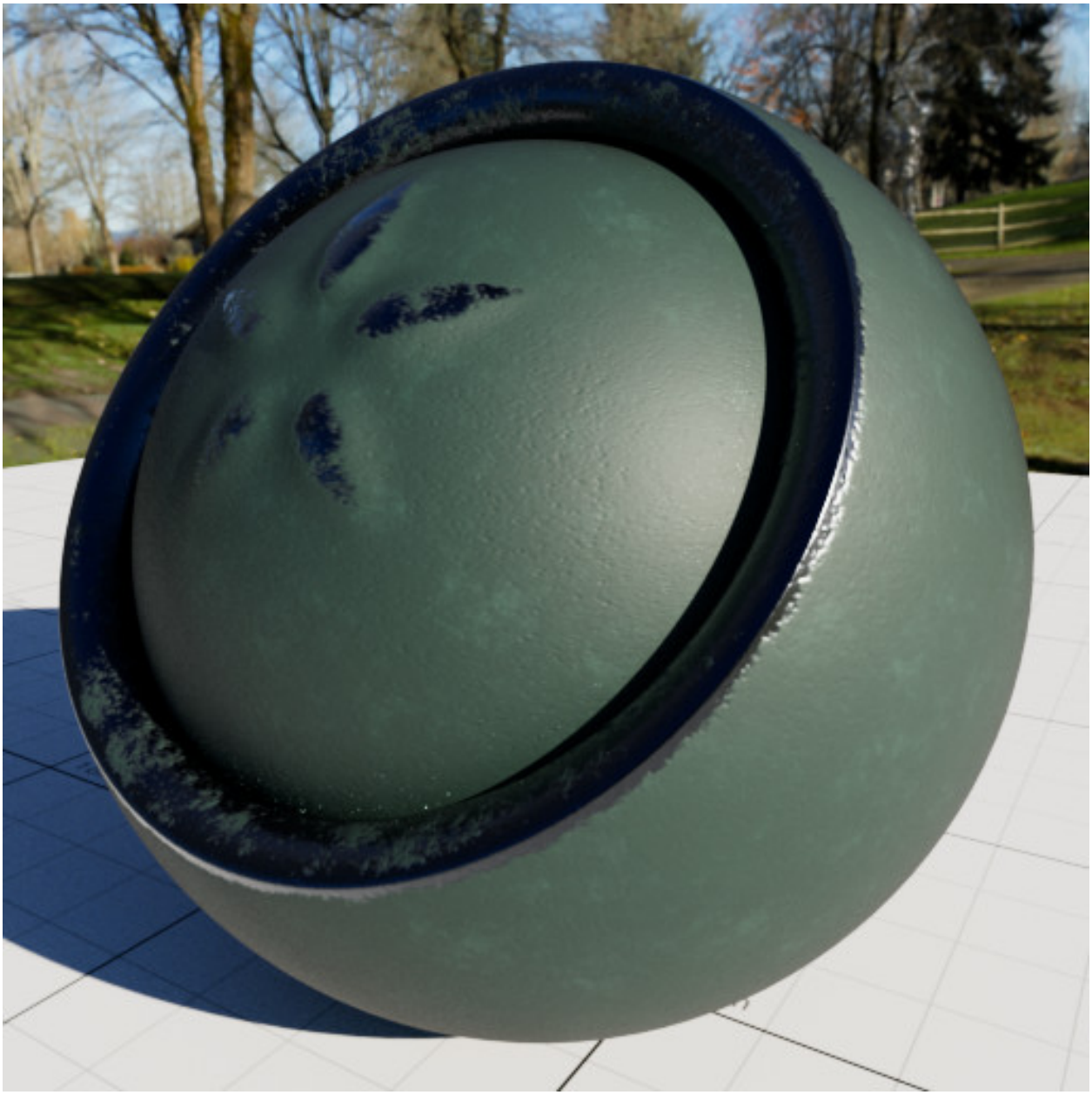
Output Transparency The final transparency color.

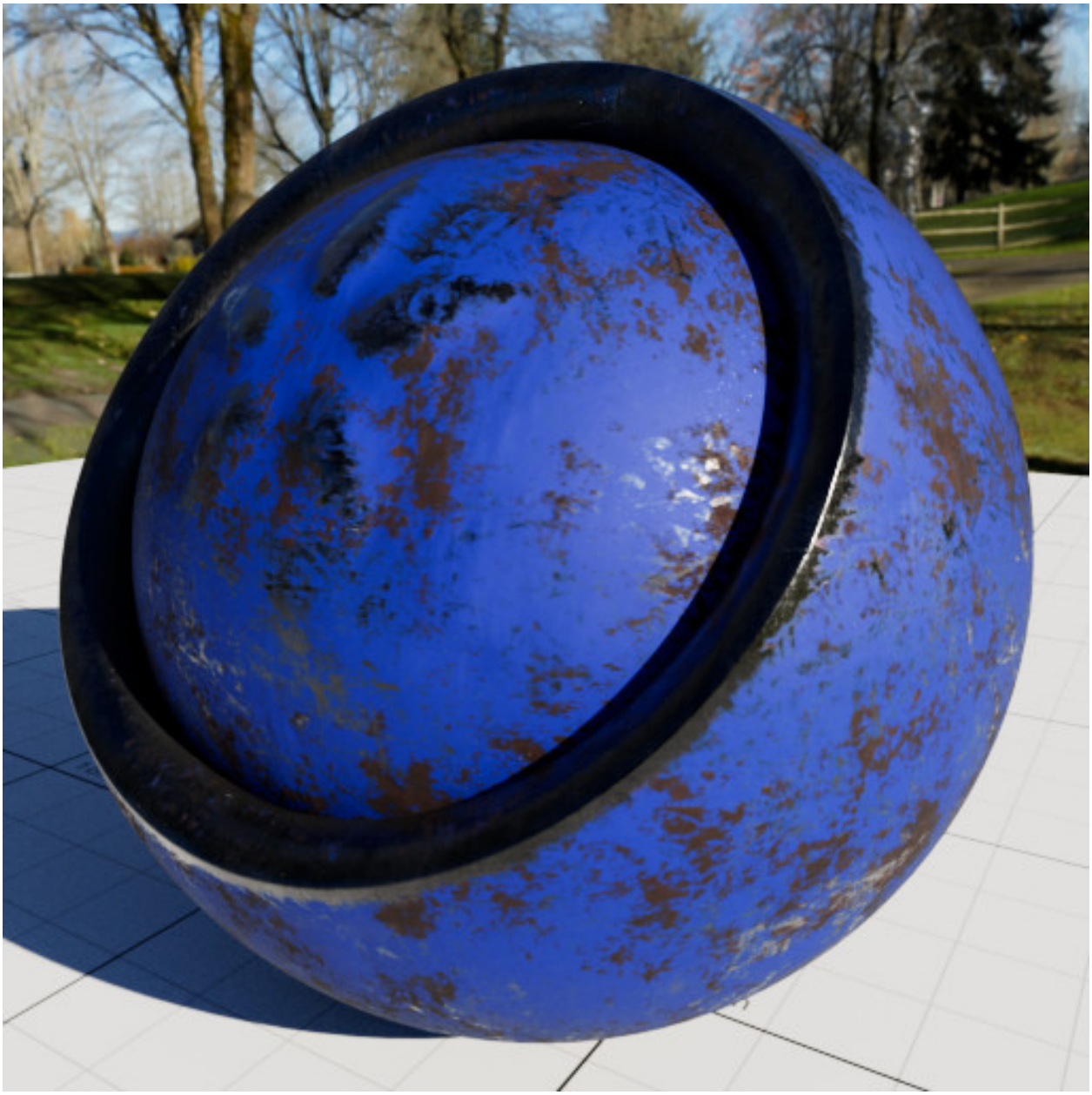
Output Matte Opacity The final matte opacity. Note that OSL *holdout* is unsupported at the moment.

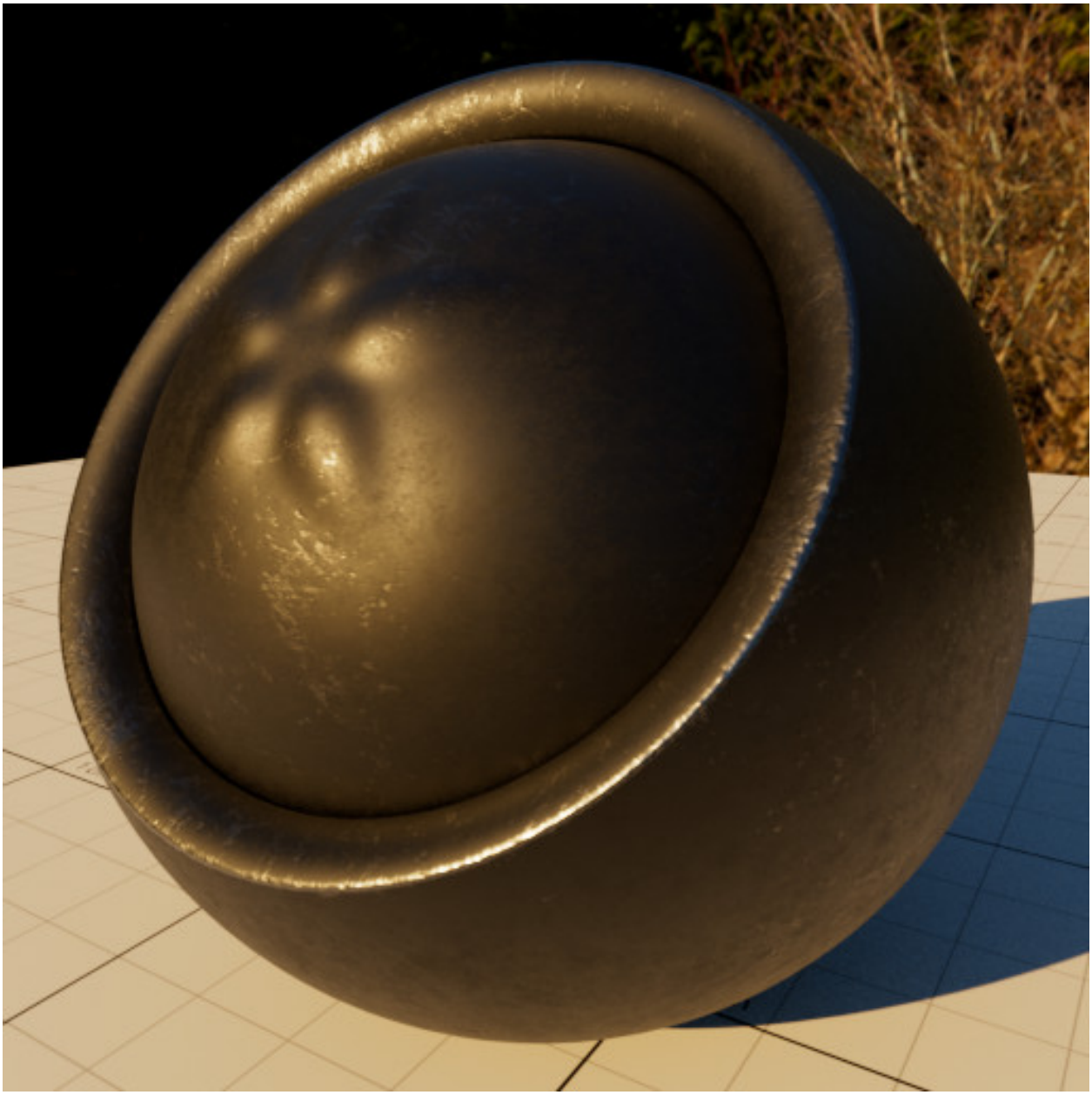
Notes

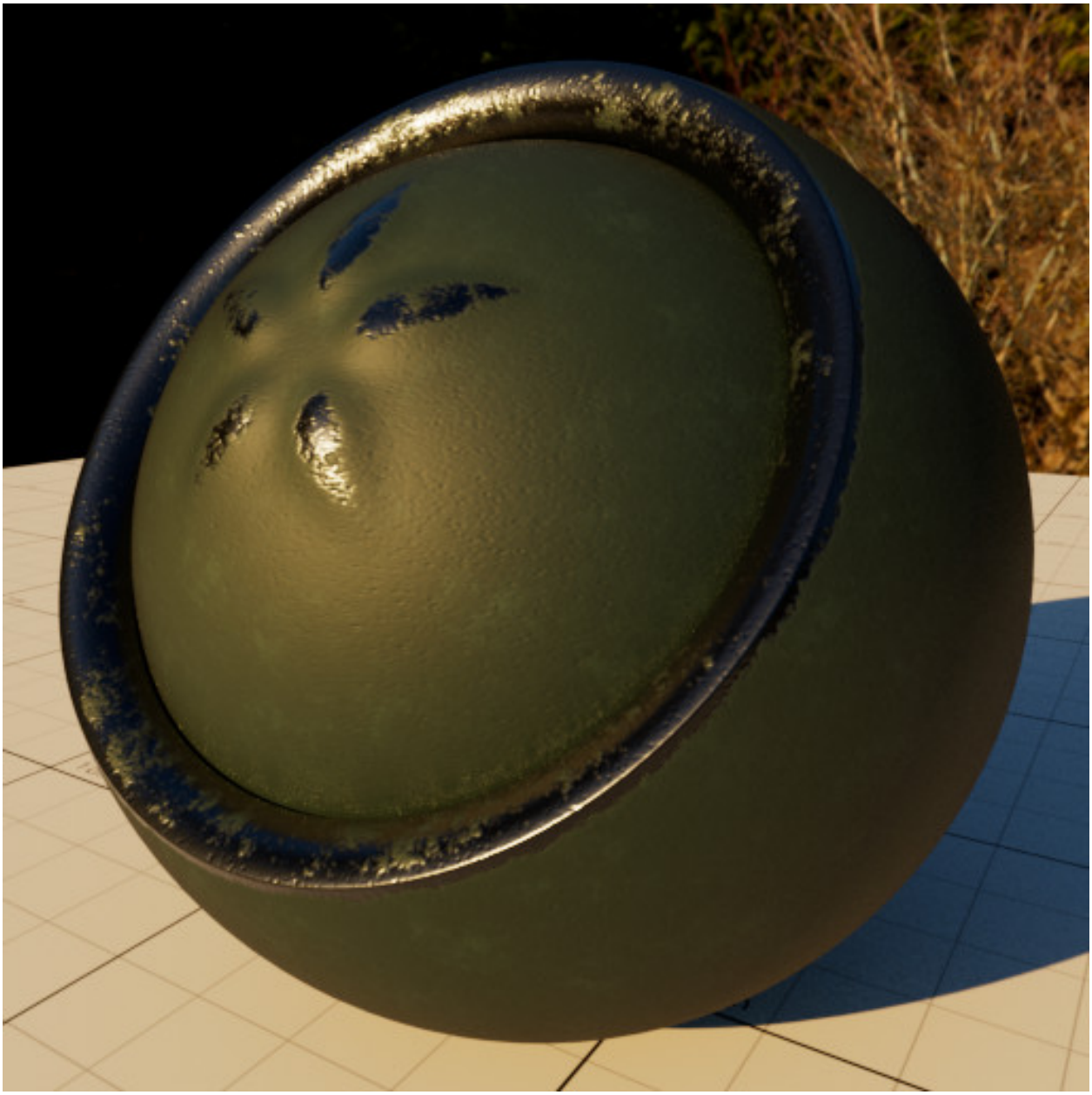
Attention: When using texture atlas such as UDIMs, have in mind that you're probably going to have a large number of channels and textures to access. This will have a cost in performance. One way to maximize performance and mitigate this cost is to pre-process the textures with OpenImageIO's maketx utility.

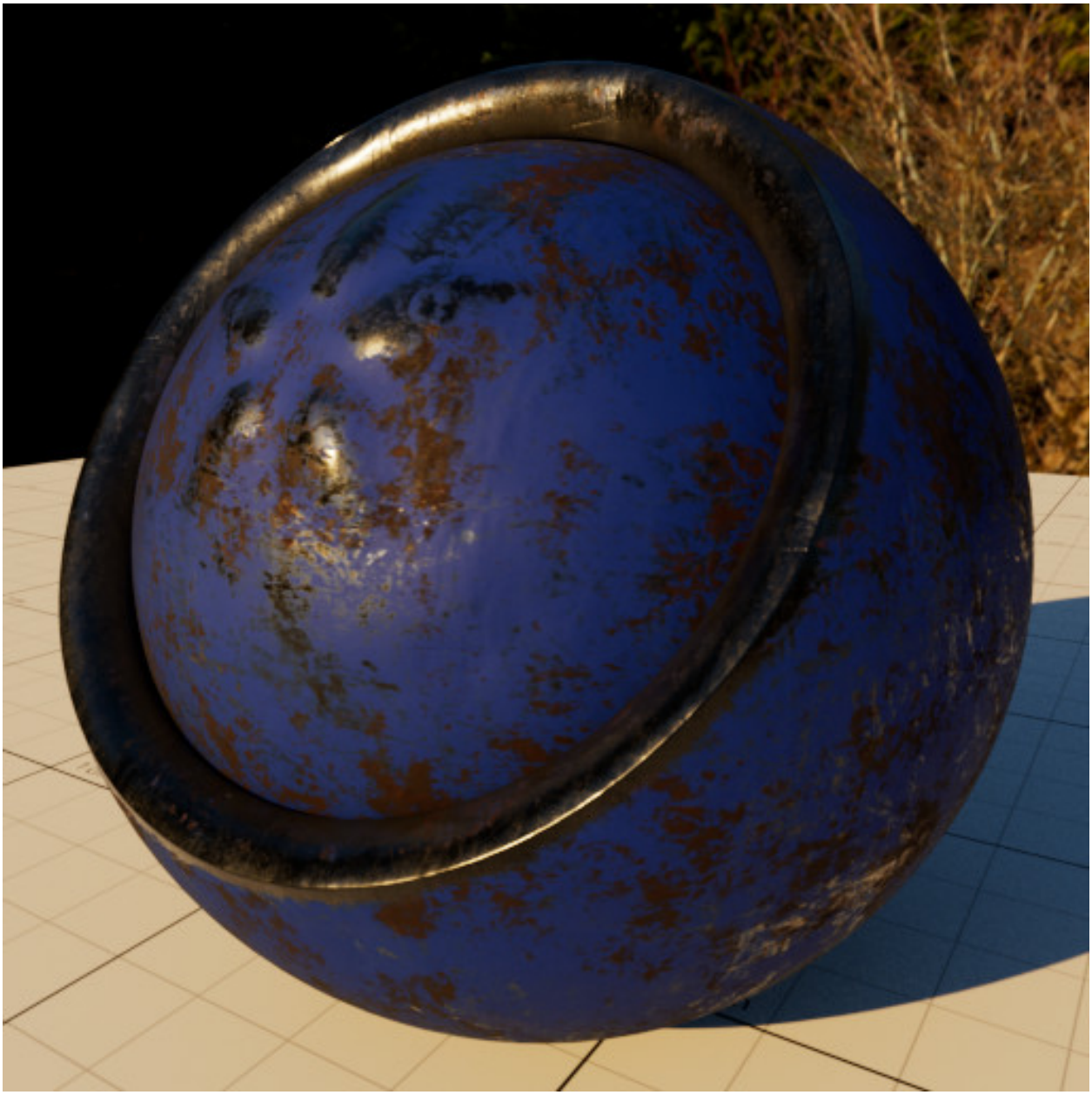
1.11.3 Screenshots

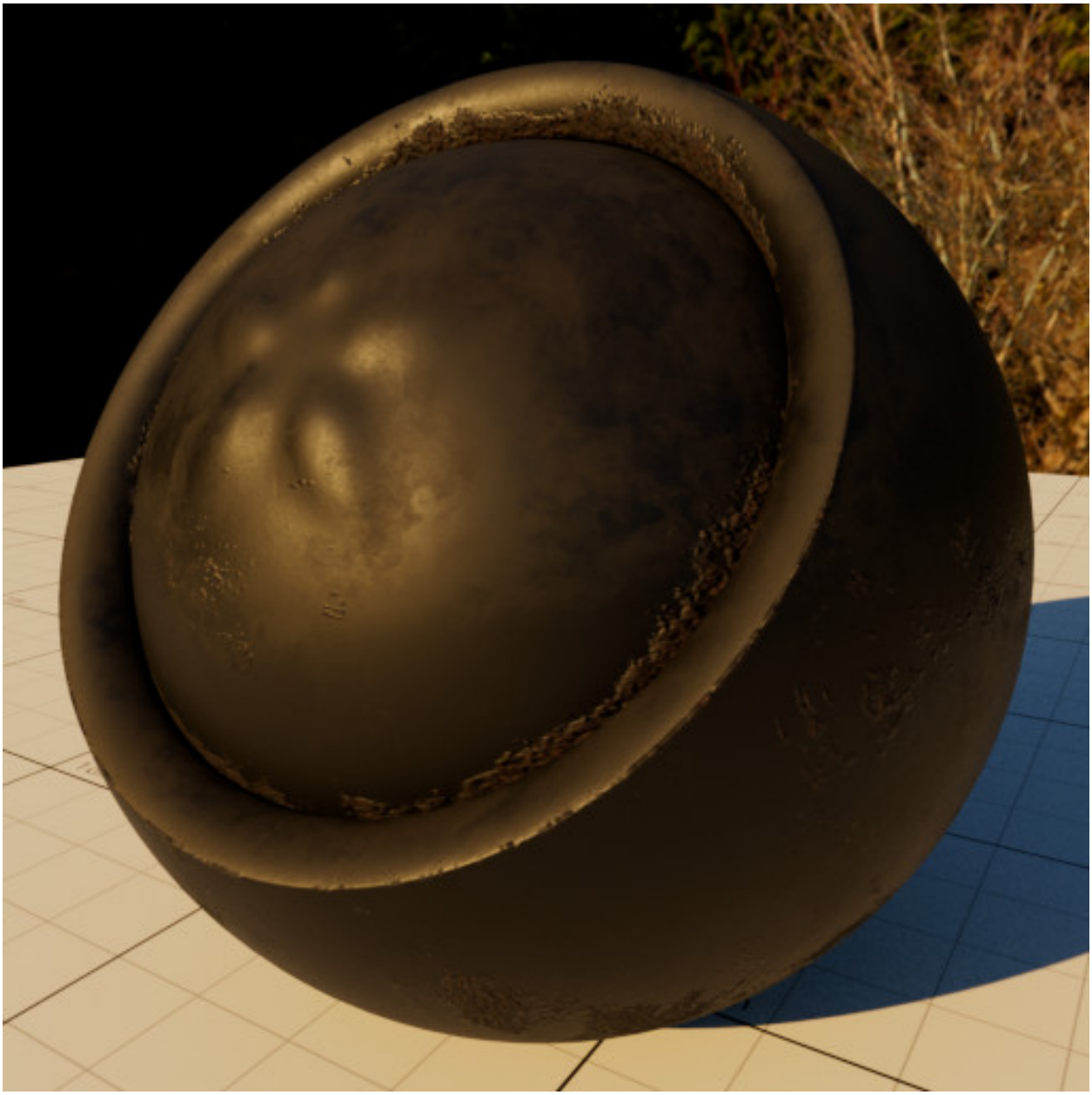












References

1.12 asStandardSurface

A physically based material with a dielectric coating with absorption, and a substrate with optional subsurface scattering, refraction, volumetric absorption.

1.12.1 Parameters

Diffuse Parameters

Diffuse Weight A scaling factor for the diffuse BRDF.

Diffuse Color The surface color.

Diffuse Roughness The diffuse roughness, with a value of 0, it corresponds to a Lambert diffuse term. Higher values *flatten* the appearance of the surface, giving it a chalky look.[1]

Subsurface Parameters

Subsurface Weight A weighting factor for the subsurface scattering term.

Subsurface Mean Free Path (MFP) Color controling how far light enters and travels within the medium.

Subsurface MFP Scale Overall scaling factor for the MFP color, which is expected in [0,1] range. Values above 1.0 are possible, resulting in increased translucency appearance.

Advanced

Subsurface Profile The diffusion profiles to use in the BSSDRF[2]. This parameter can take the following values:

- Gaussian *[dEonLE07]*
- Better Dipole
- Normalized Diffusion *[Chr15]*

[1] The diffuse BRDF used is the Oren-Nayar BRDF *[30]*

[2] The specular (microfacet) BRDF is using Student's t-distribution *[RBMS17]*. This includes the Beckmann *[BS63]*, *[CT82]* and GGX *[WMLT07]* distributions.

- Random Walk *[MHD16]*

Translucency Parameters

Translucency Weight Controls the amount of thin translucency of the object.

Translucency Color Color affecting the thin translucency term.

Specular Parameters

Specular Color Overall specular tint for the specular BRDF.

Specular Roughness The apparent surface roughness affecting the specular highlights.

Specular Spread Specular highlights *tails*, with low values producing sharp highlights, and higher values producing soft veiled highlights.[3]

Fresnel

Fresnel Type Allows the user to choose the specular mode, of a dielectric such as plastic or glass, or of a conductor or metal.

Note: To use refraction, the mode **must** be set to *dielectric* and the index of refraction set. When in *conductor* mode, the *face tint* and *edge tint* parameters are used to derive the complex index of refraction instead *[Gul14]* and no refraction is used.

IOR The index of refraction for the dielectric mode.

Facing Tint The reflectance at normal incidence.

Edge Tint Reflectance at grazing incidence.

Anisotropy

Anisotropy Amount Overall intensity of the anisotropy effect, with a value of 0.0 representing a isotropic specular highlight.

Anisotropy Angle Rotation angle for the anisotropic highlight in [0,1], mapping a rotation from 0 to 360 degrees.

Anisotropy Map Also known as tangent field, encodes the anisotropy directions along X and Y in the Red and Green or Red and Blue channels of the image. Appleseed expects values encoded in the Red and Green channels.

Refraction Parameters

Refraction Amount Intensity of the refraction, only taking place when *Fresnel* is *dielectric [WMLT07]*.

Refraction Tint Overall tinting factor, it affects the BTDF equally, unlike volumetric absorption.

[3] See also Extending the Disney BRDF to a BSDF with Integrated Subsurface Scattering for details.

Volumetric Absorption

Absorption Depth Sets the depth at which full absorption takes place. Low values result in dense absorbing materials, high values in transparent appearance ones.[4]

Absorption Color The color used for the volumetric absorption.

> **Warning:** The refraction BSDF cannot presently support layered coatings nor coating absorption. This will be covered in a future release supporting layered closures.

Coating Parameters

Coating Reflectivity Intensity of specular highlights on the coating.

Coating Roughness Apparent surface roughness of the coating specular highlights.

Coating IOR Index of refraction of the coating layer, usually a dielectric, with values around 1.5.

Coating Absorption

Coating Thickness Thickness of the coating layer, controlling the intensity of coating absorption, with 0 being no absorption, 1 being full absorption.

Coating Absorption Absorption color for the coating, white has no effect, black absorbs fully.

Incandescence Parameters

Incandescence Amount The overall intensity of the incandescence effect.

Incandescence Type Color choice for incandescence color, with *constant* taking as input the user-set value, and *blackbody* using a blackbody radiator. **[#]_**

Incandescence Color Incandescence color, ignored in *blackbody* mode.

Temperature Temperature in Kelvin degrees, ignored in *constant* mode.

Options

Area Normalize EDF Normalize by the object area, so that object deformations keep the incandescence energy. If unset, deforming the object will retain the incandescence color.

Tonemap EDF Tonemaps the potentially high energy result of the blackbody radiator into the [0,1] range.

Note: The *tonemap EDF* option has effect **only** when *incandescence type* is set to *blackbody*.

---

[4] https://en.wikipedia.org/wiki/Black-body_radiation

Transparency Parameters

Transparency Color transparency. Affects the *presence* of an object. When transparency is binary (full opaque or full transparent, with no in-between values), appleseed alpha masks should be used instead.

Bump Parameters

Coating Normal The bump normal for the coating layer.

Substrate Normal The bump normal for the substrate.

Matte Parameters

Enable Matte Flag toggling matte holdouts on or off.

Matte Opacity Overall scaling factor for the matte, from solid black to normal.

Matte Opacity Color Color for the matte.

Advanced Parameters

SSS Ray Depth Maximum number of ray bounces for the subsurface scattering term.

SSS Threshold Defines the distance light has to travel within the medium to start the subsurface scattering effect. A low enough mean free path value will have a visually negligible difference from a diffuse term. This parameter sets the threshold at which the subsurface calculations start, instead of the ordinary diffuse term.

Maximum Ray Depth The maximum number of bounces a ray is allowed to travel.

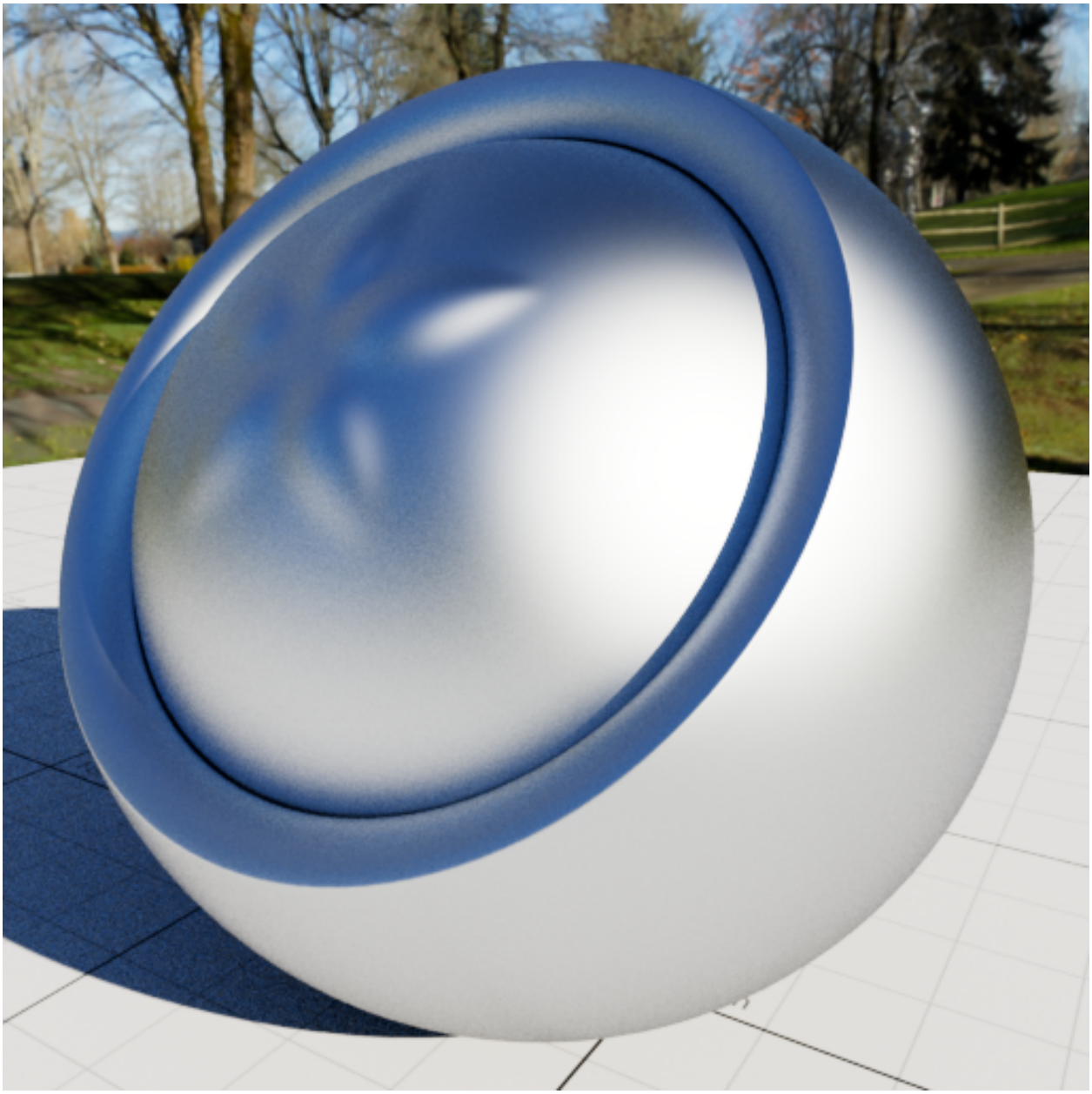
1.12.2 Outputs

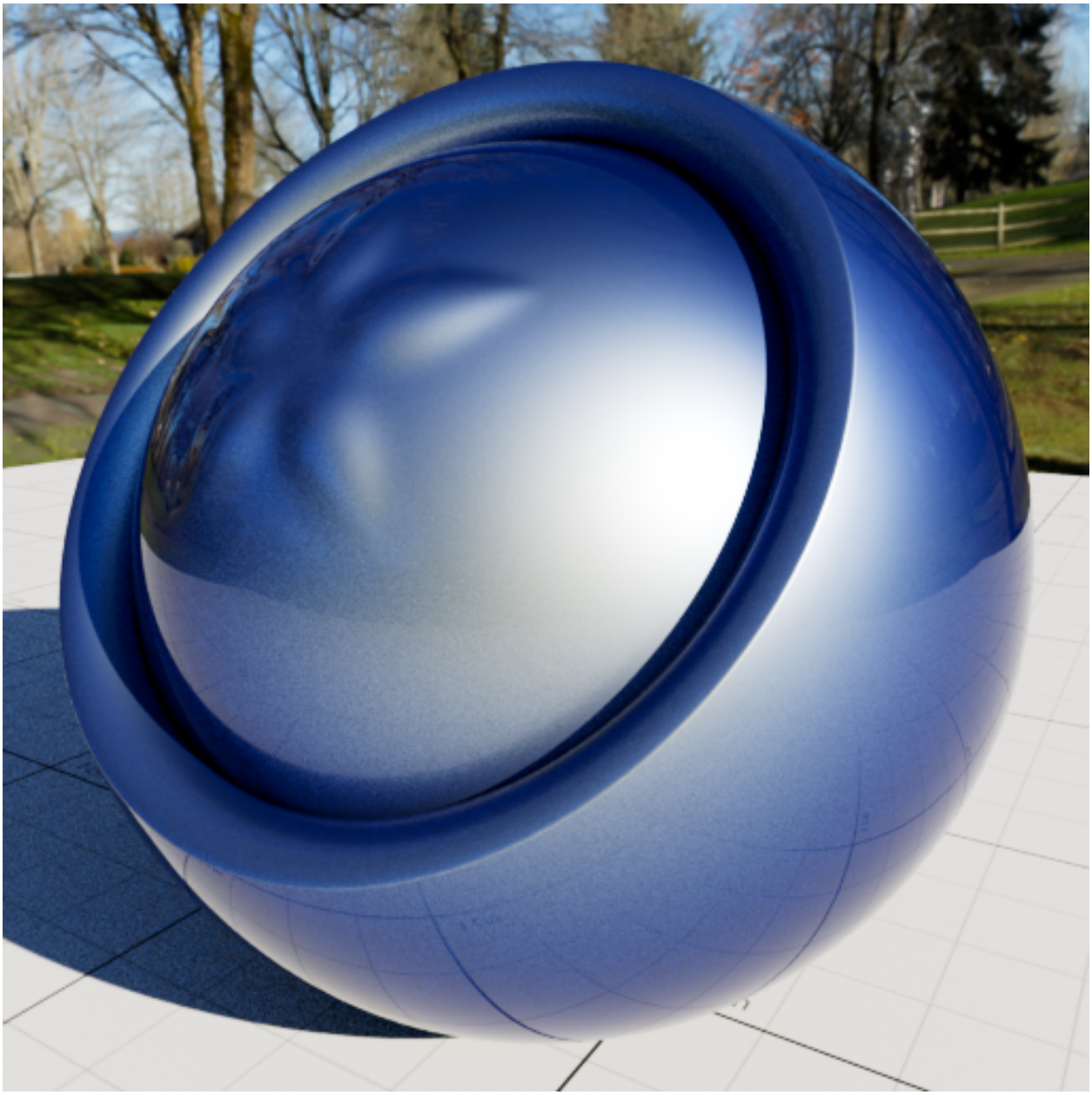
Output Color The final result color.

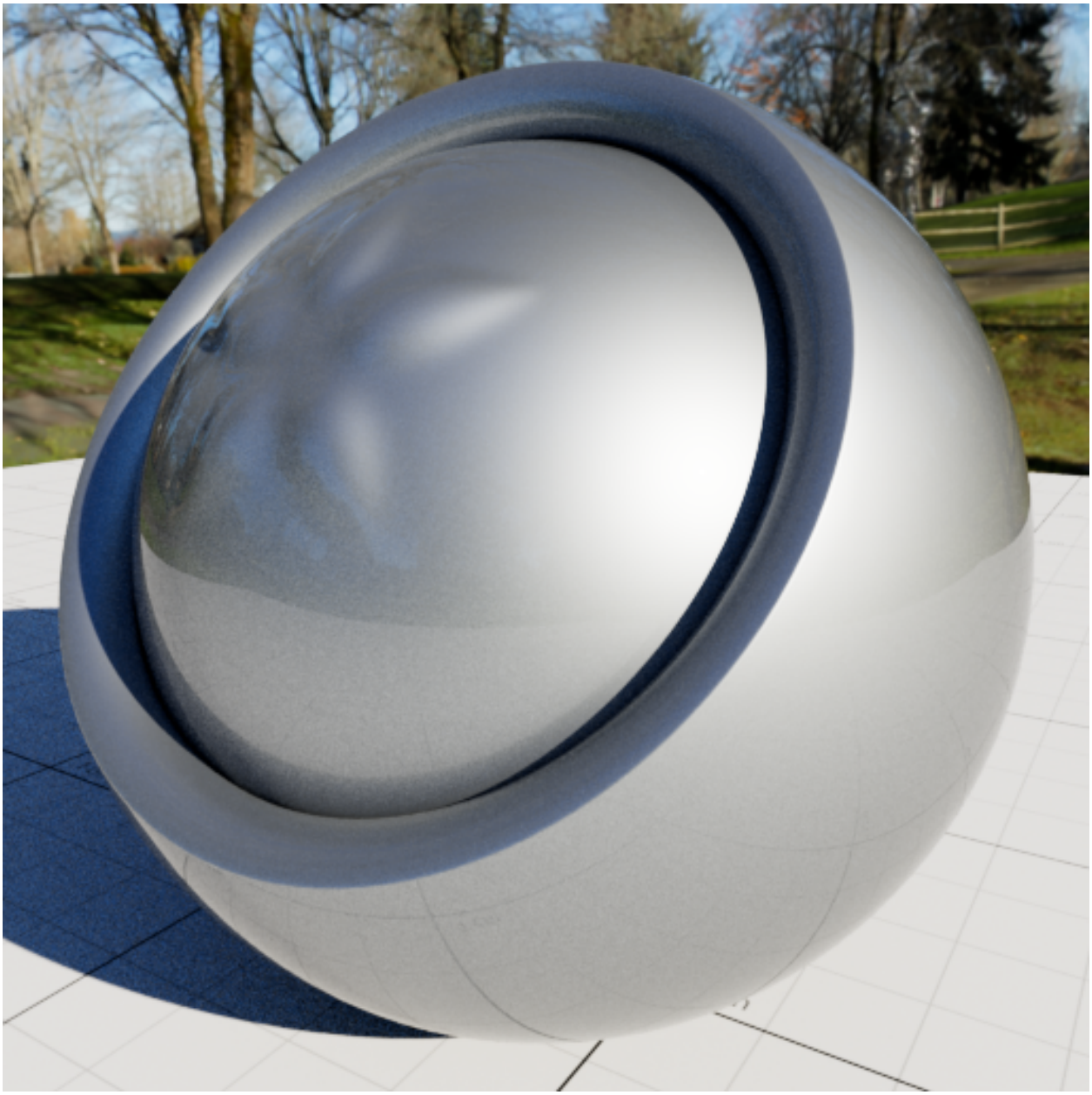
Output Transparency The final transparency color.

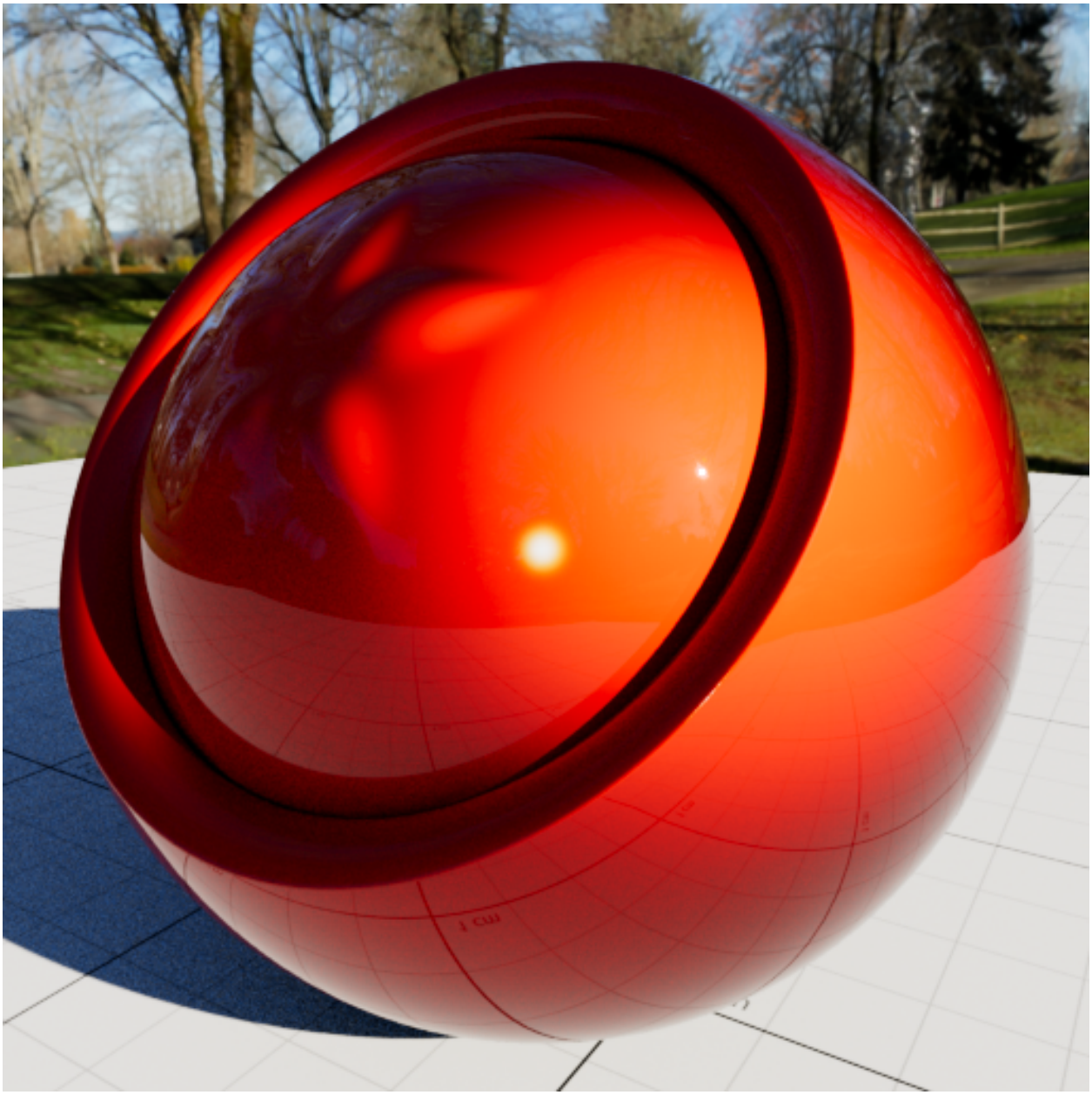
Output Matte Opacity The final matte opacity. Note that OSL *holdout* is unsupported at the moment.

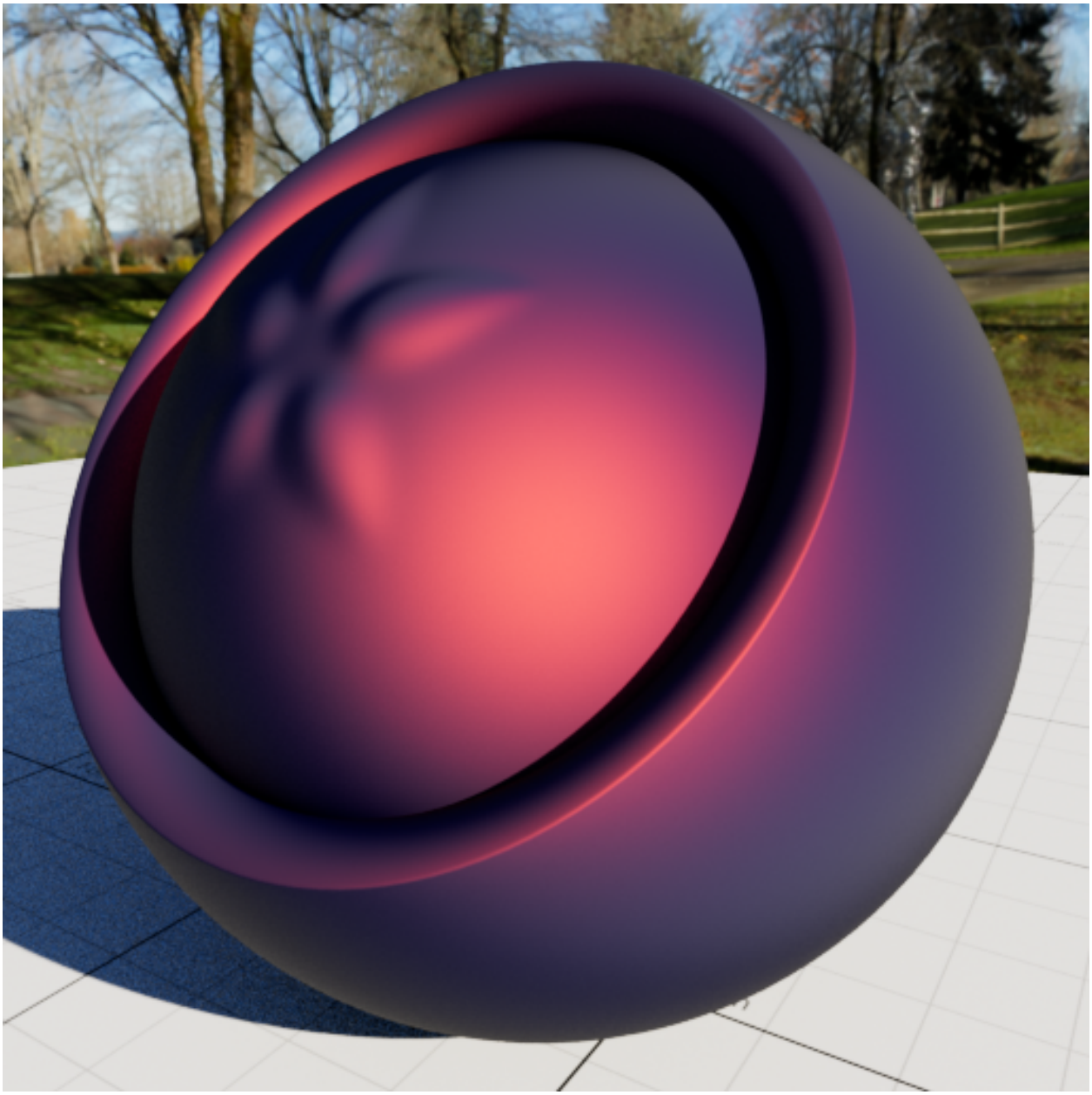
1.12.3 Screenshots

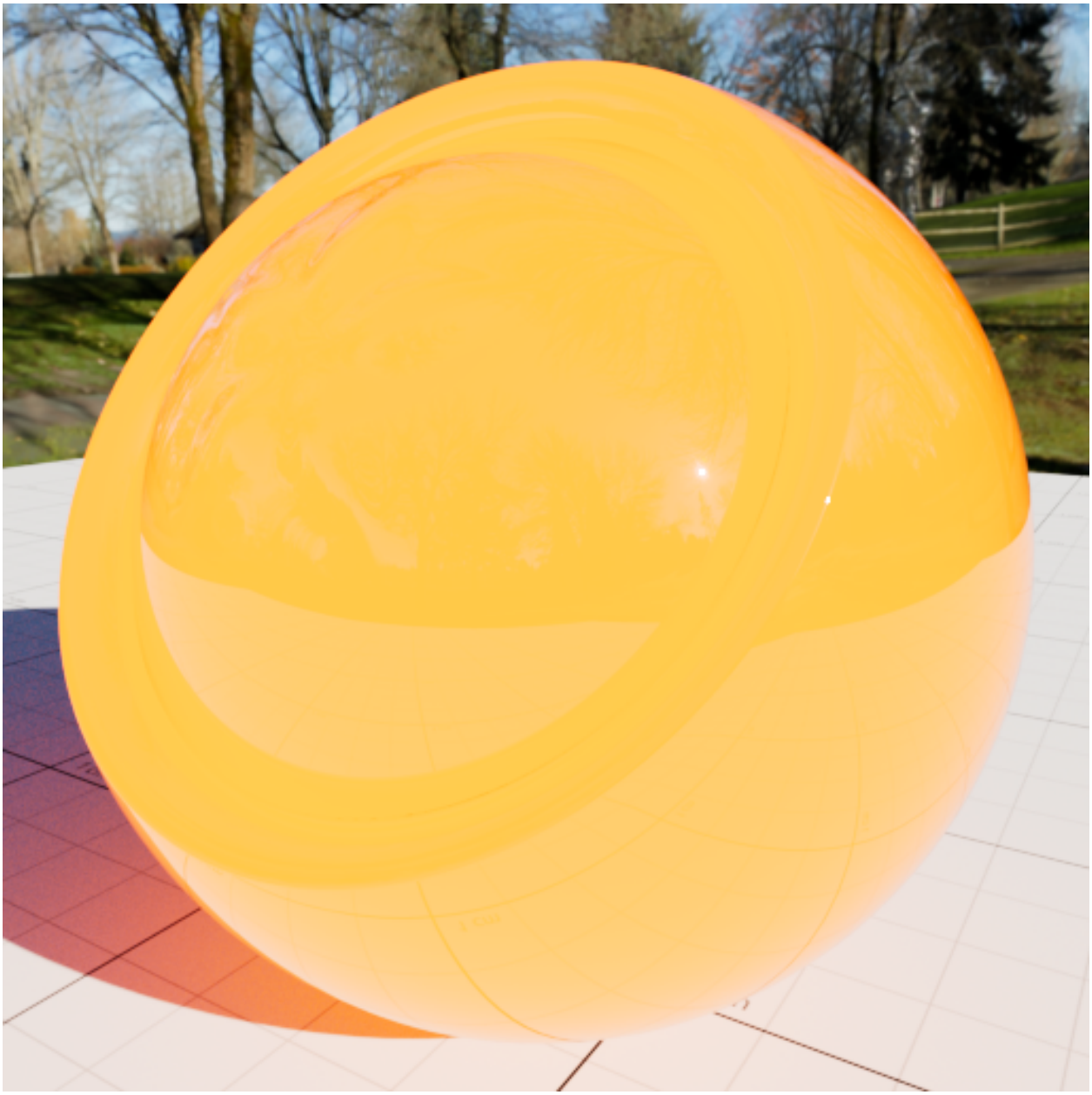


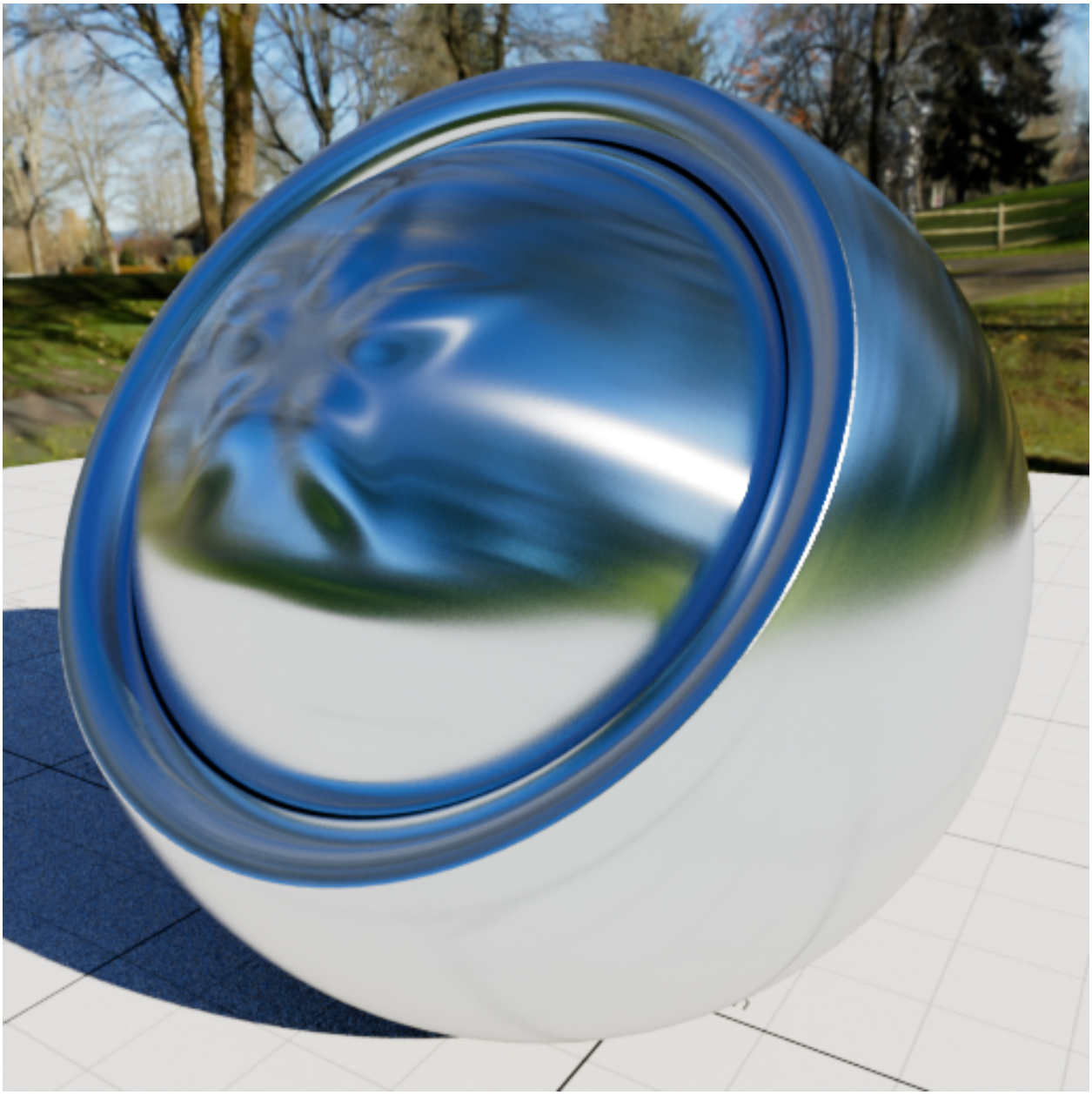


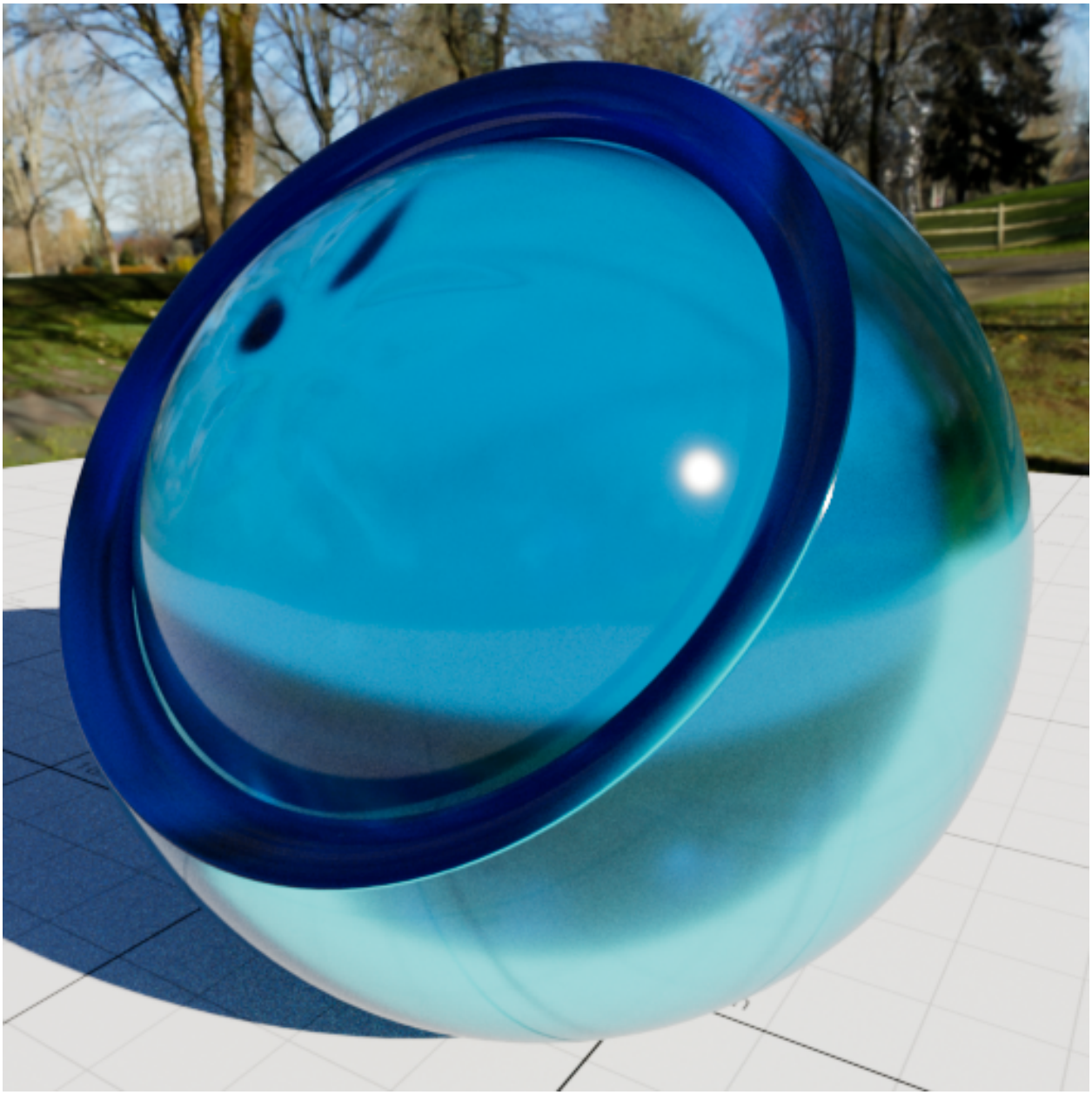


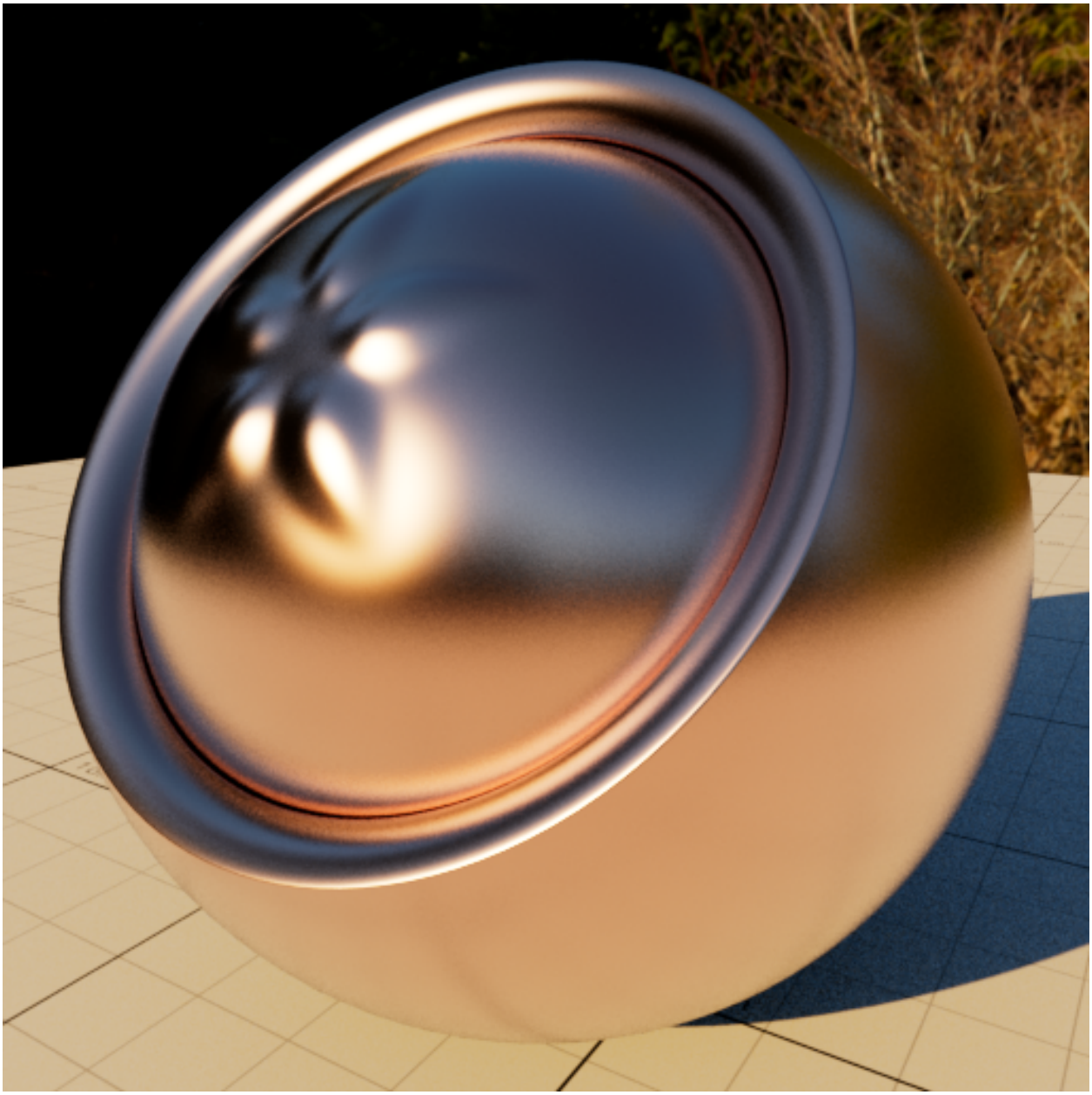


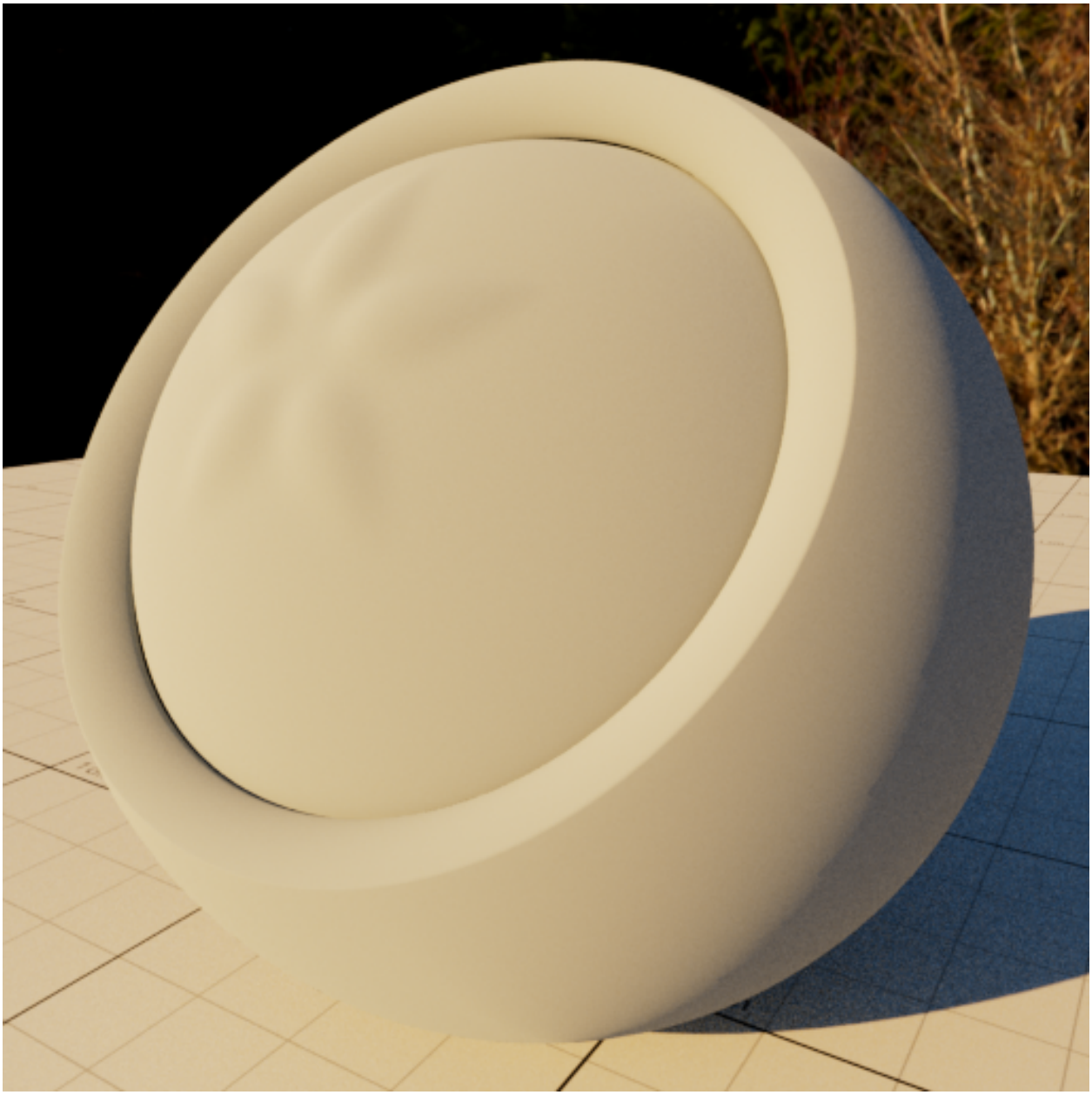


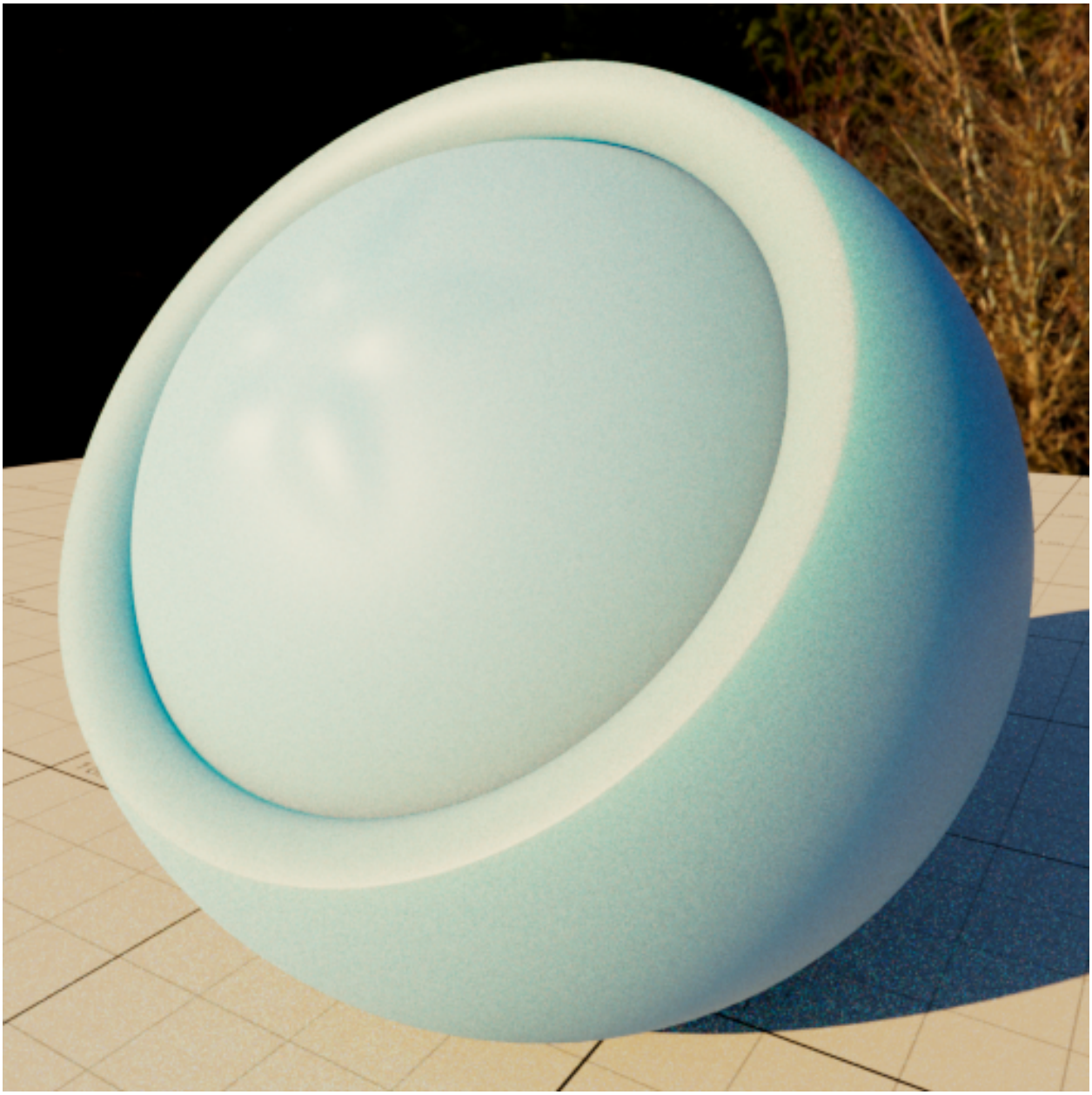


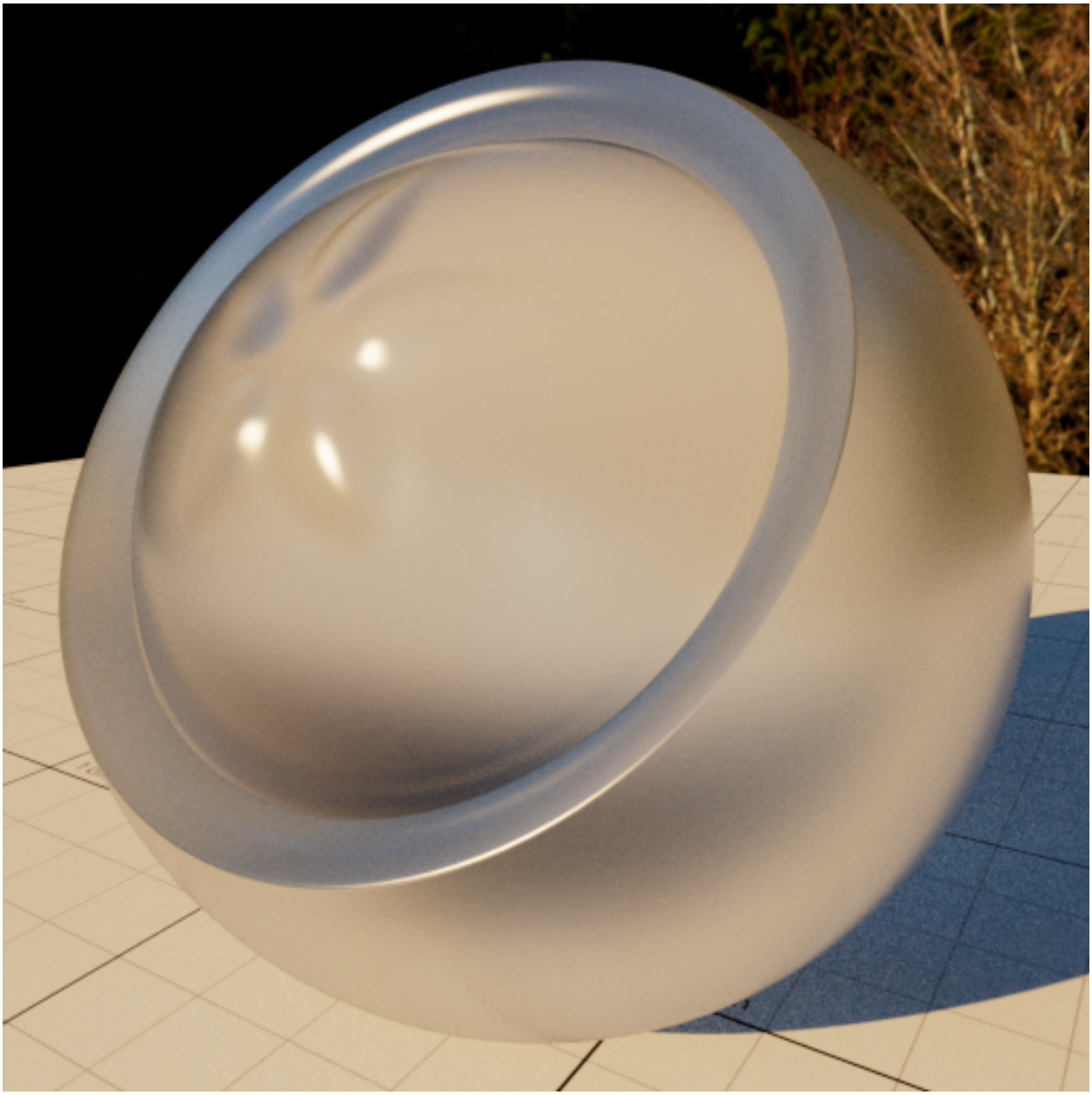


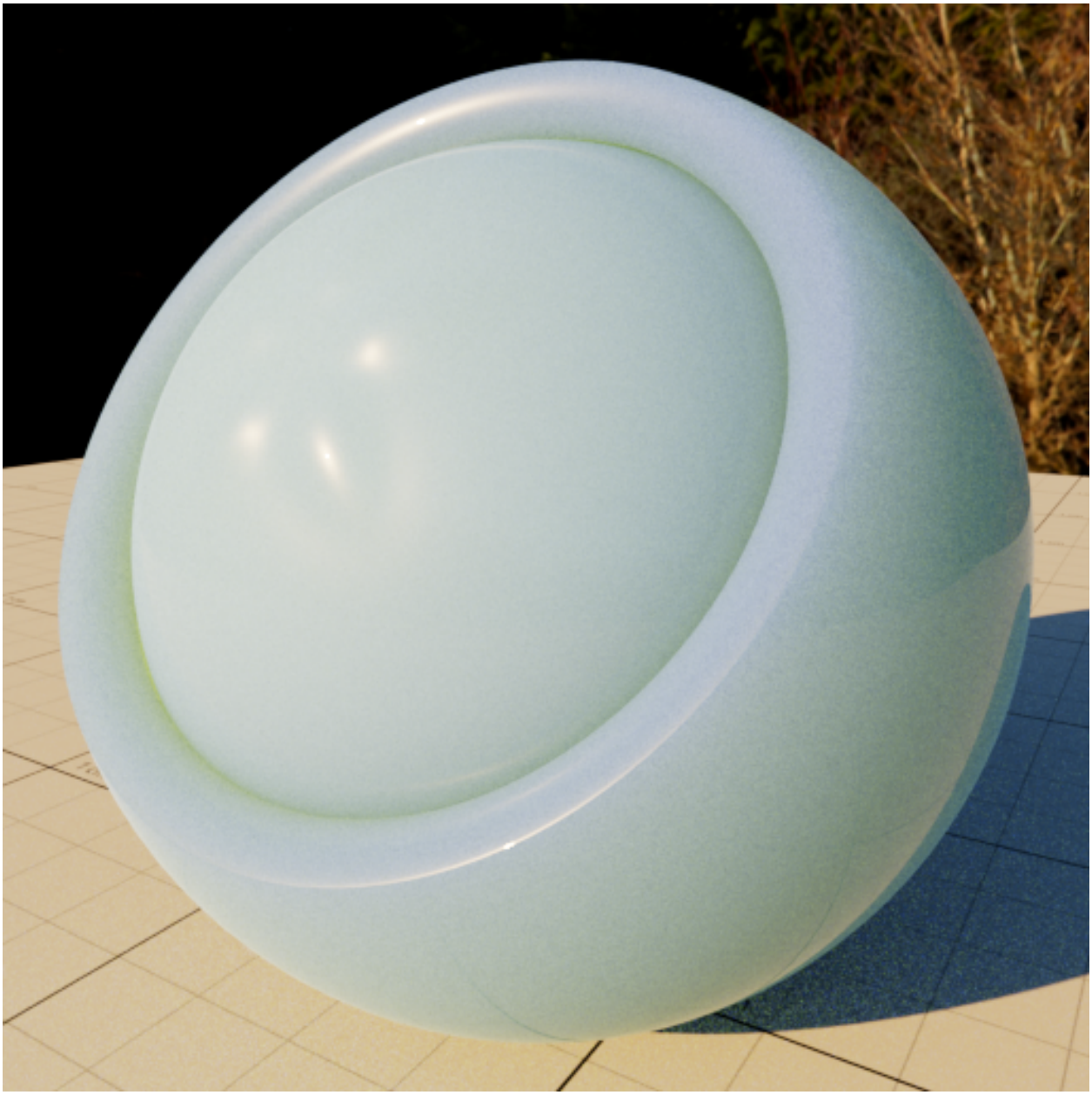


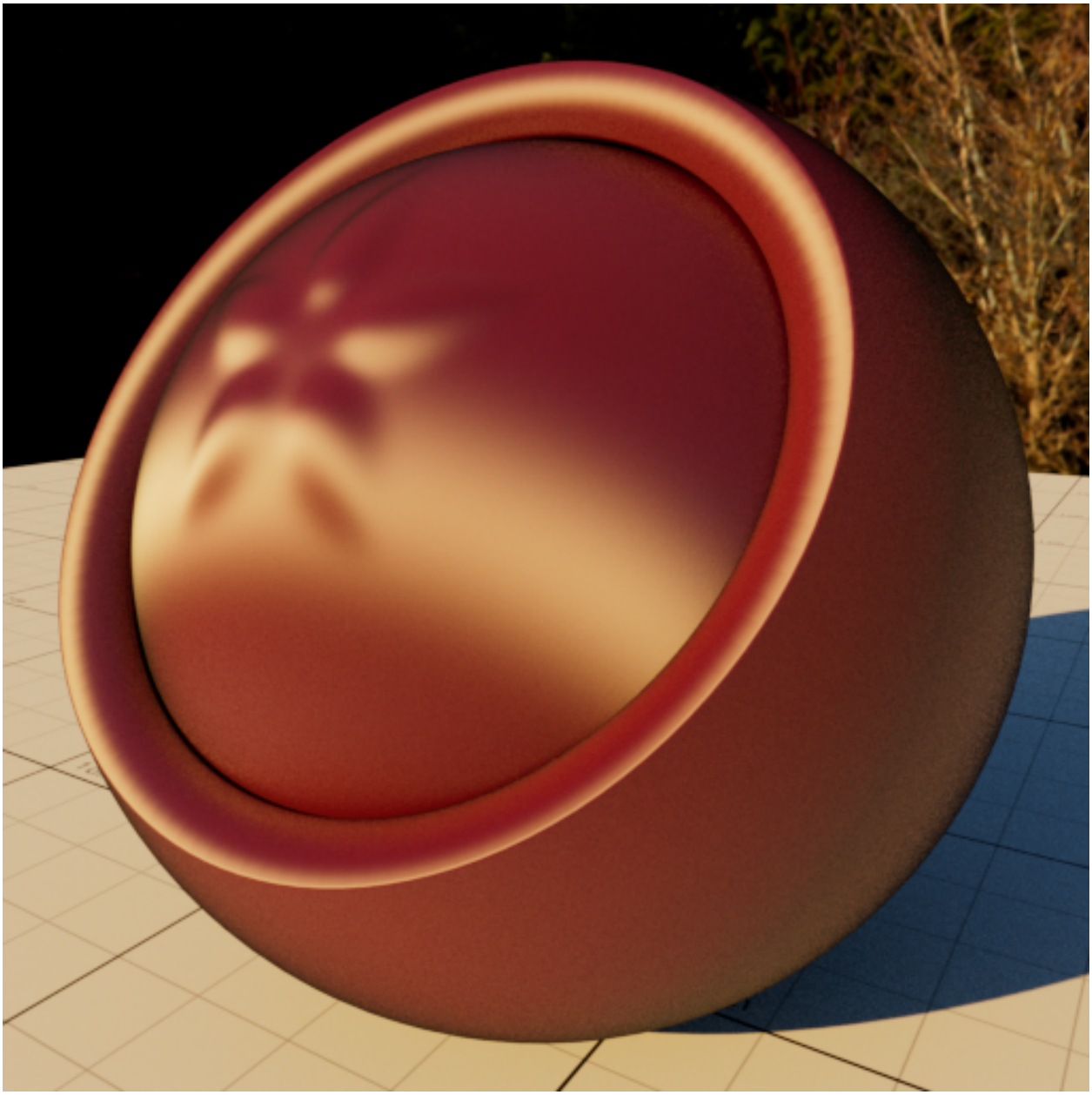


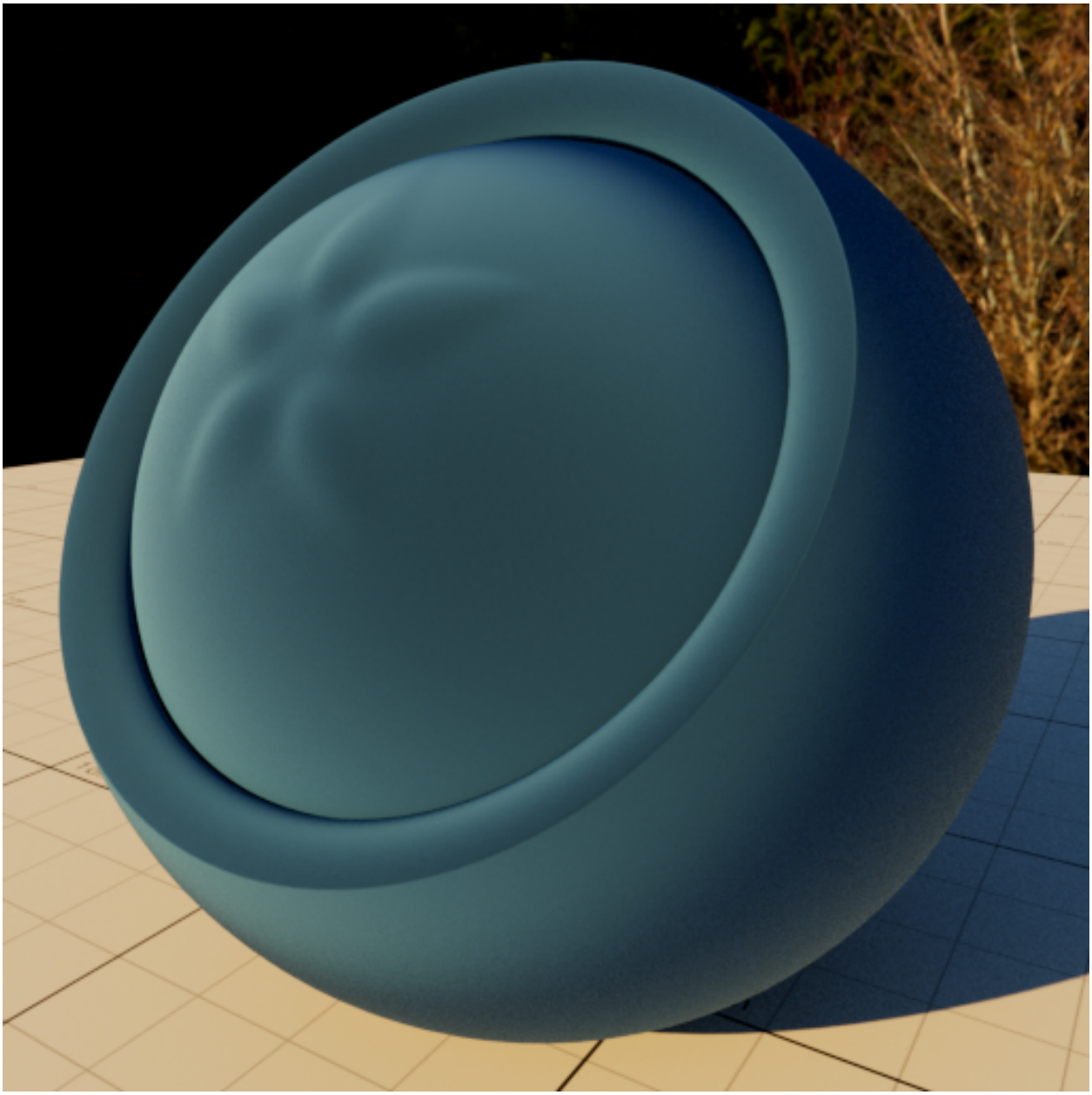


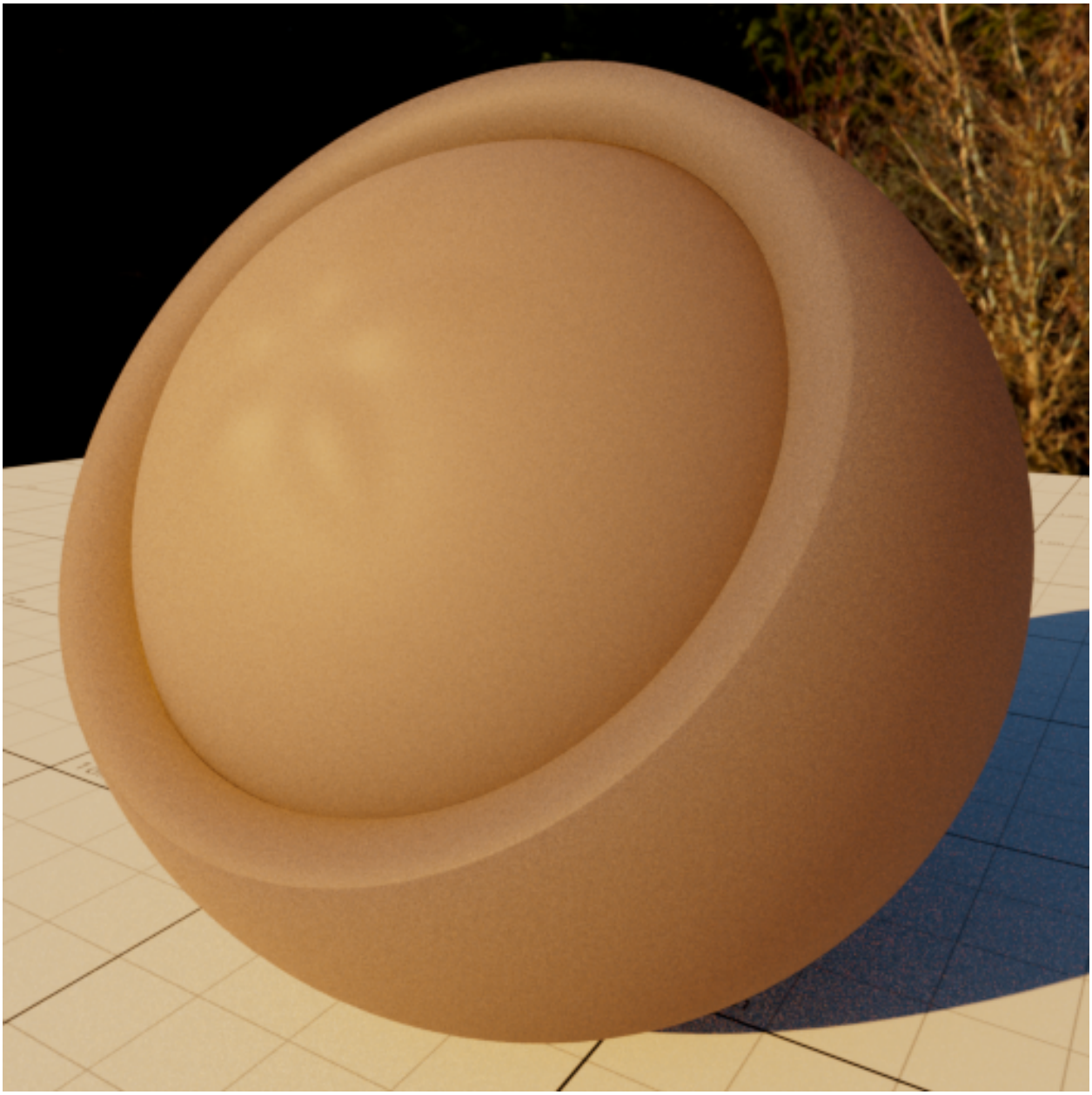


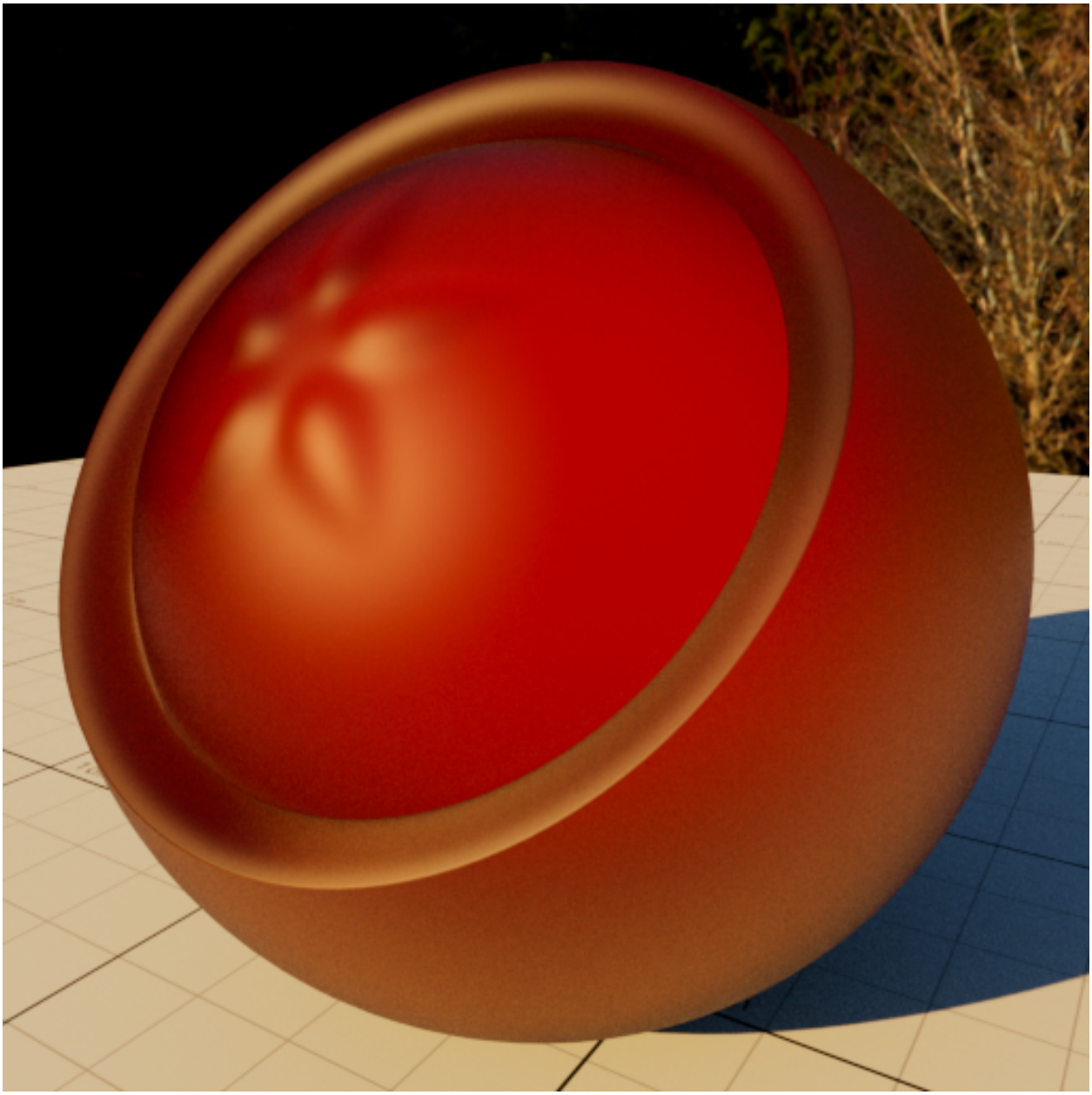












References

1.13 asSubsurface

A raytraced, no precomputed required, subsurface-scattering material with a coupled specular term. This node capable of modelling a wide range of appearance thanks to built-in support for several diffusion profiles.

1.13.1 Parameters

Subsurface Parameters

Subsurface Profile The diffusion profiles to use in the BSSDRF[1]. This parameter can take the following values

- Better Dipole
- Directional Dipole *[FHK14]*
- Gaussian *[dEonLE07] [DEonI11]*
- Normalized Diffusion *[Chr15]*
- Standard Dipole *[JMLH01]*, *[DJ05]*
- Random Walk *[MHD16]*

Reflectance The surface reflectance.

SSS Amount A scaling factor for the overall contribution of the subsurface scattering term.

Mean Free Path Mean free path[2] controls how deep light can travel within the volume, scattering internally, until it's fully absorbed or exits the medium. Lower values in the limit would have a response similar to a diffuse term, while high values would allow light to travel almost unhindered, producing a translucent like appearance.

MFP Scale An overall scaling factor for the *Mean Free Path*.

SSS Advanced Parameters

SSS Ray Depth The maximum ray depth allowed for a ray of type *subsurface*.

---

[1] See also bidirectional scattering distribution function.
[2] See mean free path wikipedia page for more details.

Fresnel Parameters

Index of Refraction The index of refraction of the material.

Fresnel Advanced Parameters

Fresnel Weight A value of 1.0, will scale the subsurface contribution by the Fresnel transmittance, while a value of 0.0 will disable any scaling. When using the provided specular BRDF, in order to keep energy conservation, this scaling factor should be set to 1.0. If you're not using the provided specular term and/or want to compose one later in your workflow, you can disable this. You also have the freedom to use intermediary values.

Specular Parameters

Specular Weight An overall scaling factor for the specular term. A value of 1.0 provides full intensity, 0.0 disabling it.

Note: This shader allows the user to texture map the specular weight, to control the specular term intensity, but it does not provide a way to *tint* or color the specular term. That is intentional sually only dielectrics[3] have subsurface scattering, and dielectrics have no tinted specular highlights.

Specular Roughness The apparent surface roughness of the material. The distribution used is the *GGX [WMLT07]*, and energy conservation to take into account multiple scattering *[HHdEonD16]* is applied automatically.

Anisotropy Parameters

Anisotropy Amount The overall weight of the anisotropy, with a value of 0.0 producing isotropic specular highlights, and a value of 1.0 producing full anisotropic specular highlights.

Anisotropy Angle A rotation angle in [0,1] range, that is mapped internally to a full 360 degrees rotation and applied on top of the anisotropy value provided by the explicit anisotropy vector or anisotropy vector map.

Anisotropy Mode The anisotropy mode, which can either be a anisotropy vector map with the XY anisotropy encoded in the *red* and *green* channels of the image, or an explicit anisotropy vector, which can be provided via a *asAnisotropyVectorField* node. It can therefore take the values

- Anisotropy Map
- Explicit Vector

Anisotropy Map The anisotropy vector map to use when *Anisotropy Mode* is set to *Anisotropy Map*.

Anisotropy Direction An explicit anisotropy vector to use when the *Anisotropy Mode* parameter is set to *Explicit Vector*.

---

[3] Dielectric is a material which is an electric insulator, the opposite of *conductors* which as the name says, conducts electricity. See this page on dielectric materials for more details. In terms of look development an accepted simplification is that dielectrics have white or non-tinted specular highlights, while conductors have tinted or coloured specular highlights.

Bump

Bump Normal The unit length world space normal of the bumped surface.

Matte Opacity

Enable Matte Opacity Parameter that toggles matte holdouts.

Matte Opacity Matte opacity scaling factor.

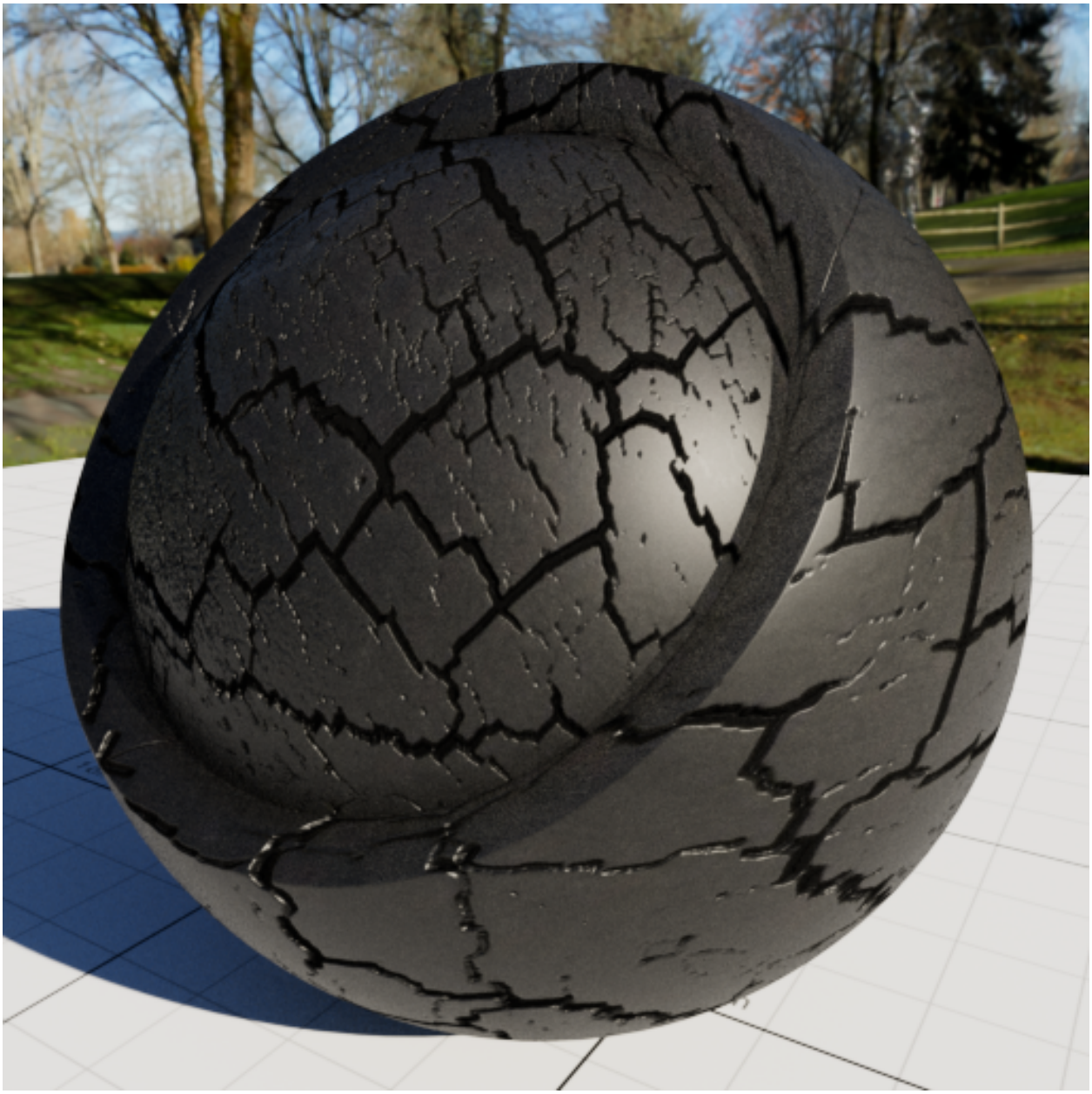
Matte Opacity Color Holdout color.

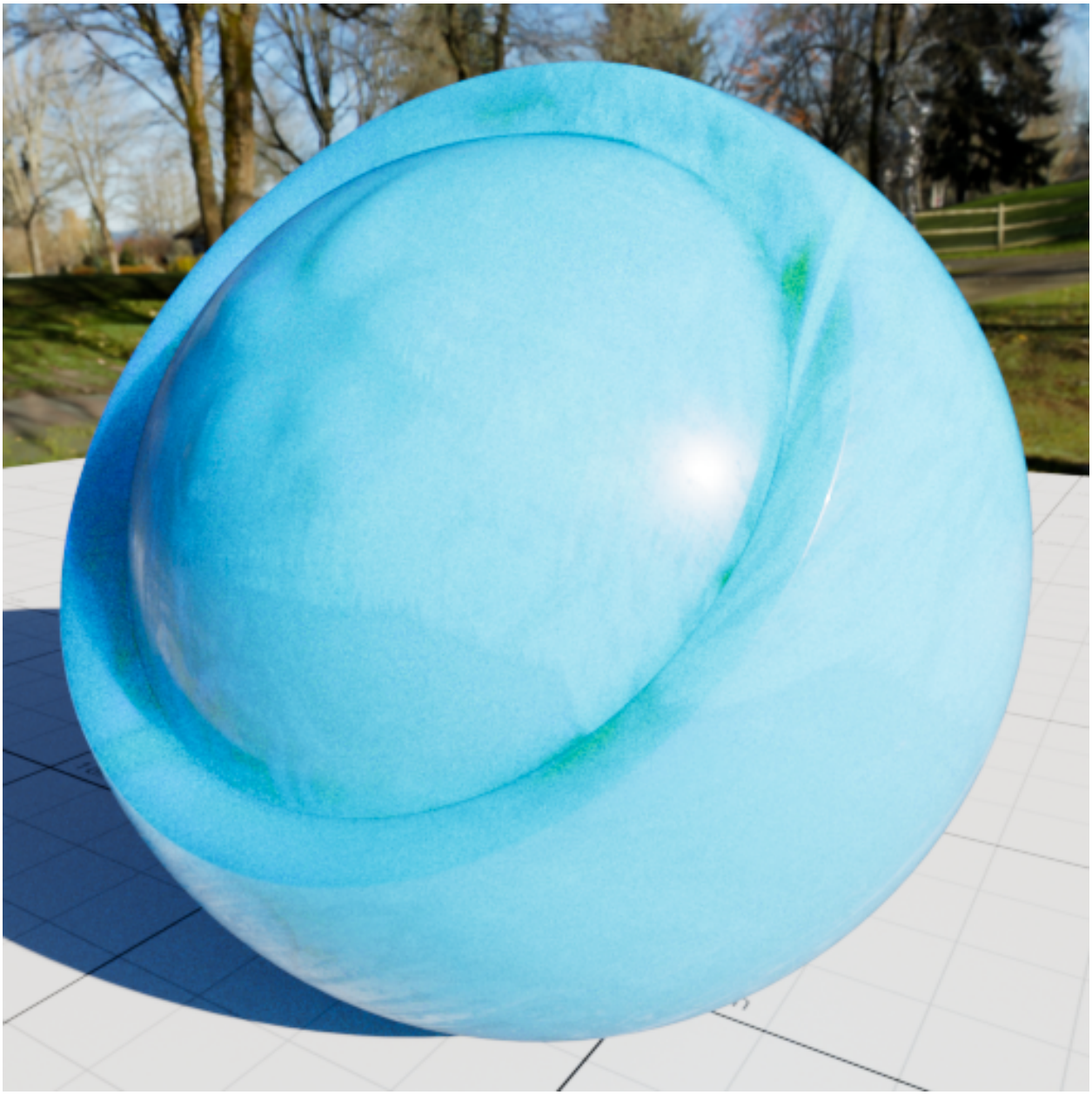
1.13.2 Outputs

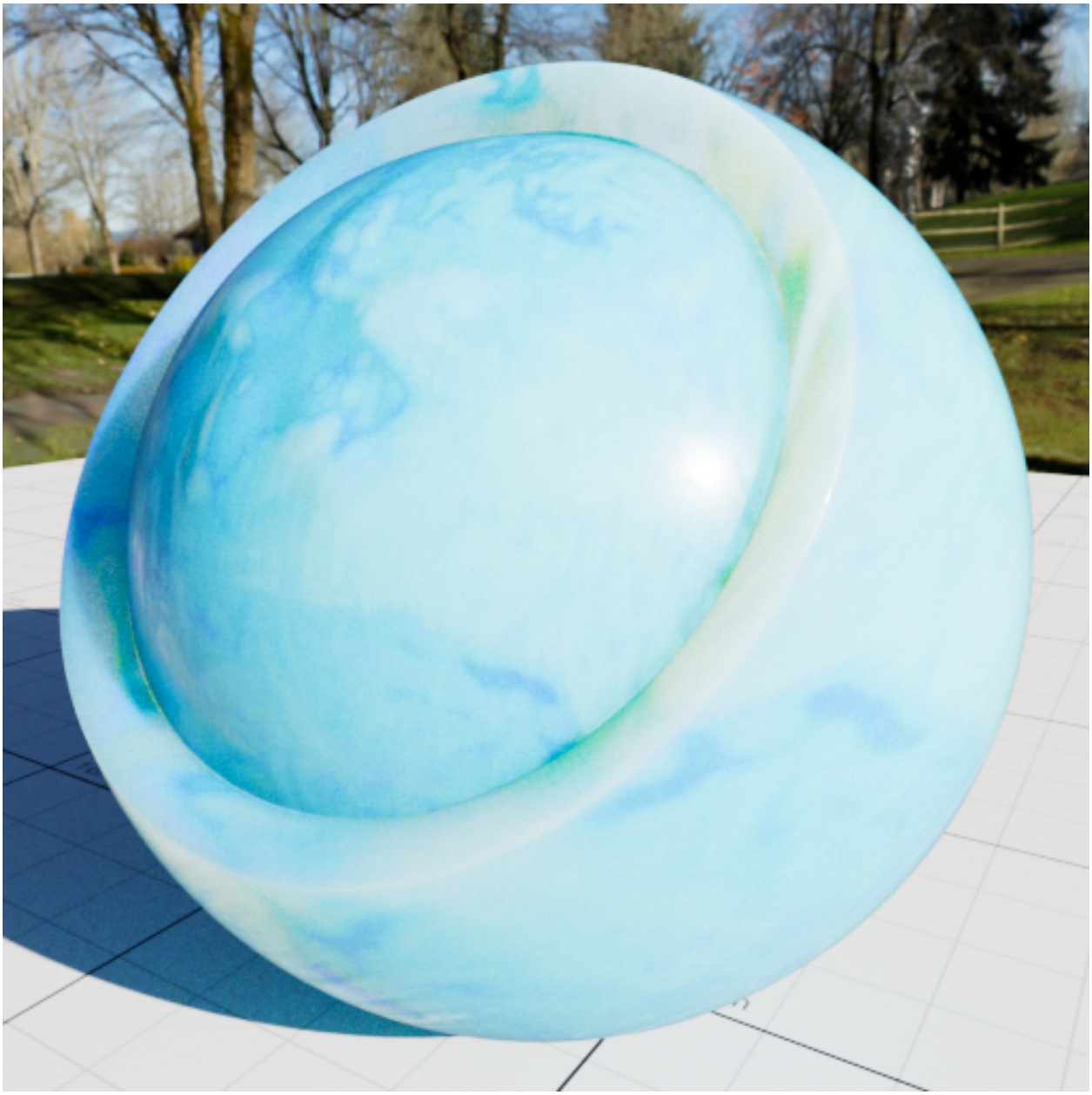
Output Color The BSSRDF output with the optional added specular BRDF.

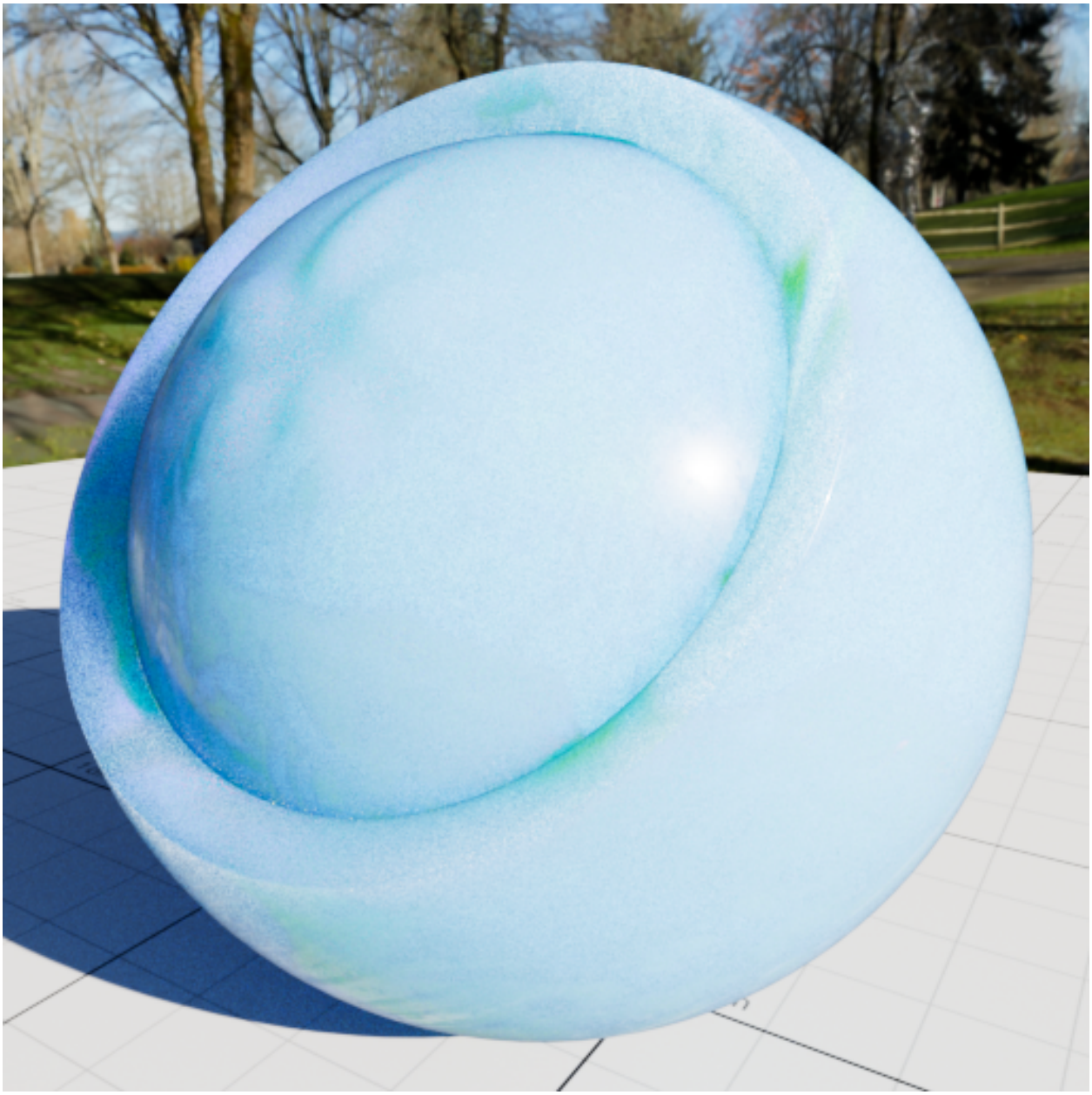
Output Matte Opacity The matte holdout.

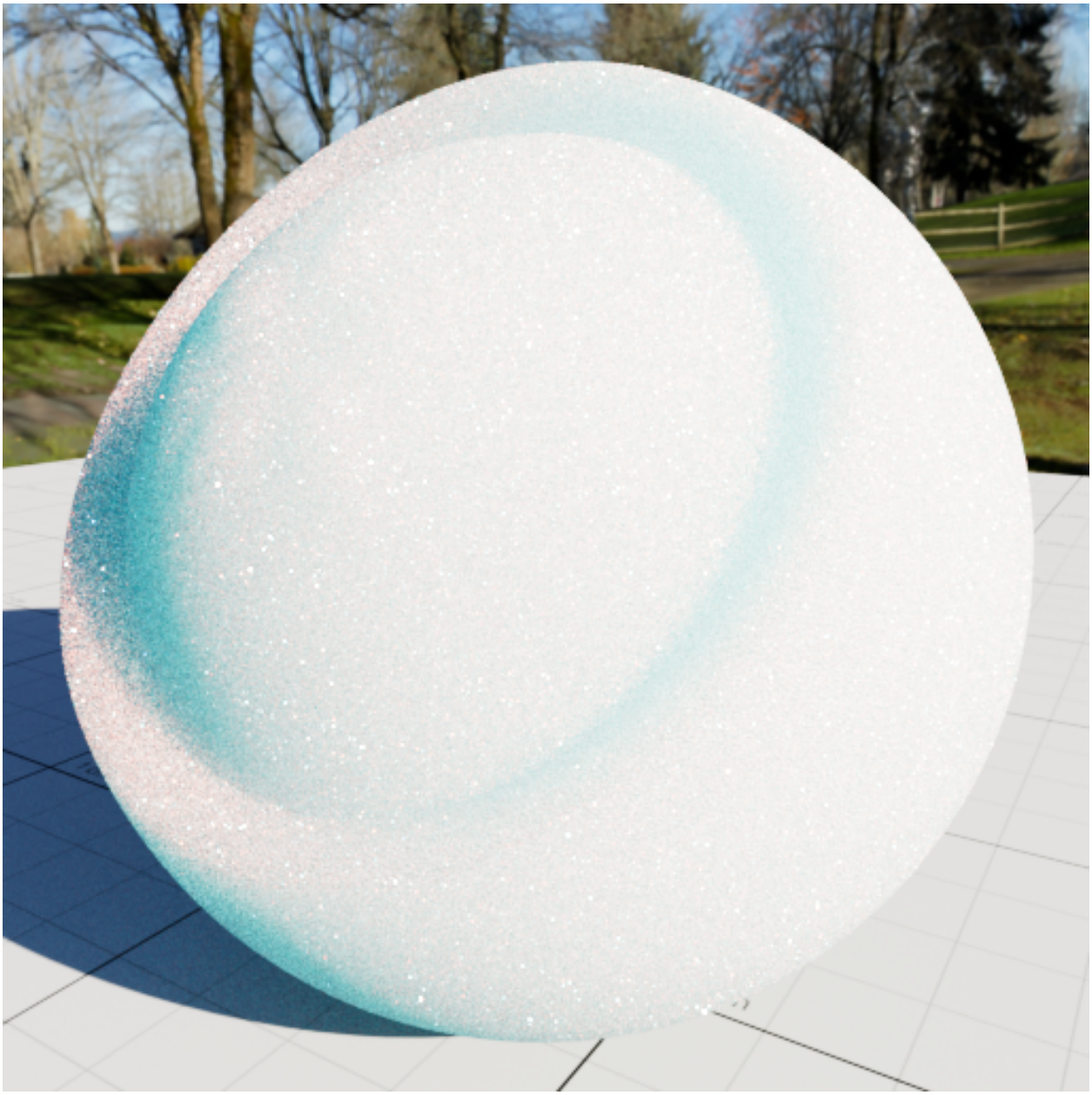
1.13.3 Screenshots

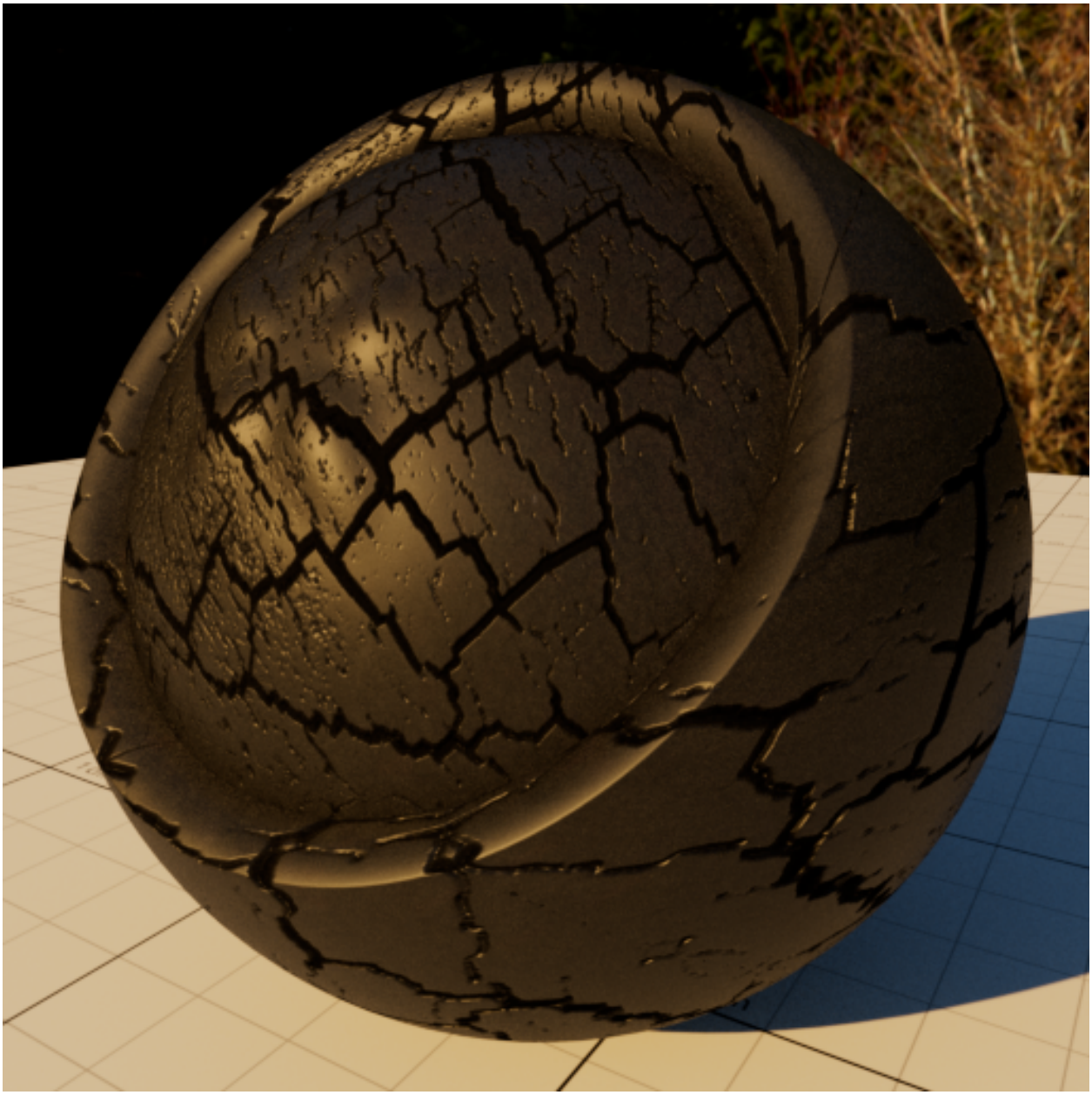


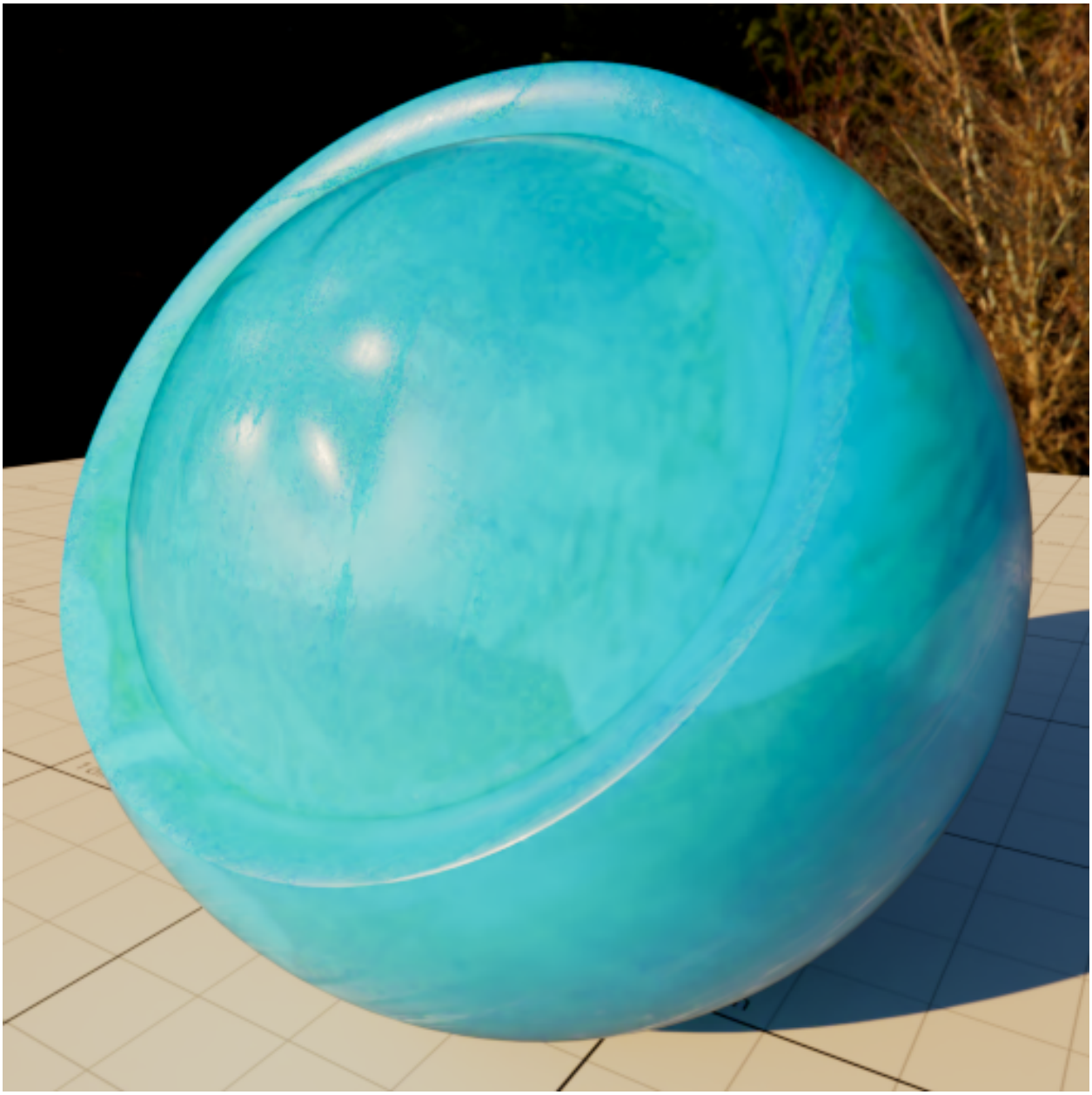


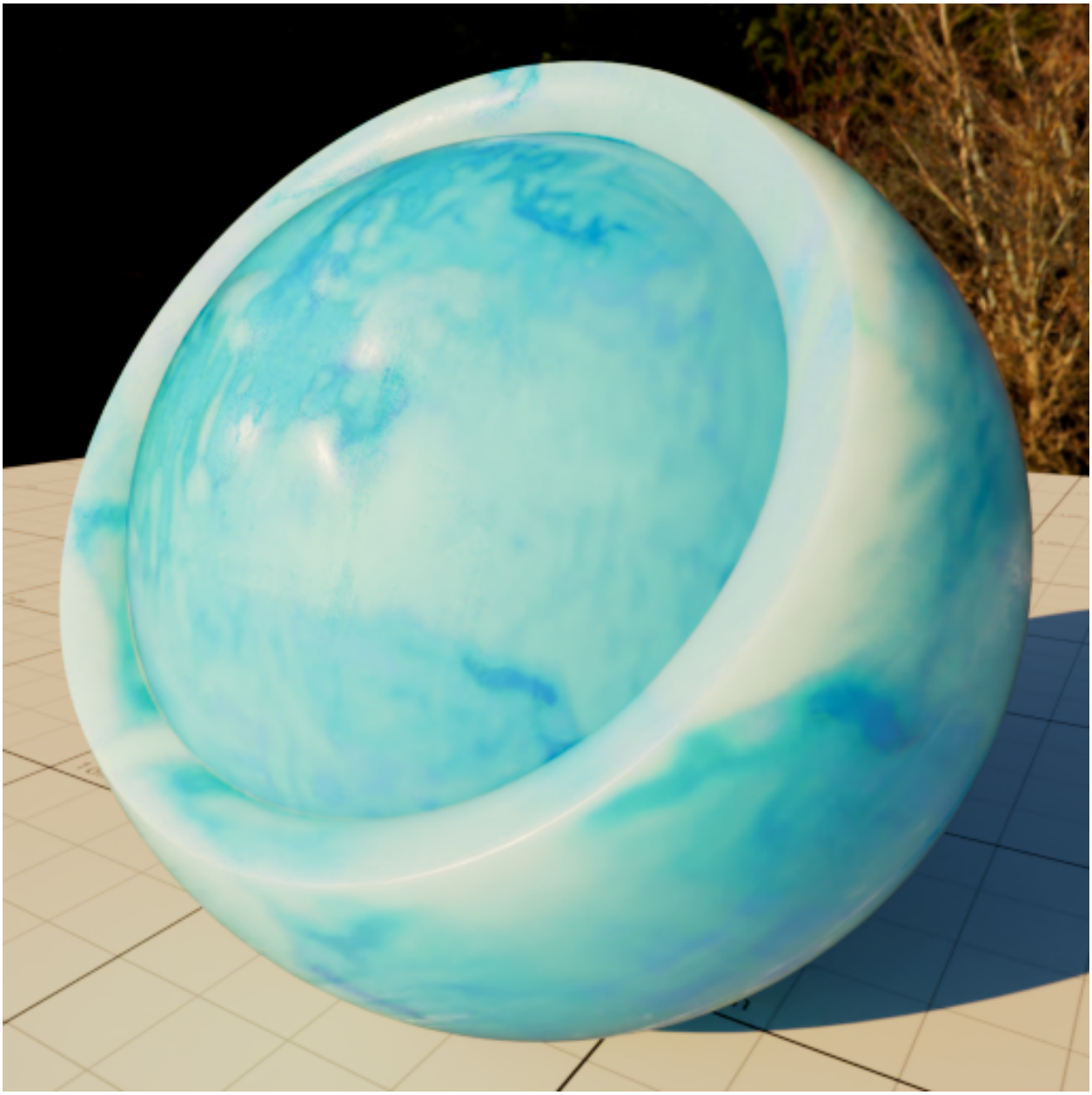


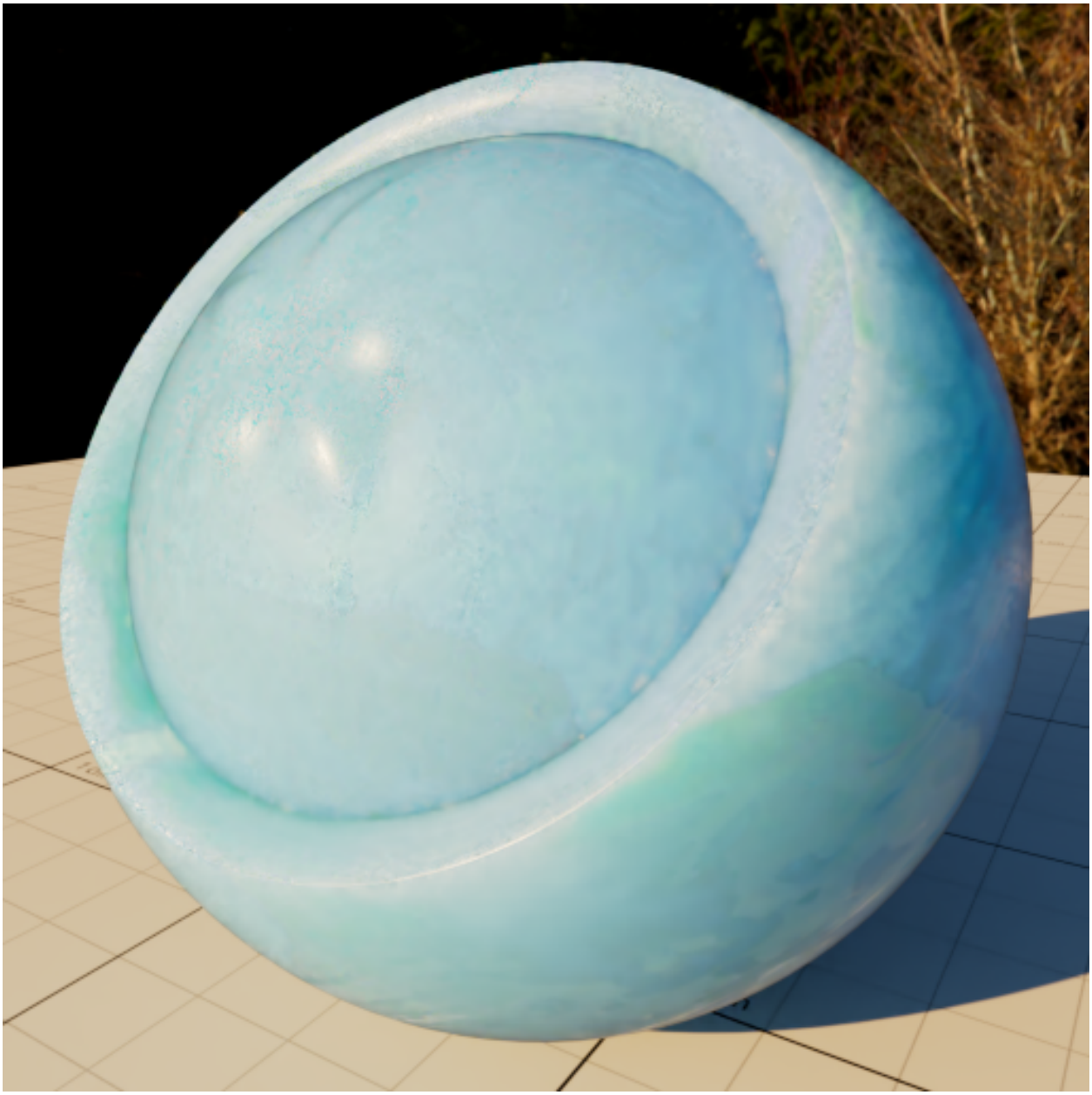












References

1.14 asSwitchSurface

A node that switches the output material from a list of 8 inputs, based on object or instance IDs, names, or string patterns, facilitating the creation of automatic variation in large scenes, specially when coupled with other variation nodes such as *asSwitchTexture*, *asVaryColor* or *asIdManifold*.

1.14.1 Parameters

Color Parameters

Input Color 0 The first material in the list of materials evaluate.

(. . .)

Note: The subsequent inputs follow exactly the same structure and parameterization.

Input Color 7 The last material in the list of materials to evaluate.

Cycle Mode When cycling through the list of materials, if the index goes outside the list bounds, then one has the choice to cycle back to the beginning of the list, or if to clamp to the list size.

Connected Slots Only When enabled (the default behaviour), it will *look up* and cycle through t he inputs which are connected to upstream material nodes. When disabled it will cycle through all available inputs, even those that aren't connected to anything at all.

Manifold Parameters

Manifold Type The type of manifold to use in order to determine which lookup to do in the list of materials. It can be one of the following

- Object Name
- Object Instance Name
- Assembly Name
- Assembly Instance Name

- Face ID
- String Prefix
- String Suffix
- Find String

String

Expression The expression[1] to search in the expression domain.

Domain The domain used to search an expression for.

Seed The seed[2] to use for the manifold.

1.14.2 Outputs

Output Color The resulting material color.

1.15 asToon

A Non-Photo-Realistic/NPR *[15]* toon shader.

[1] Regular expressions, or regex. If you're unfamiliar with it, it allows the creation of complex patterns for string and substring matching. You can validate your expressions here at regex101.

[2] A number used to initialize a pseudo random number generator, to allow some degree of determinism in a system. See random seed for more information.

1.15.1 Parameters

Incandescence Parameters

Incandescence Weight The amount of incandescence, not restricted to an upper bound of 1.0.

Incandescence Attenuation The amount of attenuation for the incandescence, based on the facing ratio towards the viewer. A high value will darken the values towards the edges.

Toon Control

Incandescence Tint The incandescence color.

Blending Mode The blending mode for the *Incandescence Tint* parameter. It can be one of the following modes:

- Darken
- Multiply
- Color Burn
- Linear Burn
- Lighten
- Screen
- Color Dodge
- Linear Dodge
- Overlay
- Soft Light
- Hard Light
- Vivid Light
- Linear Light
- Pin Light
- Difference
- Exclusion
- Subtract
- Divide
- Hue
- Saturation
- Color
- Luminosity

See also:

The node *asBlendColor* for more details.

EDF Dark Color The color to use in the darkest areas of this term[1].

EDF Dark Level The threshold at which to transition from darkest tones to mid-tones.

EDF Midtone Color The color to use for the mid-tones.

EDF Midtone Level The threshold at which to transition from mid-tones to bright areas.

EDF Bright Color The brightest color.

EDF Softness The transition softness between the tonal areas.

Advanced

Normalize by Area When unchecked, the incandescence term is constant. When checked, it's scaled by the surface area of the object. With deforming or scaling objects, the incandescence might therefore change in order to keep the same energy level being emitted per surface area.

Diffuse Parameters

Diffuse Weight The contribution of the diffuse term to the toon shading, between 0 and 1.

Toon Control

Diffuse Tint The color for the diffuse term.

Blending Mode The blending mode for the *Diffuse Tint* parameter. It can be one of the following modes:

- Darken
- Multiply
- Color Burn
- Linear Burn
- Lighten
- Screen
- Color Dodge
- Linear Dodge
- Overlay
- Soft Light
- Hard Light
- Vivid Light
- Linear Light
- Pin Light
- Difference
- Exclusion

[1] Emittance Distribution Function.

- Subtract
- Divide
- Hue
- Saturation
- Color
- Luminosity

See also:

The node *asBlendColor* for more details.

Shadow Color The color to use in the shadow areas, or umbra.

Shadow Level The threshold at which to transition from shadow to mid-tones.

Midtone Color The color to use for the mid-tones, or penumbra.

Midtone Level The threshold at which to transition from mid-tones to bright areas.

Highlight Color The bright areas color.

Diffuse Softness The transition softness between the tonal areas.

Advanced

Diffuse Ray Depth The maximum number of bounces allowed for *diffuse* paths.

Specular Parameters

Specular Weight The amount of specular highlights to add to the shading, between 0 and 1.

Specular Roughness The apparent surface roughness of the specular highlights. This works in conjunction with the *Specular Softness* parameter to determine the softness of transition from highlights to non lit area. If you want soft highlights, increase roughness and *Specular Softness*. If you want bigger yet sharper highlights, increase roughness and keep the *Specular Softness* low.

Index of Refraction The index of refraction for the specular term, with high values increasing the reflectivity, and low values decreasing it. This also works in conjunction with the *Specular Softness* control.

Anisotropy

Anisotropy Amount Overall intensity of the anisotropy, with a value of 0.0 representing a isotropic specular highlight, and 1.0 fully anisotropic along the main anisotropy direction.

Anisotropy Angle Rotation angle for the anisotropic highlight in [0,1], mapping a rotation from 0 to 360 degrees.

Anisotropy Mode Toggles between accepting a direct texture map in the form of an anisotropy vector map, or between an explicit vector (or a connection to a node that generates such a vector[2]). It can take the values

- Anisotropy Map
- Explicit Vector

[2] Such as the *anisotropy vector field node*.

Anisotropy Map Also known as tangent field, encodes the anisotropy directions along X and Y in the Red and Green or Red and Blue channels of the image. Appleseed expects values encoded in the Red and Green channels. Valid when the *Anisotropy Mode* is set to *Anisotropy Map* only.

Anisotropy Direction The explicit vector passed as the anisotropy direction. Valid when the *Anisotropy Mode* is set to *Explicit Vector* only.

Toon Controls

Specular Tint The color for the specular term.

Blending Mode The blending mode for the *Specular Tint* parameter. It can be one of the following modes:

- Darken
- Multiply
- Color Burn
- Linear Burn
- Lighten
- Screen
- Color Dodge
- Linear Dodge
- Overlay
- Soft Light
- Hard Light
- Vivid Light
- Linear Light
- Pin Light
- Difference
- Exclusion
- Subtract
- Divide
- Hue
- Saturation
- Color
- Luminosity

See also:

The node *asBlendColor* for more details.

Glossy Color The color to use for the specular highlights.

Glossy Level The threshold at which to transition from specular highlights to areas without specular highlights.

Glossy Softness The transition softness between the highlights and areas without highlights. This works in conjunction with *Specular Roughness*, since a very rough specular highlight will still look sharp if the threshold between the lit areas have a harsh transition. On the other side this gives the user the ability to increase the size of the specular highlights, or decrease them, and vary their apparent softness independently.

Facing Attenuation A viewer Fresnel based attenuation factor. Higher values decrease the intensity of the specular reflection towards the viewer.

Advanced

Specular Ray Depth The maximum number of specular bounces allowed for *specular* paths.

Rim Lighting Parameters

Rim Weight Amount of contribution of the rim lighting effect.

Rim Tint The color for the rim lighting contribution blend.

Blending Mode The blending mode for the *Rim Tint* parameter. It can be one of the following modes:

- Darken
- Multiply
- Color Burn
- Linear Burn
- Lighten
- Screen
- Color Dodge
- Linear Dodge
- Overlay
- Soft Light
- Hard Light
- Vivid Light
- Linear Light
- Pin Light
- Difference
- Exclusion
- Subtract
- Divide
- Hue
- Saturation
- Color
- Luminosity

See also:

The node *asBlendColor* for more details.

Rim Softness The softness of the rim lighting effect.

Bump Parameters

Bump Normal The unit length world space normal of the bumped surface.

Specular Normal When using separate bump controls for the diffuse and specular terms, this is the unit length world space normal for the specular term.

Bump Control This allows the user to choose a single bump effect for both the diffuse and specular terms, or a separate bump effect for these terms. When using both, the *Bump Normal* parameter affects both terms. It can take the following values accordingly:

- *Diffuse Affects Both*
- *Split Bump*

Transparency Parameters

Transparency The transparency color, affecting shadow tinting as well.

Contour Parameters

Contour Color The overall color of the outlines.

Contour Opacity The overall opacity of the outlines.

Contour Width The width of the outlines on the object.

Contour Object Generate the contours based on the object ID.

Contour Material Generate contours based on the material ID. Objects with the same material ID will share contours.

Contour Occlusion Generate contours based on depth differences between nearby points.

Contour Creases Generate contours based on the creases of the object[3], outlining sudden changes of the surface.

Occlusion Threshold The threshold value for the depth difference comparison between nearby points

Crease Threshold The threshold value for the creases based outlining, where low values result in outlines for smaller changes in the surface of the object.

---

[3] Based on the partial derivatives of the surface normal N along the U and V directions.

Matte Opacity Parameters

Enable Matte Opacity Parameter that toggles matte holdouts.

Matte Opacity Matte opacity scaling factor.

Matte Opacity Color Holdout color.

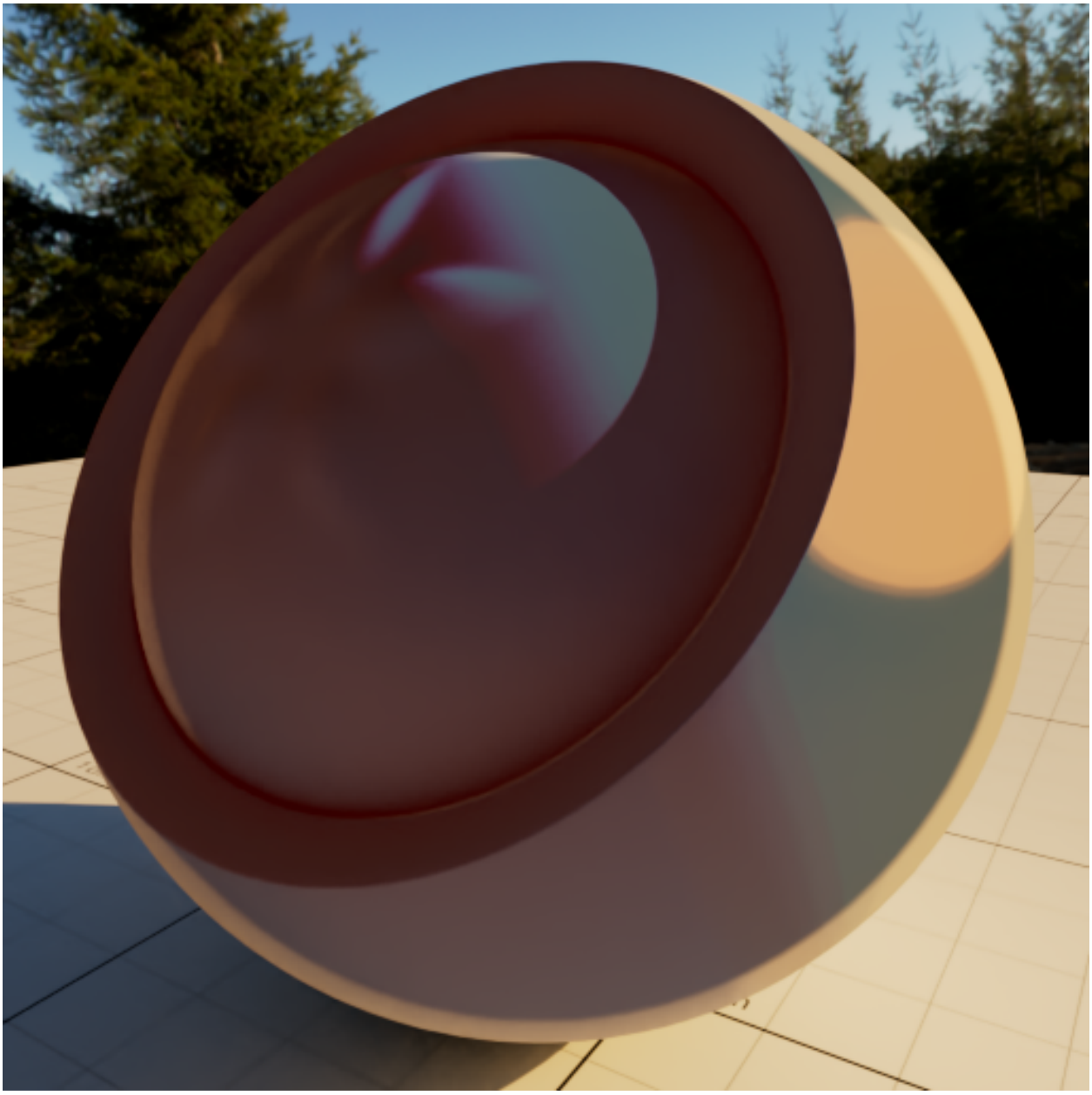
1.15.2 Outputs

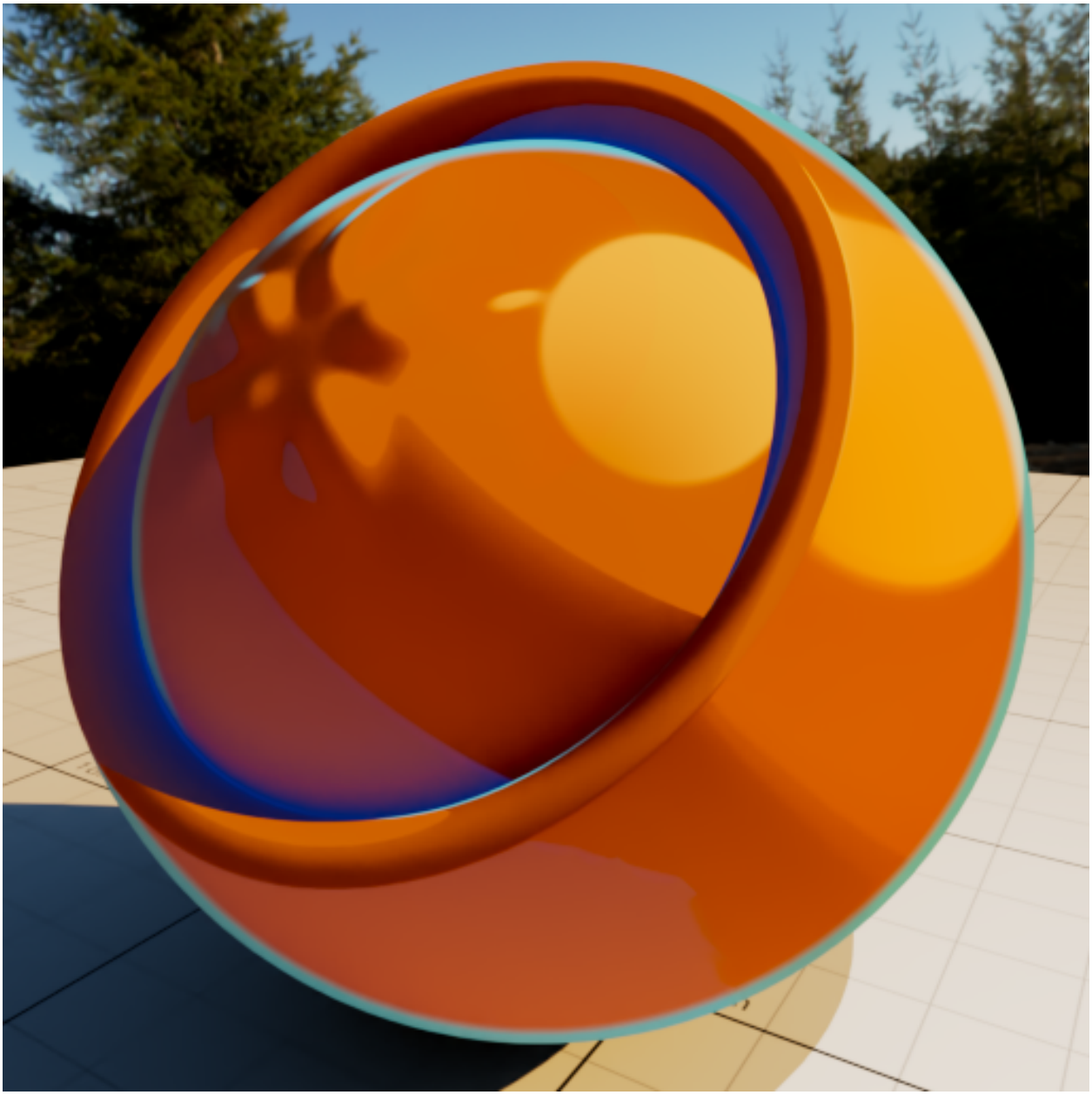
Output Color The toon BRDF output color.

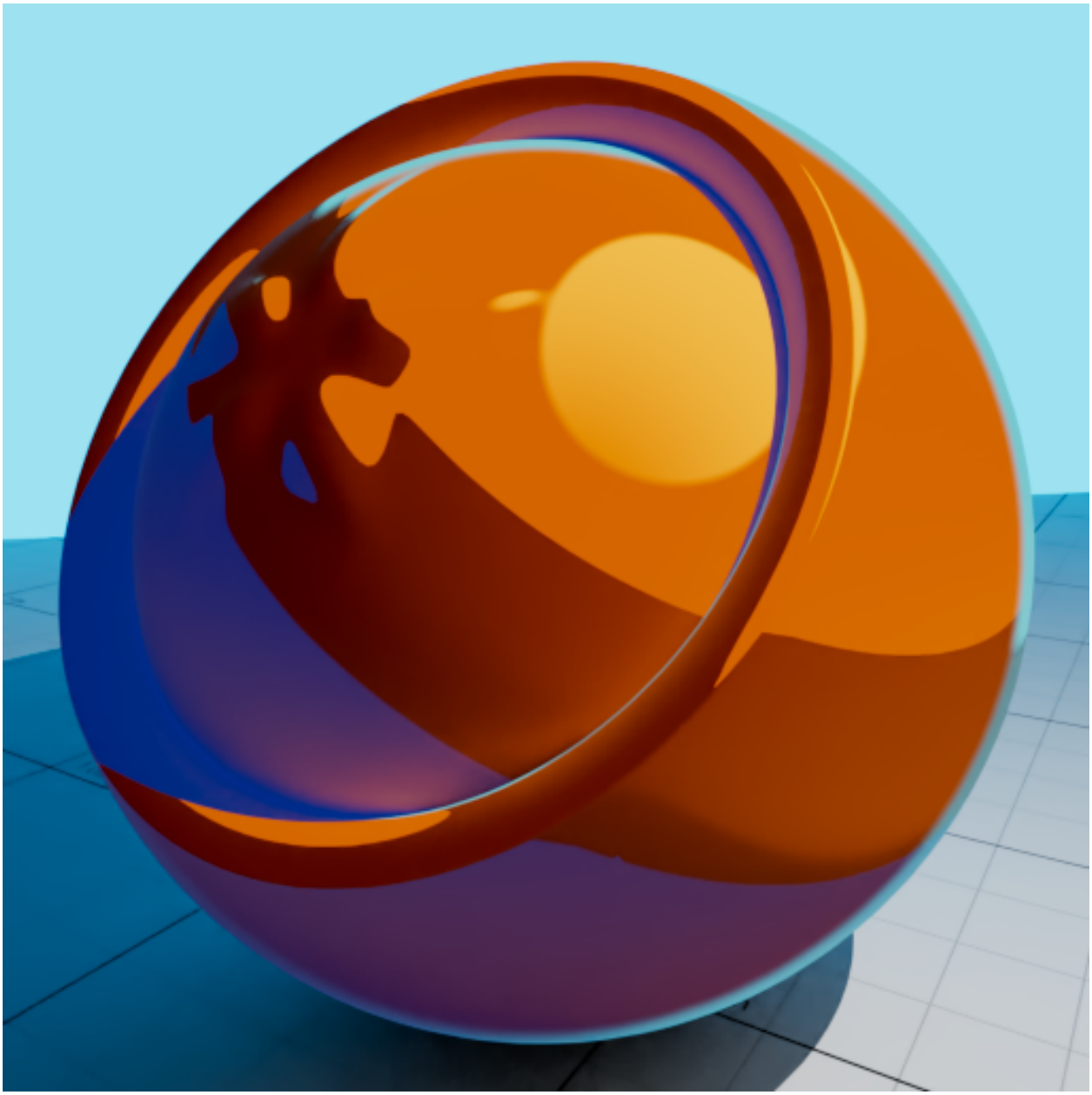
Output Transparency The resulting transparency color.

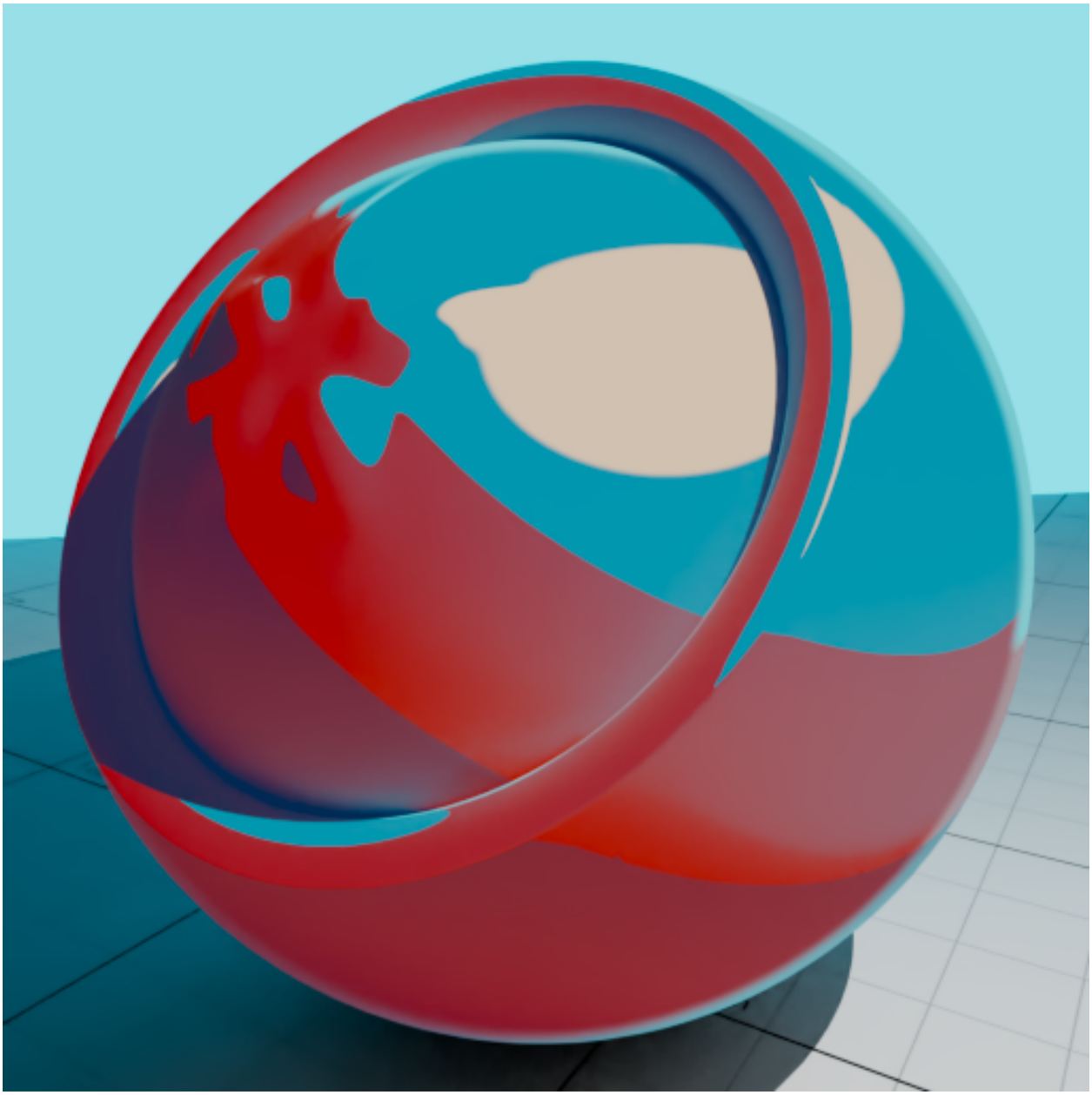
Output Matte Opacity The matte holdout.

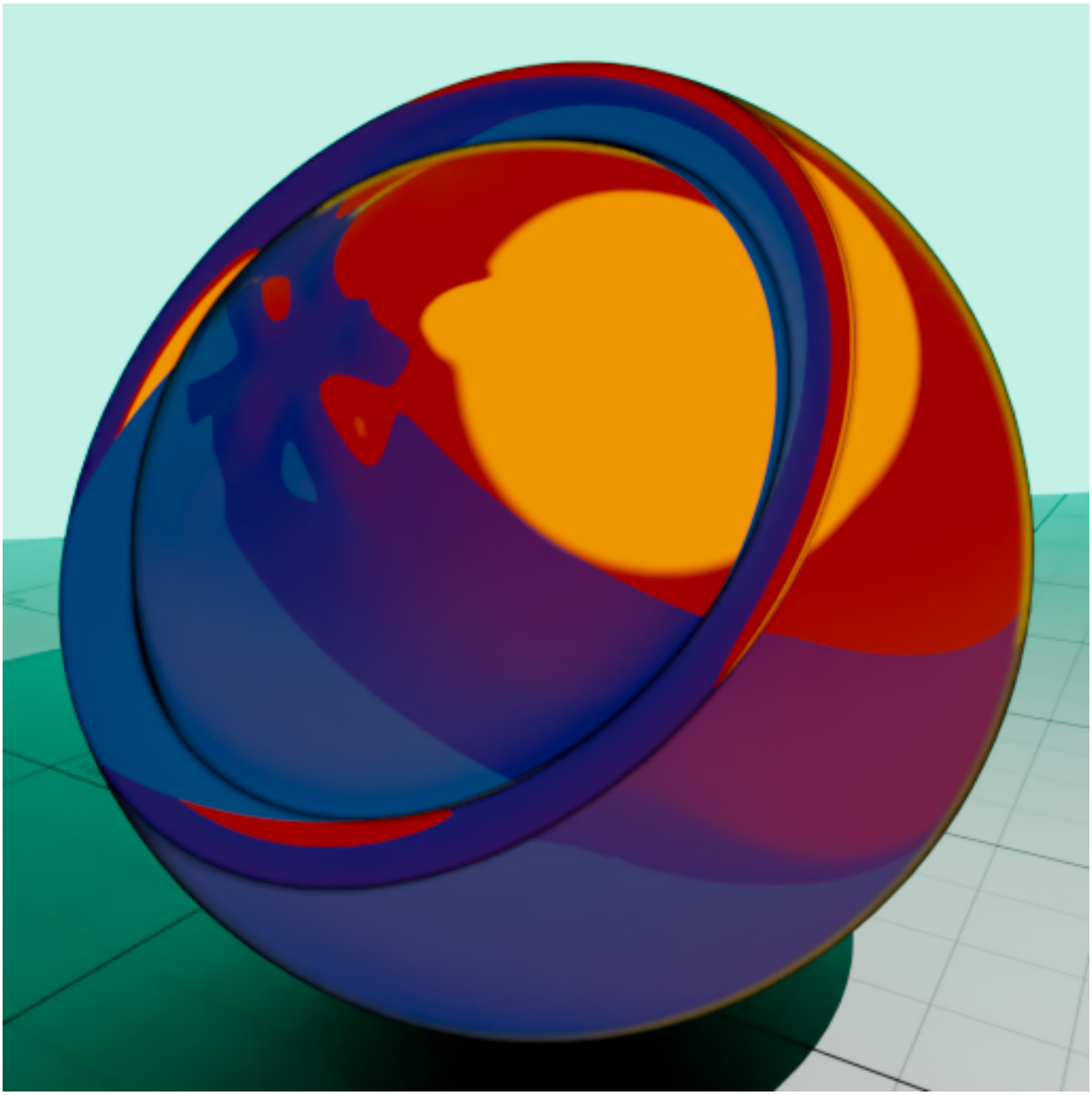
1.15.3 Screenshots

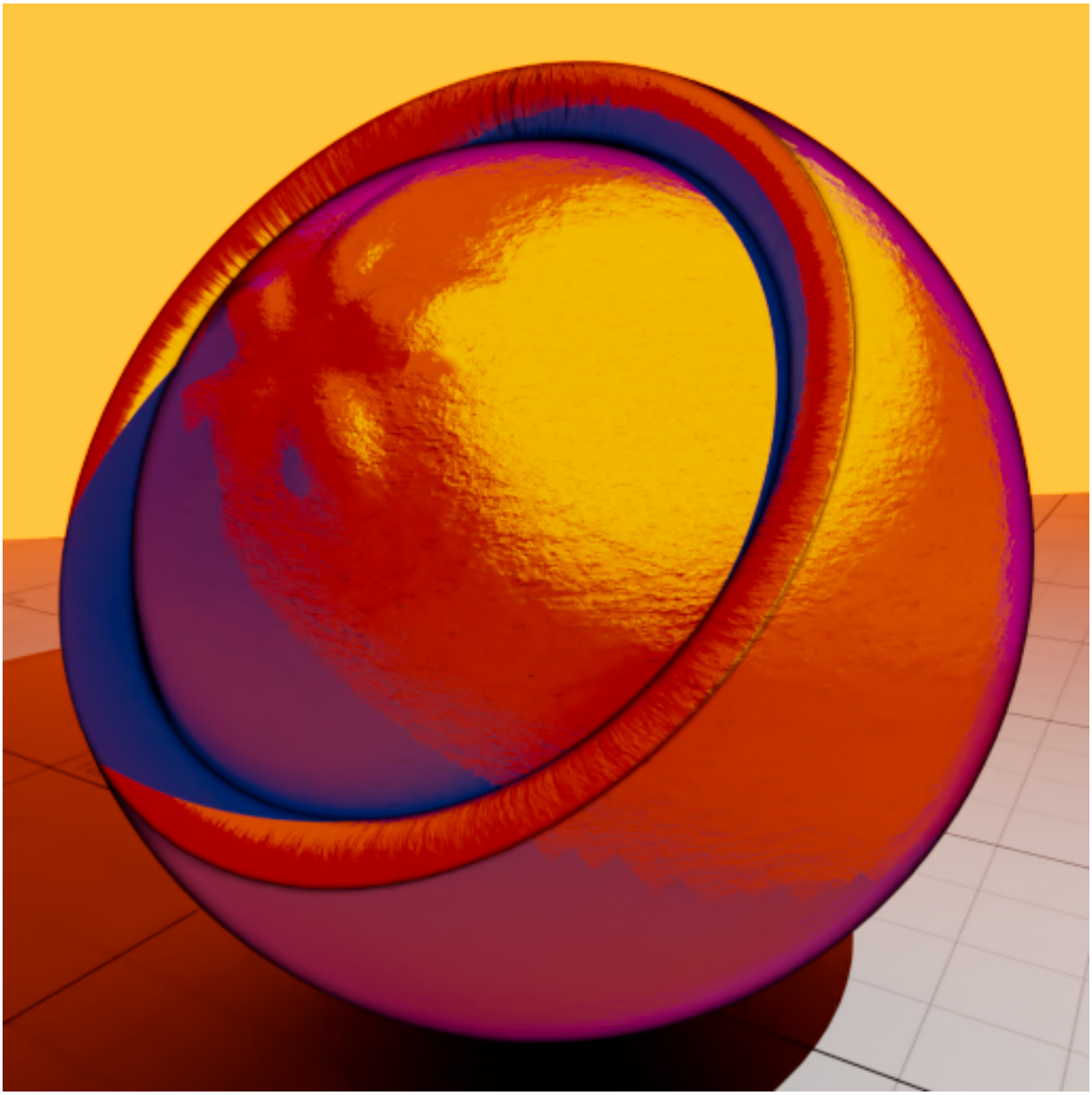


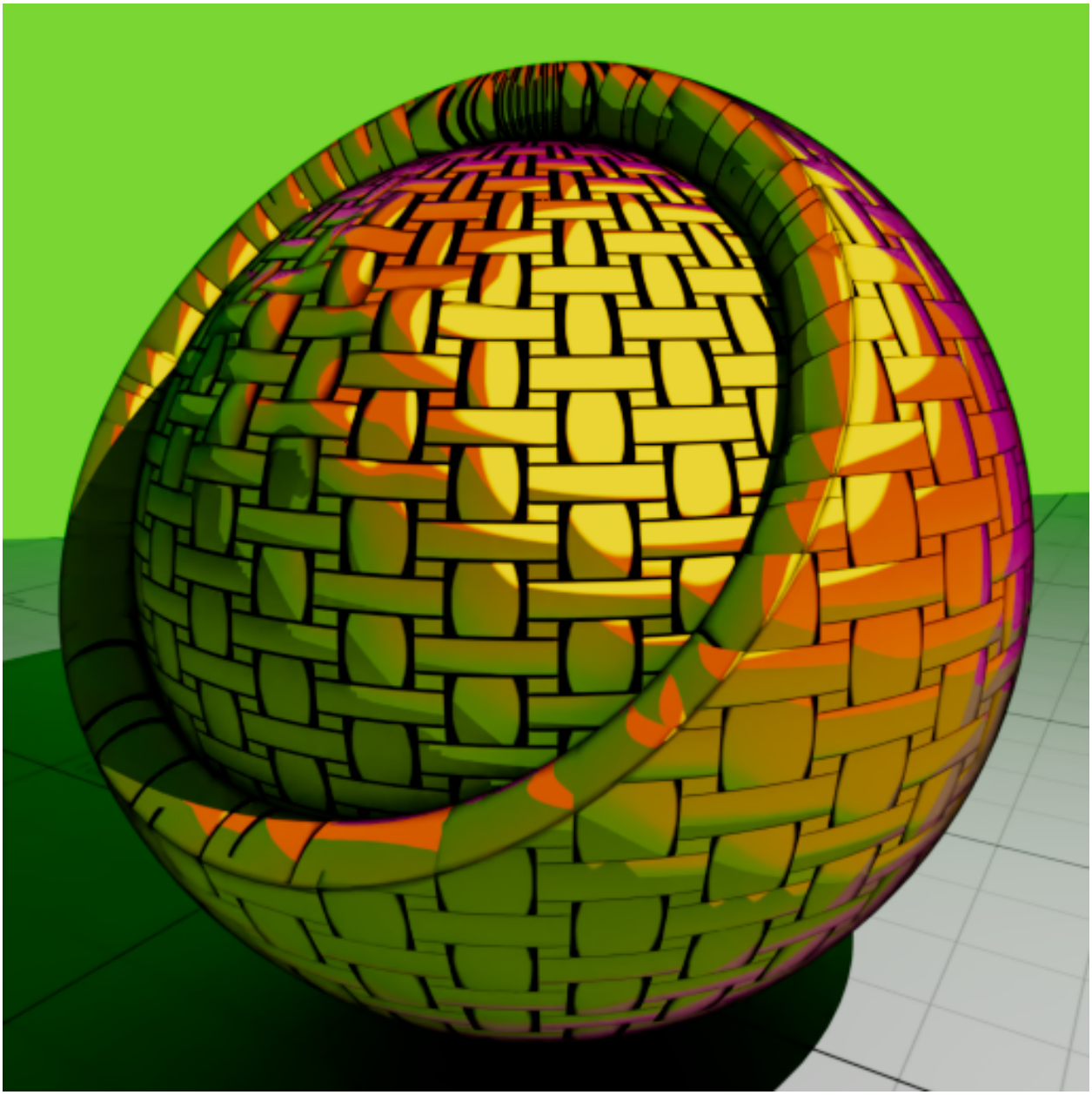


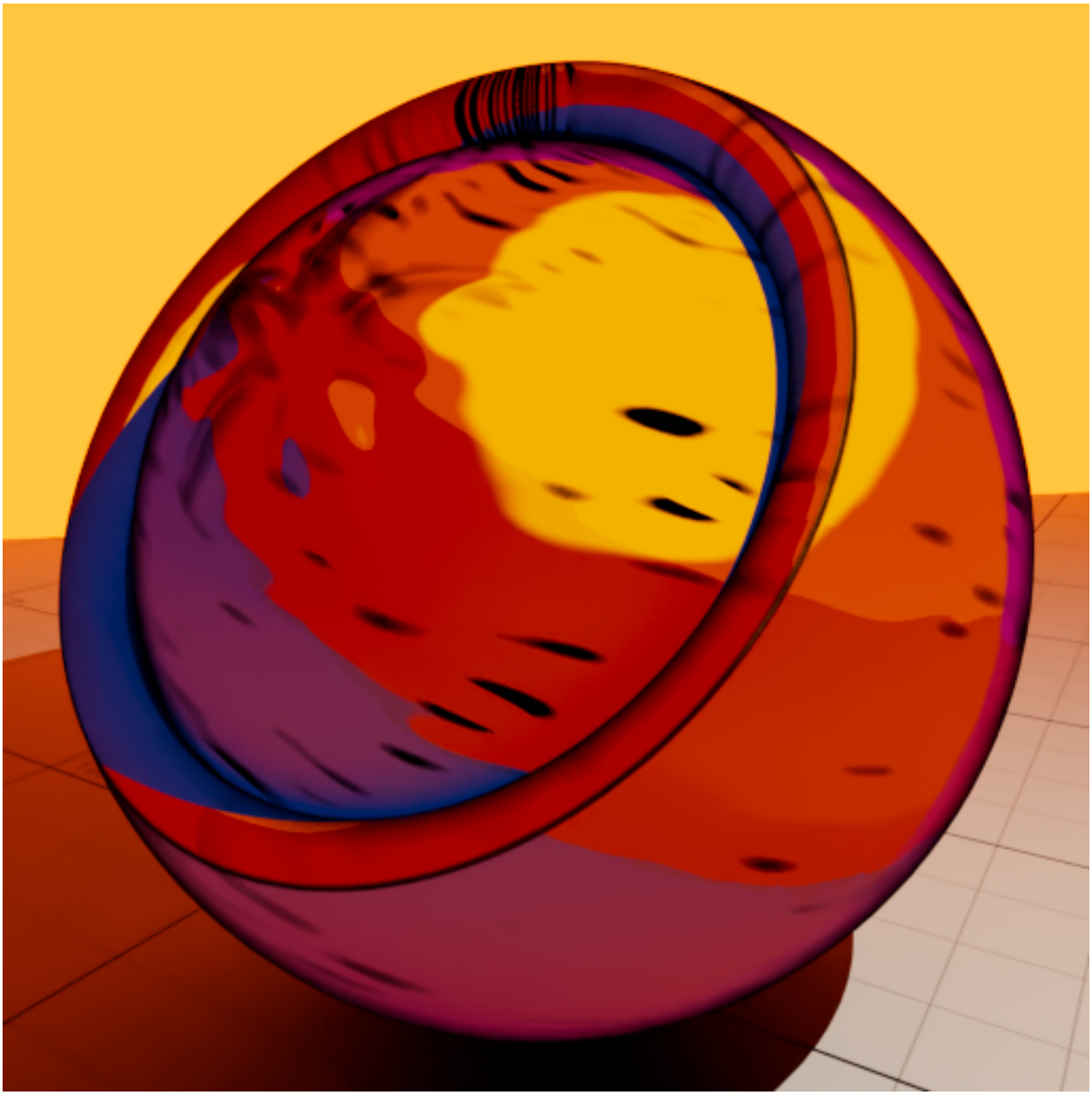


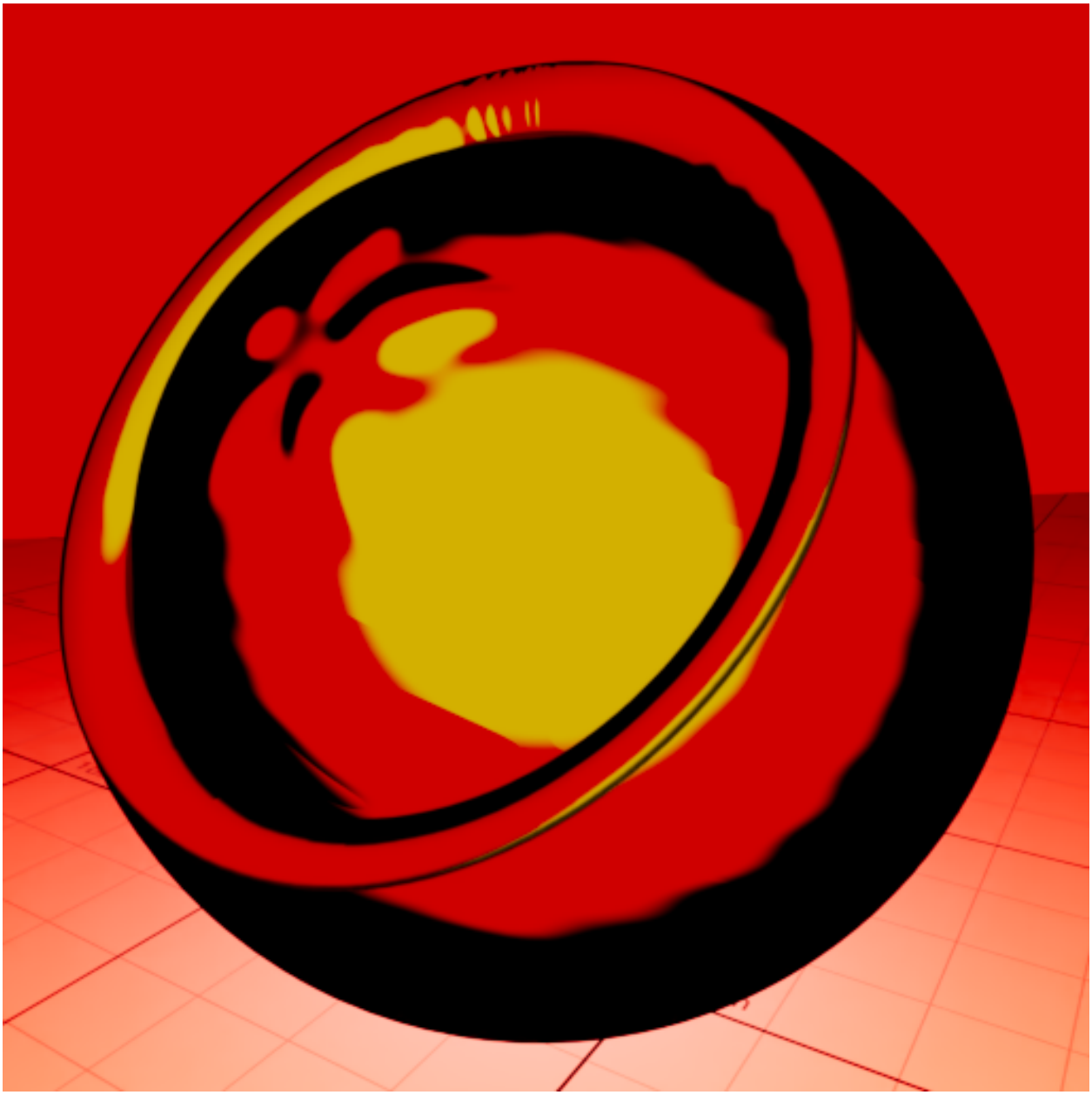












References

1.16 asNoise2D

A fractal noise node, with recursion, and an ample choice of noise primitives.

1.16.1 Parameters

Color Parameters

Color 1 Primary color.

Color 2 Secondary color.

Contrast Contrast between primary and secondary colors.

Noise Parameters

Type The noise primitive used. It can take the following values

- Perlin *[Per85]*
- Simplex *[Per02]*
- Value
- Voronoise[1] *[Wor96]*
- Gabor *[GLLD12]*, *[LLDDutre09]*

Intensity The global noise intensity.

X Frequency The noise frequency along the *x* direction.

Y Frequency The noise frequency along the *y* direction.

Ridges Toggling this checkbox will enable the *Ridged Noise* mode. A noise mode more suited to modelling the appearance of crests, mountains, when used to drive a displacement or bump map.

Inflection Enabling this checkbox will force the noise function to return a absolute value (if the noise was in [-1,1] range to begin with, otherwise it won't have any effect).

[1] Also known as generalized Voronoi. See Inigo Quilez article on voronoise.

Signed Noise Enabling this checkbox makes the noise function return values in the [-1,1] range, and disabling it will return values in [0,1] range.

Motion Parameters

Animated Noise Enabling this checkbox will animate the noise along time.

Frame Time Frame time, typically the frame number.

Time Scale Global time scale, affects the frame time.

Periodic Parameters

Periodic Enabling this checkbox will enable periodic noise, with a user-set *x* and *y* period.

X Period The *x* period when using periodic noise.

Y Period The *y* period when using periodic noise.

Voronoise Parameters

Smoothness Controls the smoothness of the generalized Voronoi noise, with low values having a sharp cell boundary, and high values having a smooth Perlin noise like appearance.

Jittering Controls the jittering of the Voronoi cells, with low values producing a regular cell grid, and high values producing a randomized cell grid.

Gabor Parameters

Anisotropy This parameter controls the type of Gabor noise used. It can take the values

- Isotropic
- Anisotropic
- Hybrid

Direction Anisotropy vector to use when the Gabor noise *Anisotropy* mode is set to *Anisotropic*.

Bandwidth The bandwidth for the Gabor noise.

Impulses The number of impulses for the Gabor noise.

Filter Noise Enabling this checkbox will produce antialiased noise.

Recursion Parameters

Amplitude Initial noise amplitude before recursion.

Octaves The number of interations to perform.

Cascade The type of iteration to perform. It can be

- Additive
- Multiplicative

In the first case, the results of each iteration are accumulated, and in the second case, they are multiplied with the previous product.

Lacunarity Control for the gap between successive noise frequencies (sucessive octaves).

Offset Controls the multifractality.

Gain Controls the contrast of the fractal.

Distortion This parameter distorts the domain of the coordinates for each frequency.

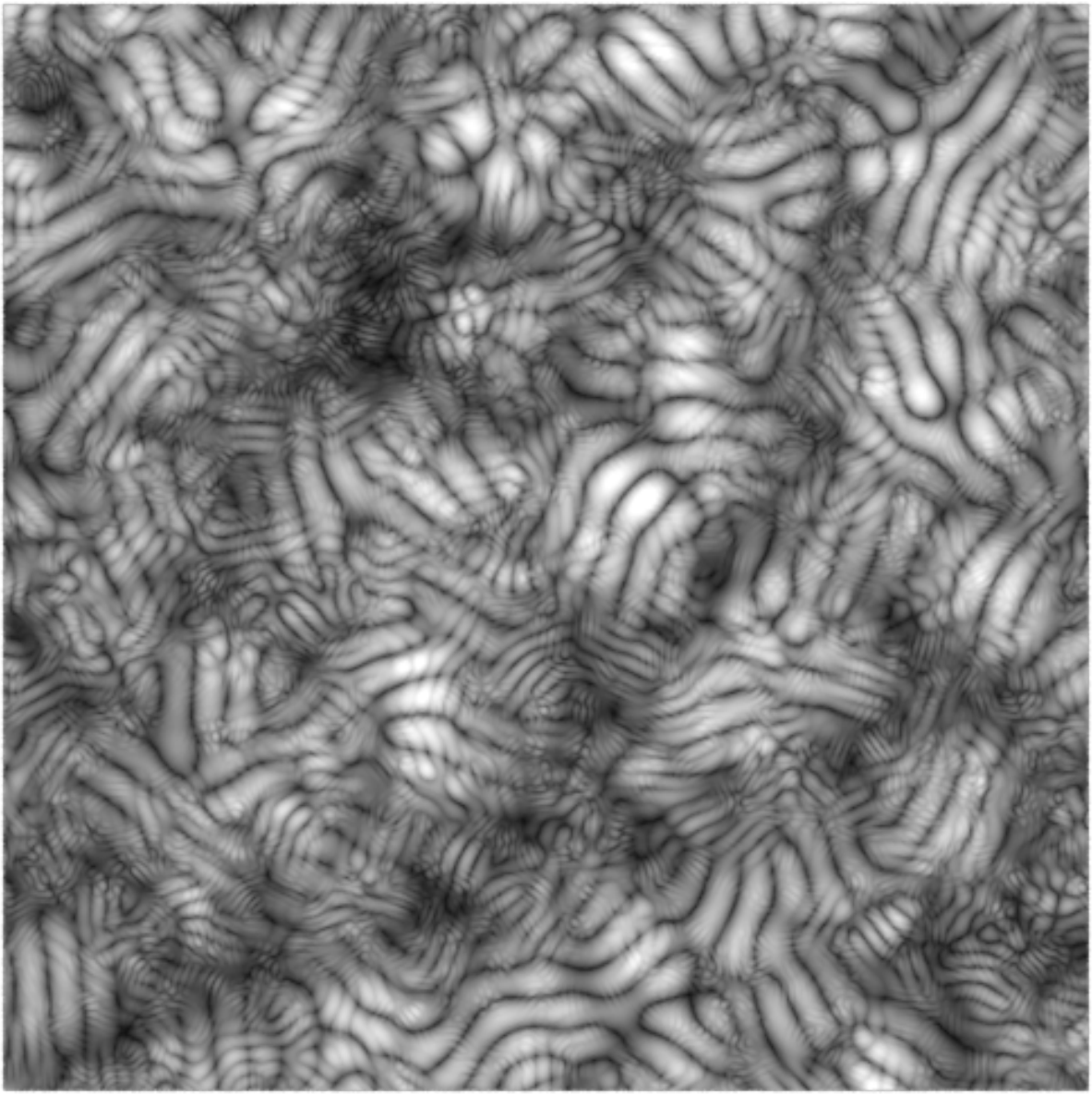
Outputs

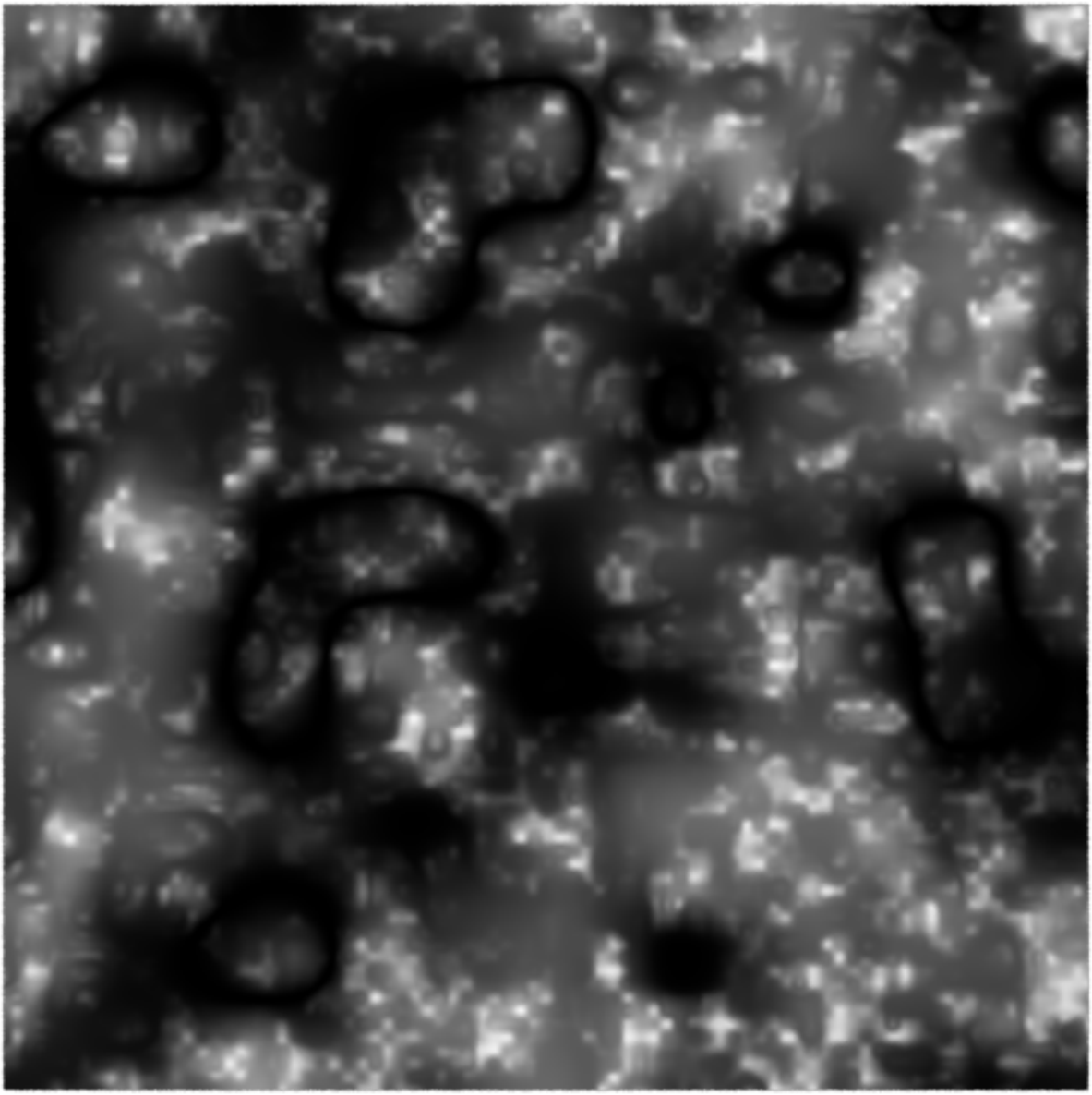
Output Color The color resulting from ghe *Features Mode* choice.

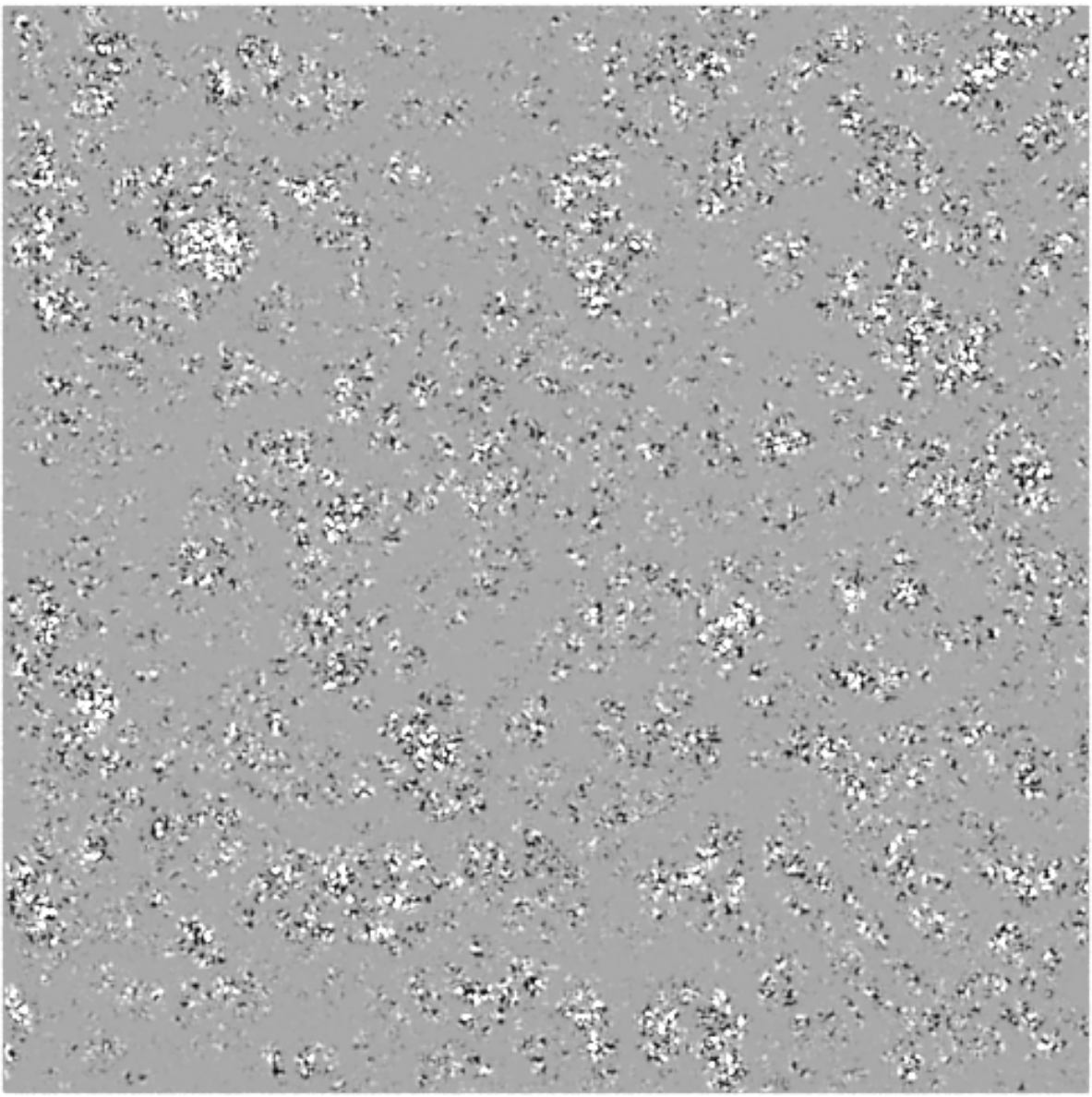
Output Alpha The alpha resulting from the *Features Mode* choice, usually luminance of the color only.

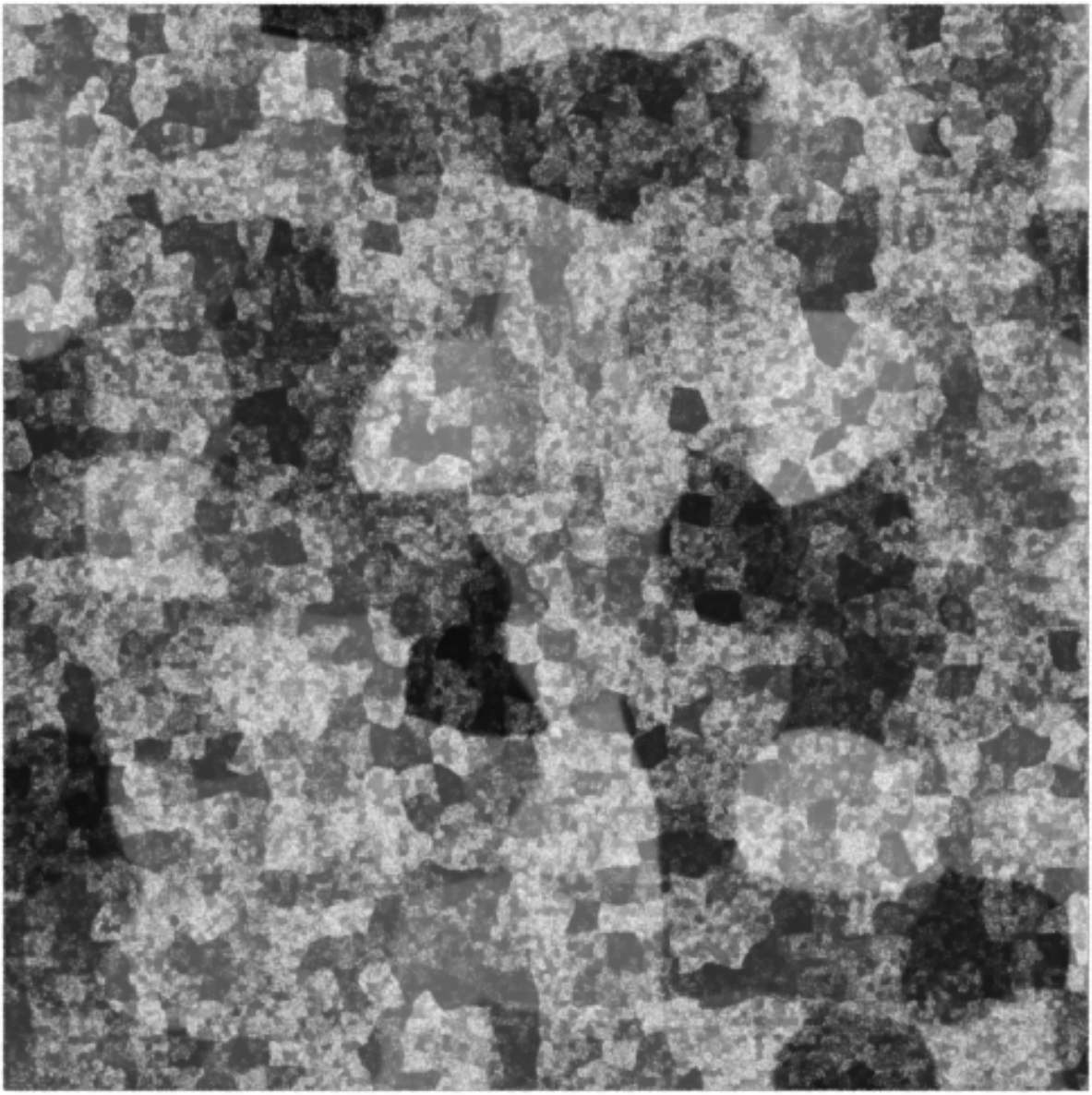
1.16.2 Screenshots

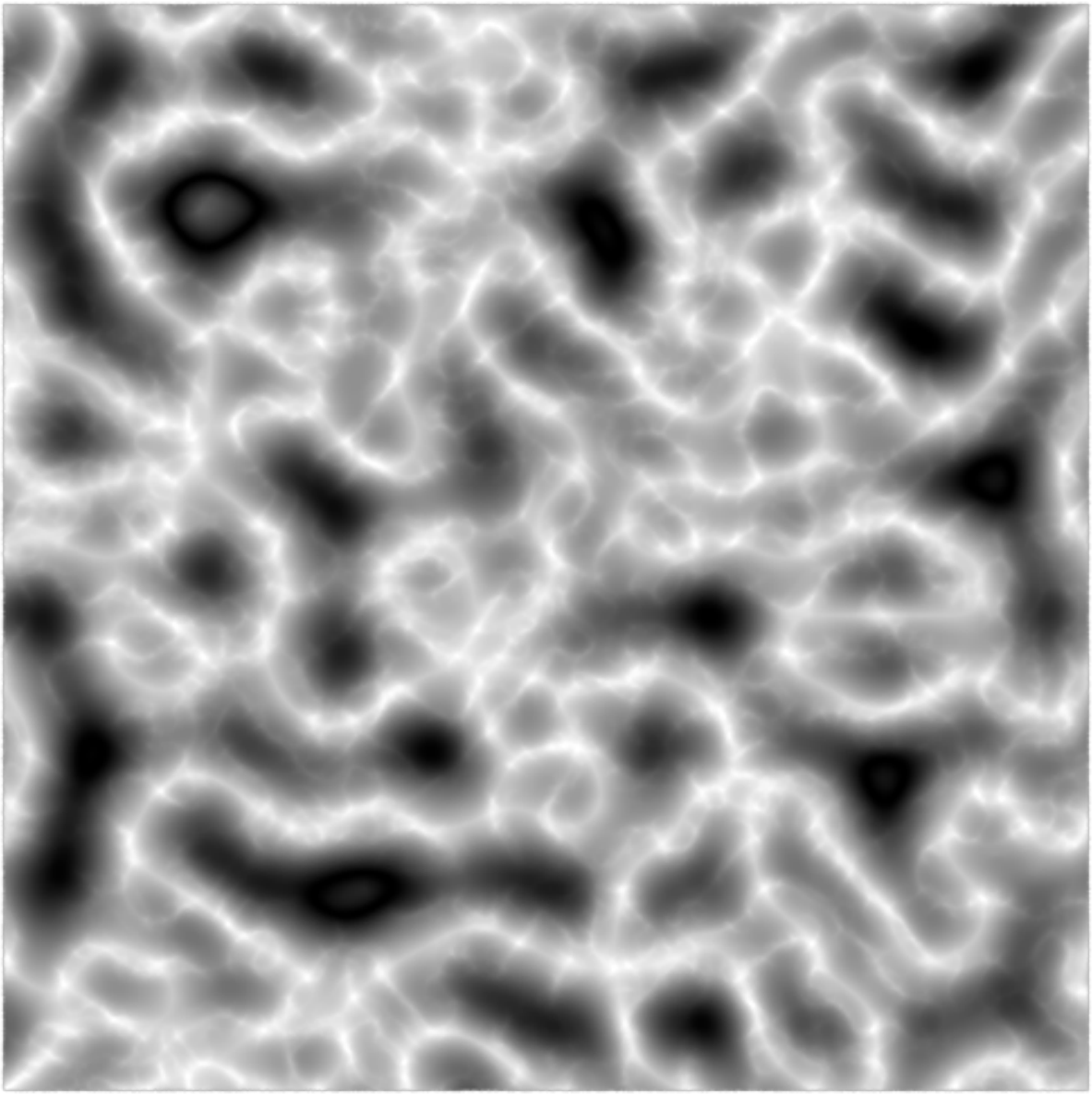
Some examples of what can be achieved, and is provided as presets.

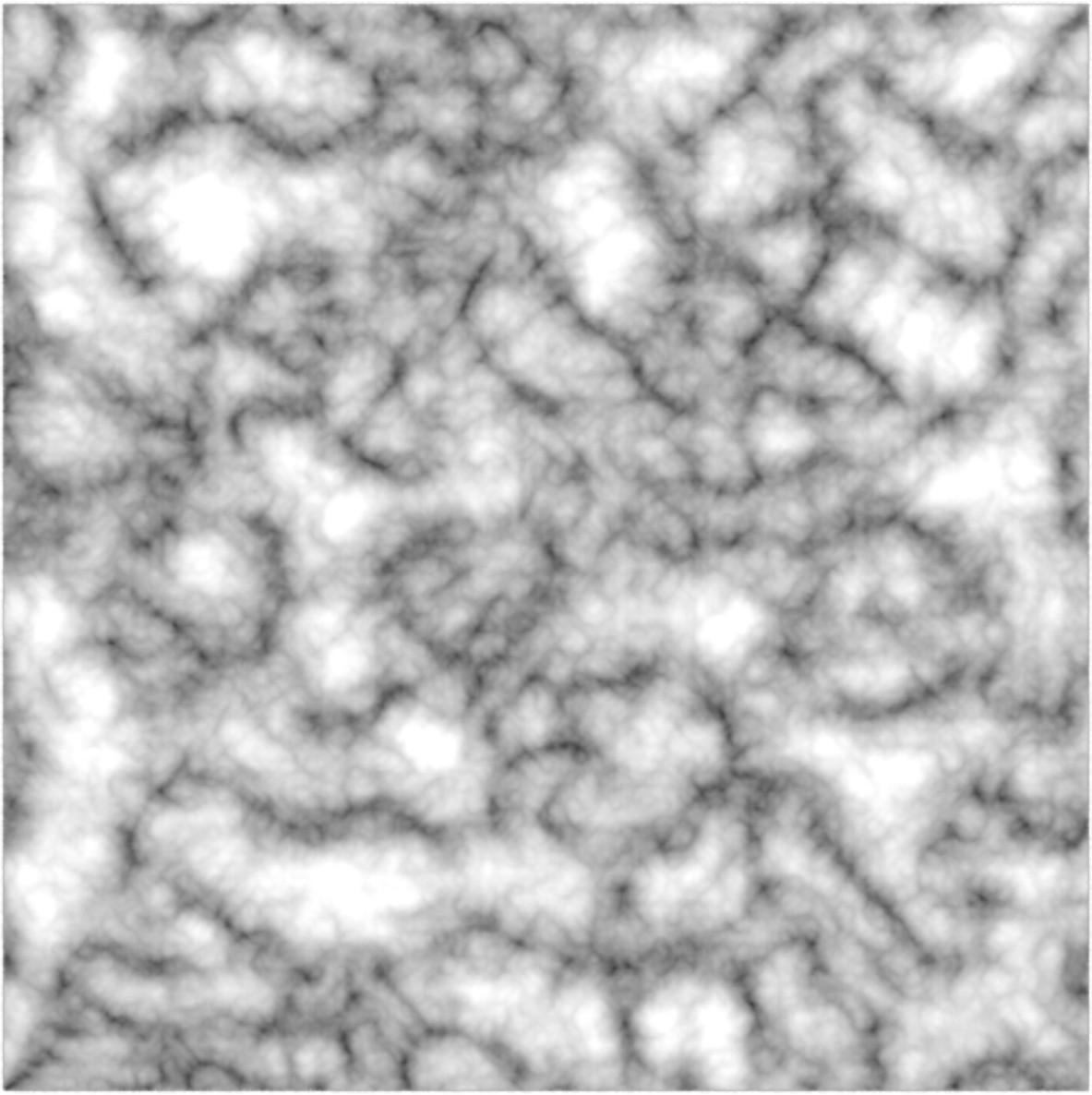


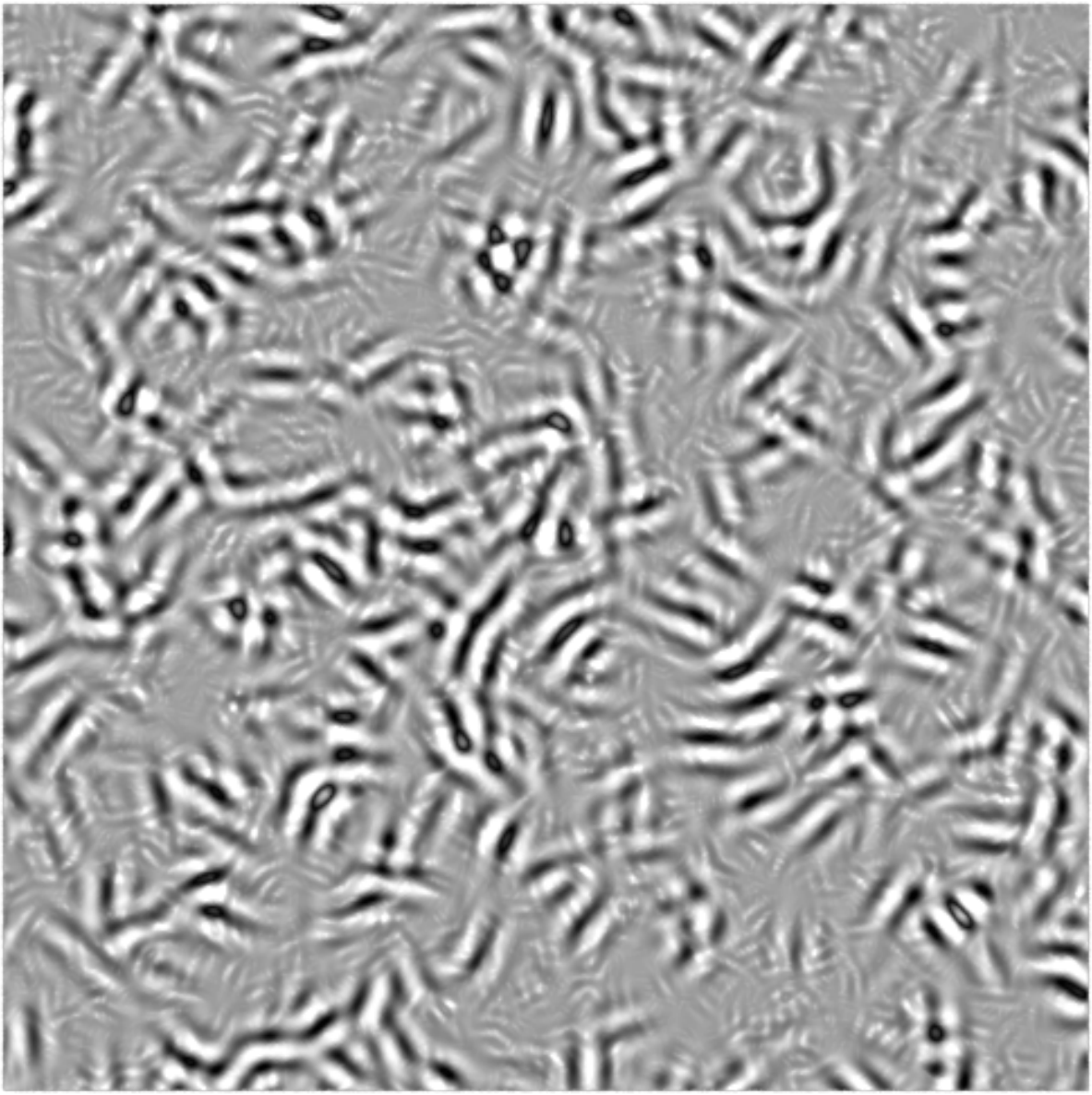


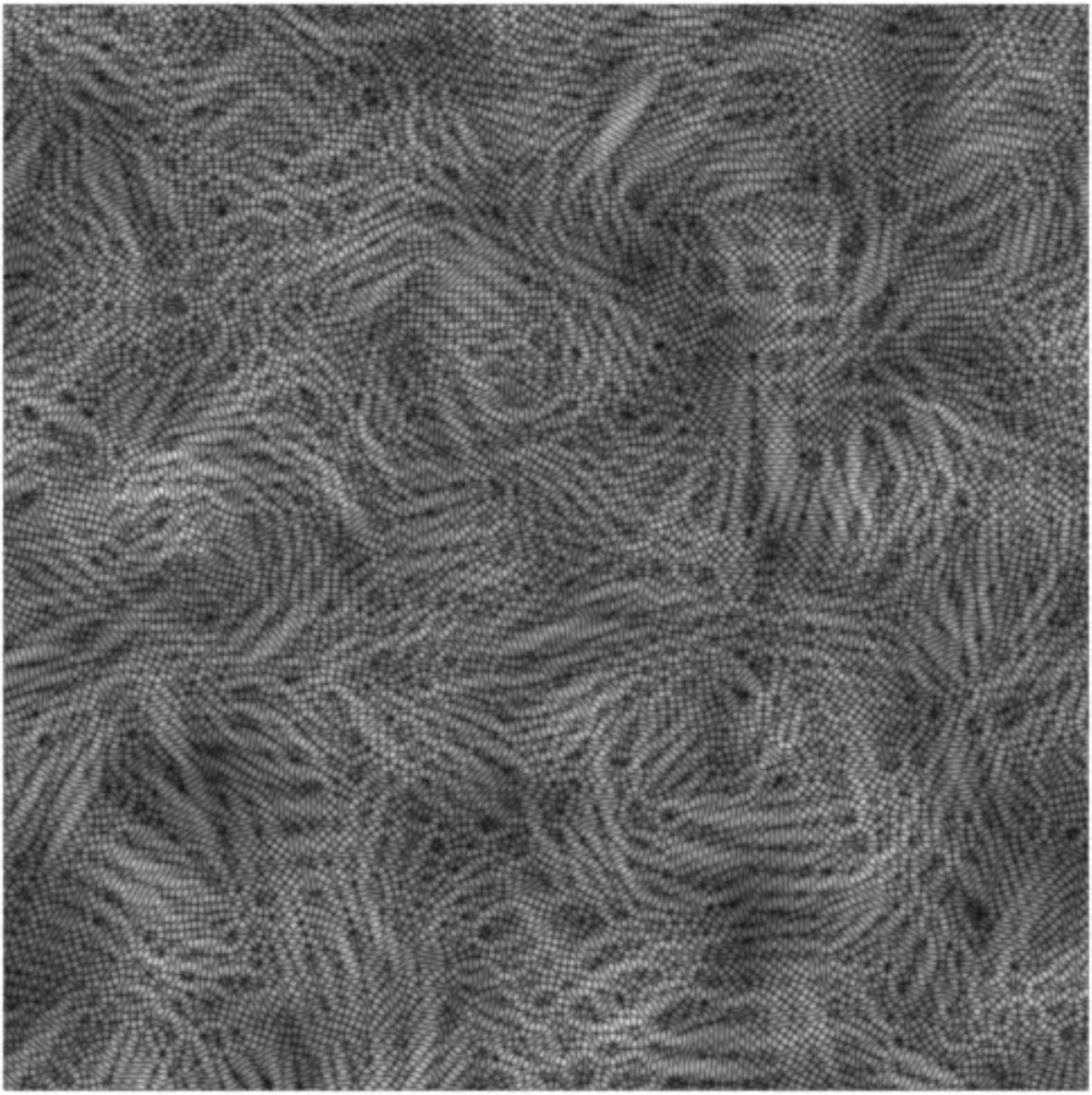












References

1.17 asNoise3D

A fractal solid noise node, with recursion, and an ample choice of noise primitives. The 3D counterpart of *asNoise2D* using the surface point P and a placement matrix, instead of UV coordinates.

1.17.1 Parameters

Color Parameters

Color 1 Primary color.

Color 2 Secondary color.

Contrast Contrast between primary and secondary colors.

Noise Parameters

Type The noise primitive used. It can take the following values

- Perlin *[Per85]*
- Simplex *[Per02]*
- Value
- Voronoise[1] *[Wor96]*
- Gabor *[GLLD12]*, *[LLDDutre09]*

Intensity The global noise intensity.

Frequency The noise frequency along the *x*, *y* and *z* directions, contained in a vector.

Ridges Toggling this checkbox will enable the *Ridged Noise* mode. A noise mode more suited to modelling the appearance of crests, mountains, when used to drive a displacement or bump map.

Inflection Enabling this checkbox will force the noise function to return a absolute value (if the noise was in [-1,1] range to begin with, otherwise it won't have any effect).

[1] Also known as generalized Voronoi. See Inigo Quilez article on voronoise.

Signed Noise Enabling this checkbox makes the noise function return values in the [-1,1] range, and disabling it will return values in [0,1] range.

Motion Parameters

Animated Noise Enabling this checkbox will animate the noise along time.

Frame Time Frame time, typically the frame number.

Time Scale Global time scale, affects the frame time.

Periodic Parameters

Periodic Enabling this checkbox will enable periodic noise, with a user-set *x*, *y* and *z* period.

Period The periodic noise *x*, *y* and *z* frequency, encoded in a vector.

Voronoise Parameters

Smoothness Controls the smoothness of the generalized Voronoi noise, with low values having a sharp cell boundary, and high values having a smooth Perlin noise like appearance.

Jittering Controls the jittering of the Voronoi cells, with low values producing a regular cell grid, and high values producing a randomized cell grid.

Gabor Parameters

Anisotropy This parameter controls the type of Gabor noise used. It can take the values

- Isotropic
- Anisotropic
- Hybrid

Direction Anisotropy vector to use when the Gabor noise *Anisotropy* mode is set to *Anisotropic*.

Bandwidth The bandwidth for the Gabor noise.

Impulses The number of impulses for the Gabor noise.

Filter Noise Enabling this checkbox will produce antialiased noise.

Recursion Parameters

Amplitude Initial noise amplitude before recursion.

Octaves The number of interations to perform.

Cascade The type of iteration to perform. It can be

- Additive
- Multiplicative

In the first case, the results of each iteration are accumulated, and in the second case, they are multiplied with the previous product.

Lacunarity Control for the gap between successive noise frequencies (sucessive octaves).

Offset Controls the multifractality.

Gain Controls the contrast of the fractal.

Distortion This parameter distorts the domain of the coordinates for each frequency.

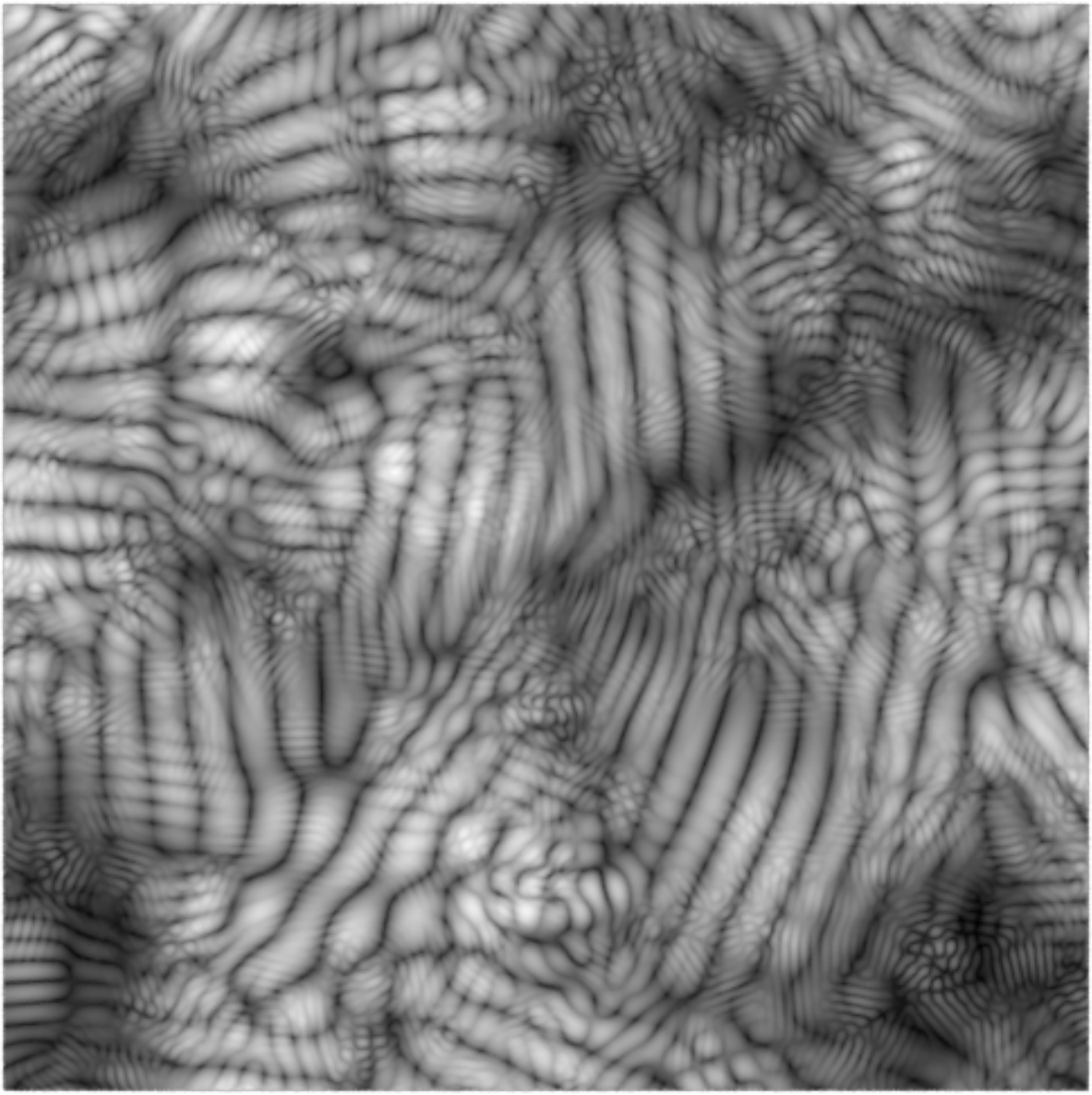
Outputs

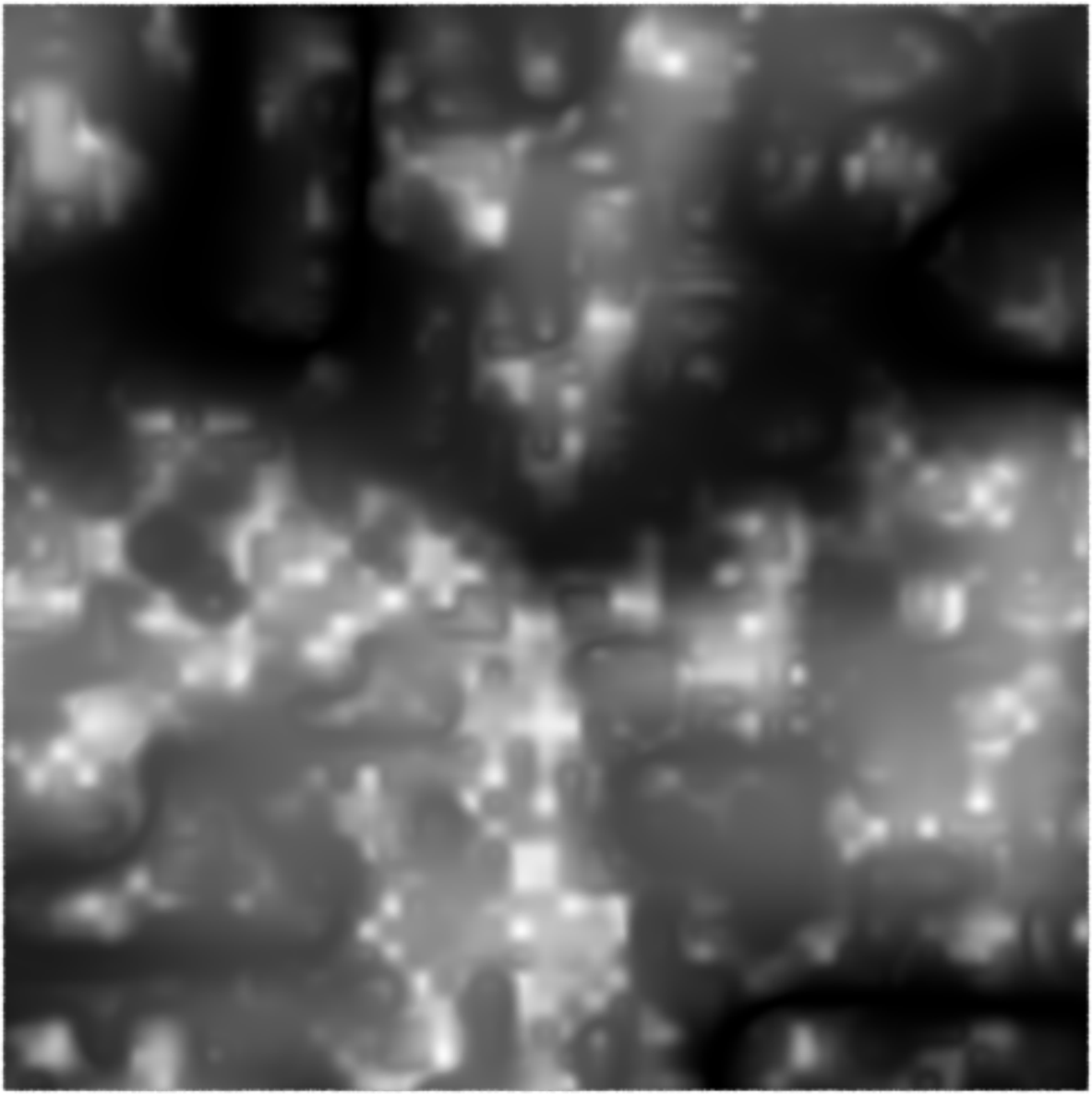
Output Color The color resulting from ghe *Features Mode* choice.

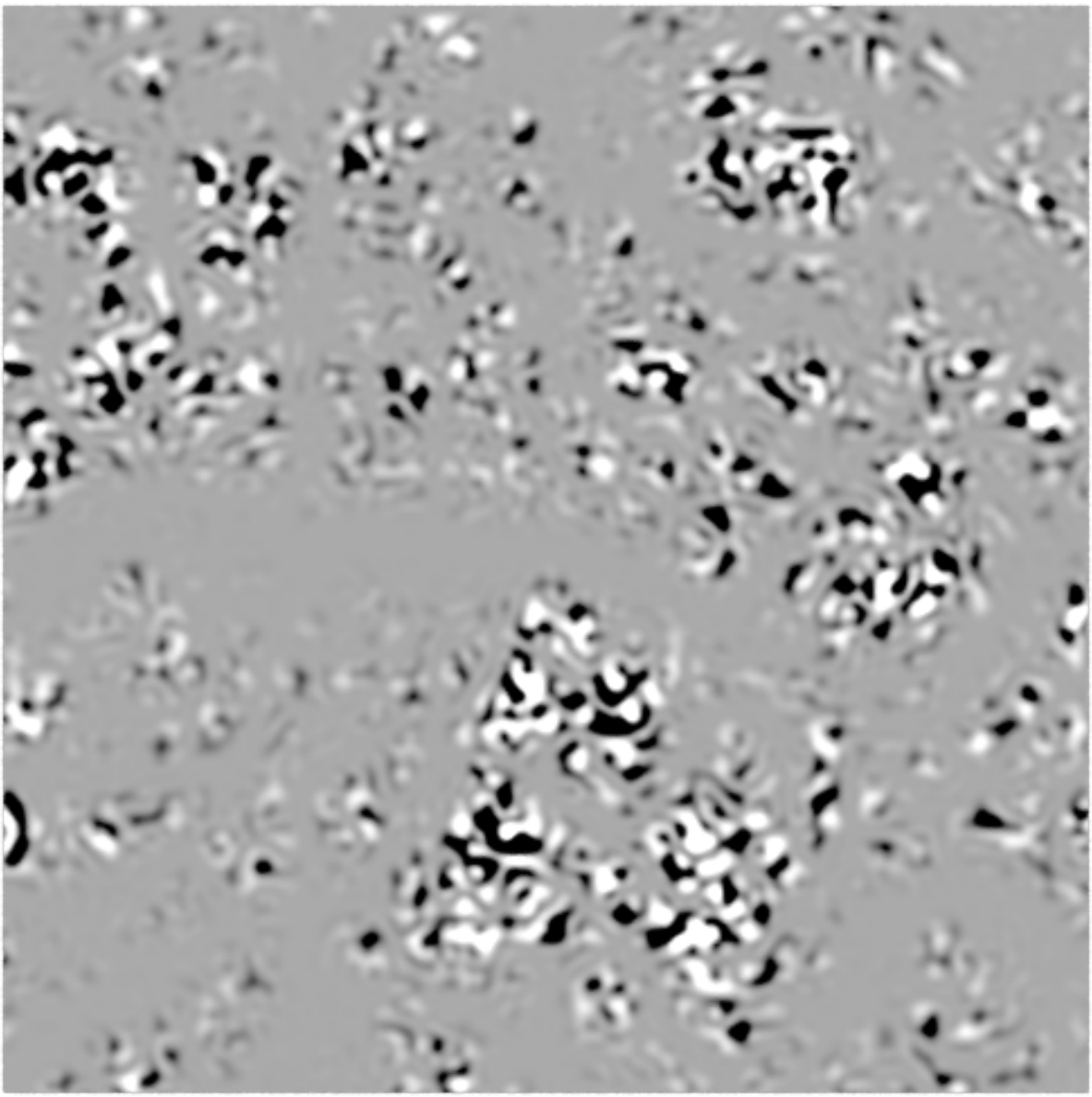
Output Alpha The alpha resulting from the *Features Mode* choice, usually luminance of the color only.

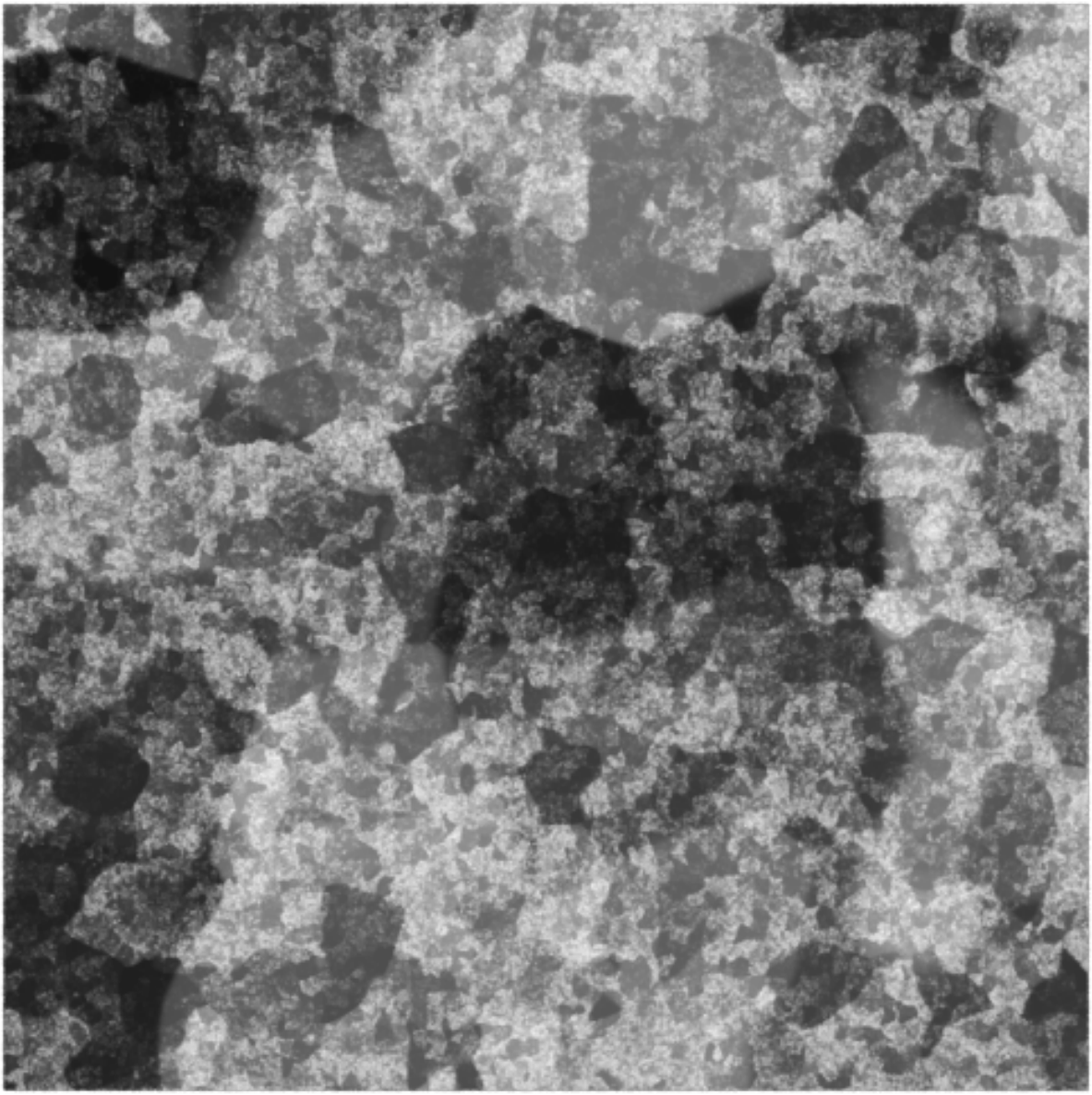
1.17.2 Screenshots

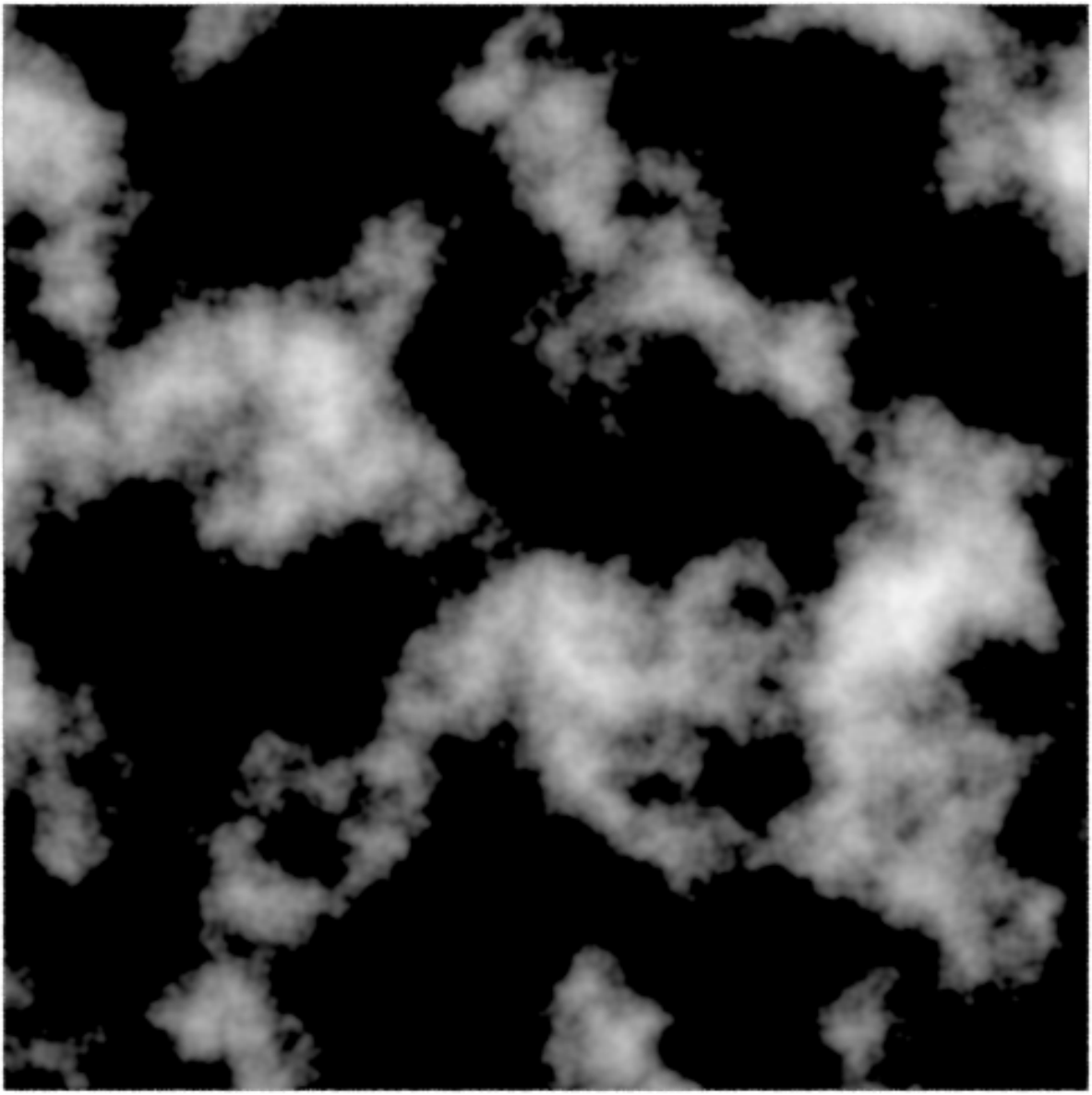
Some examples of what can be achieved with this node, besides what was illustrated by the *asNoise2D gallery*.

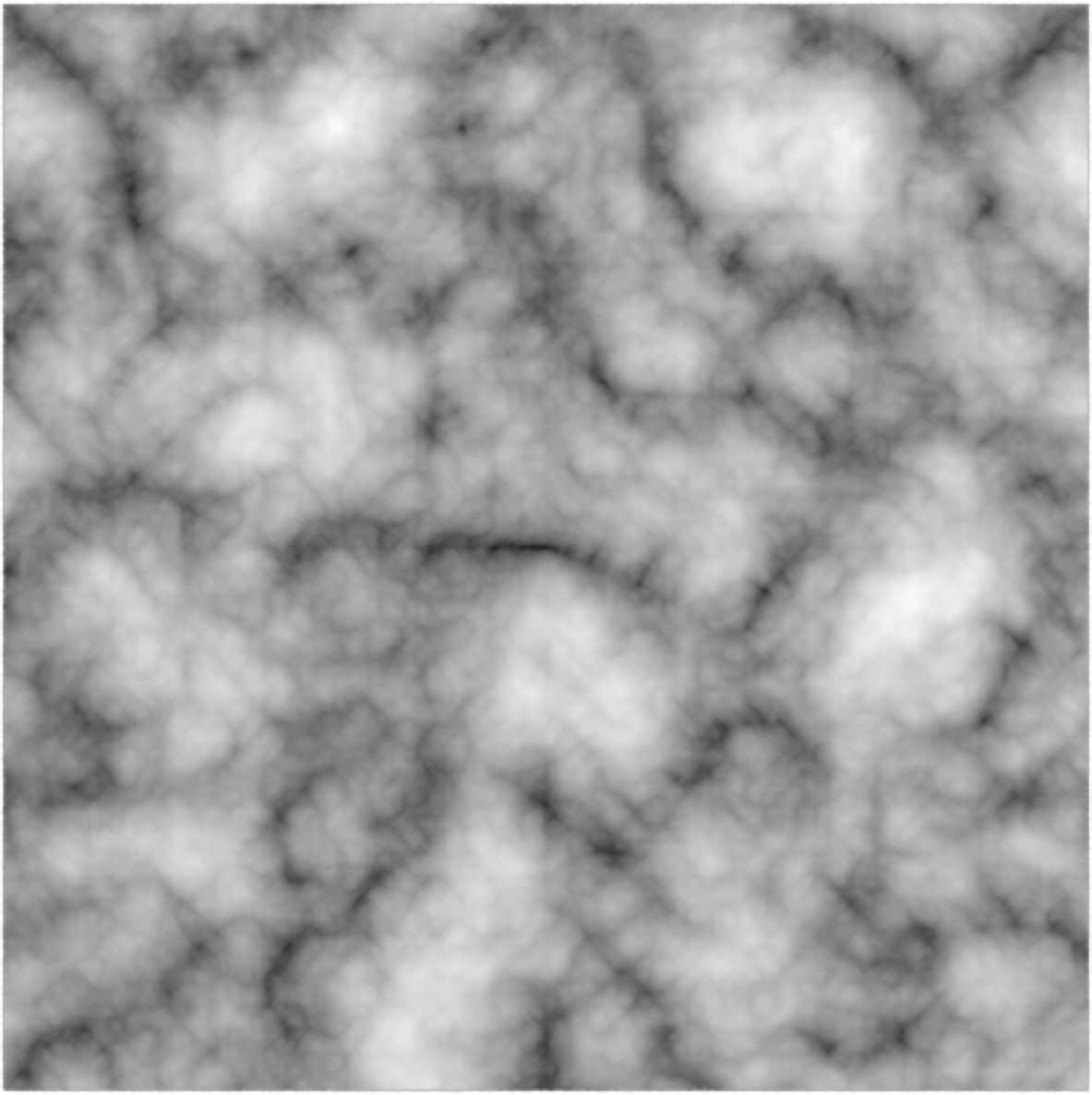


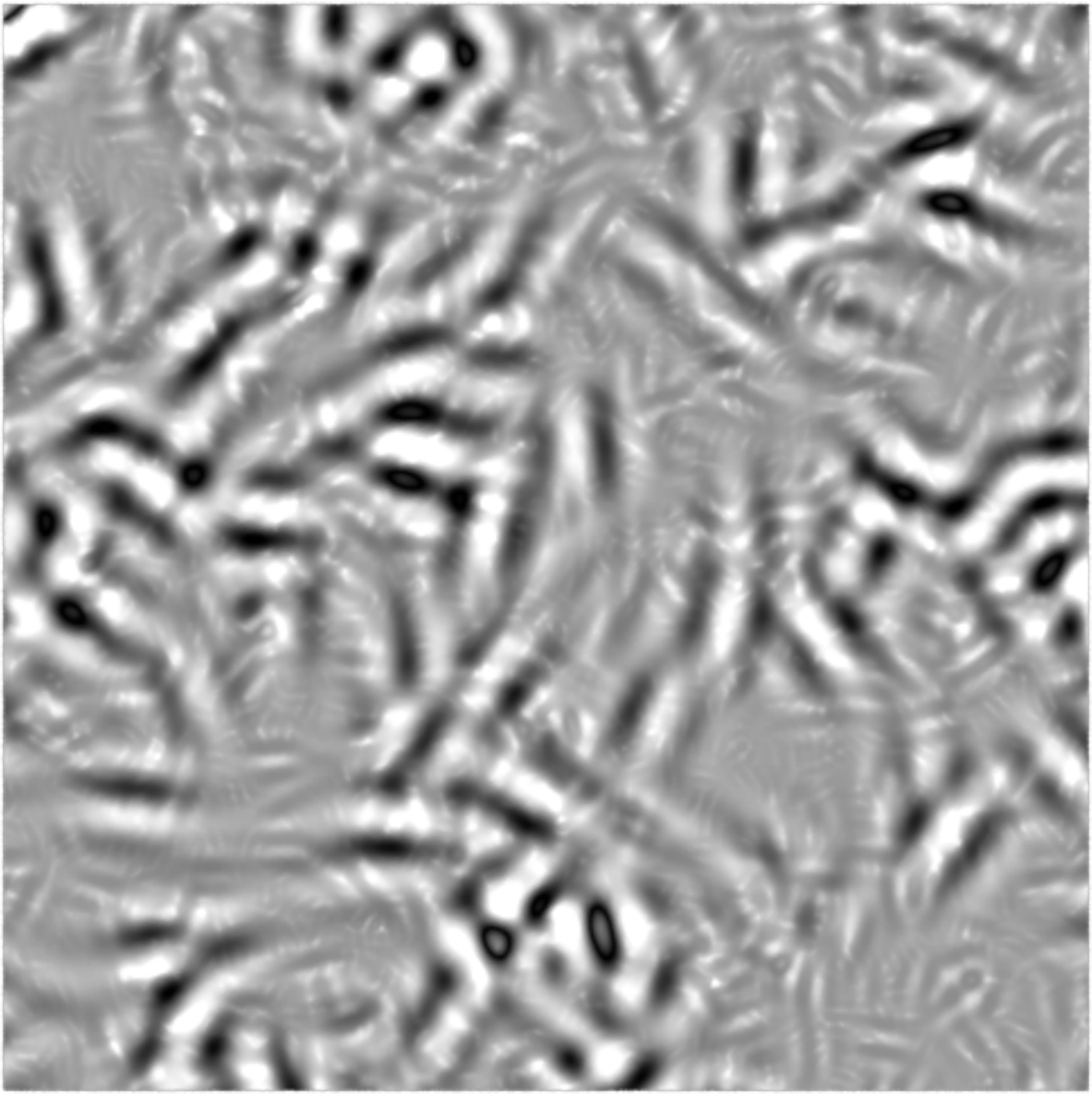


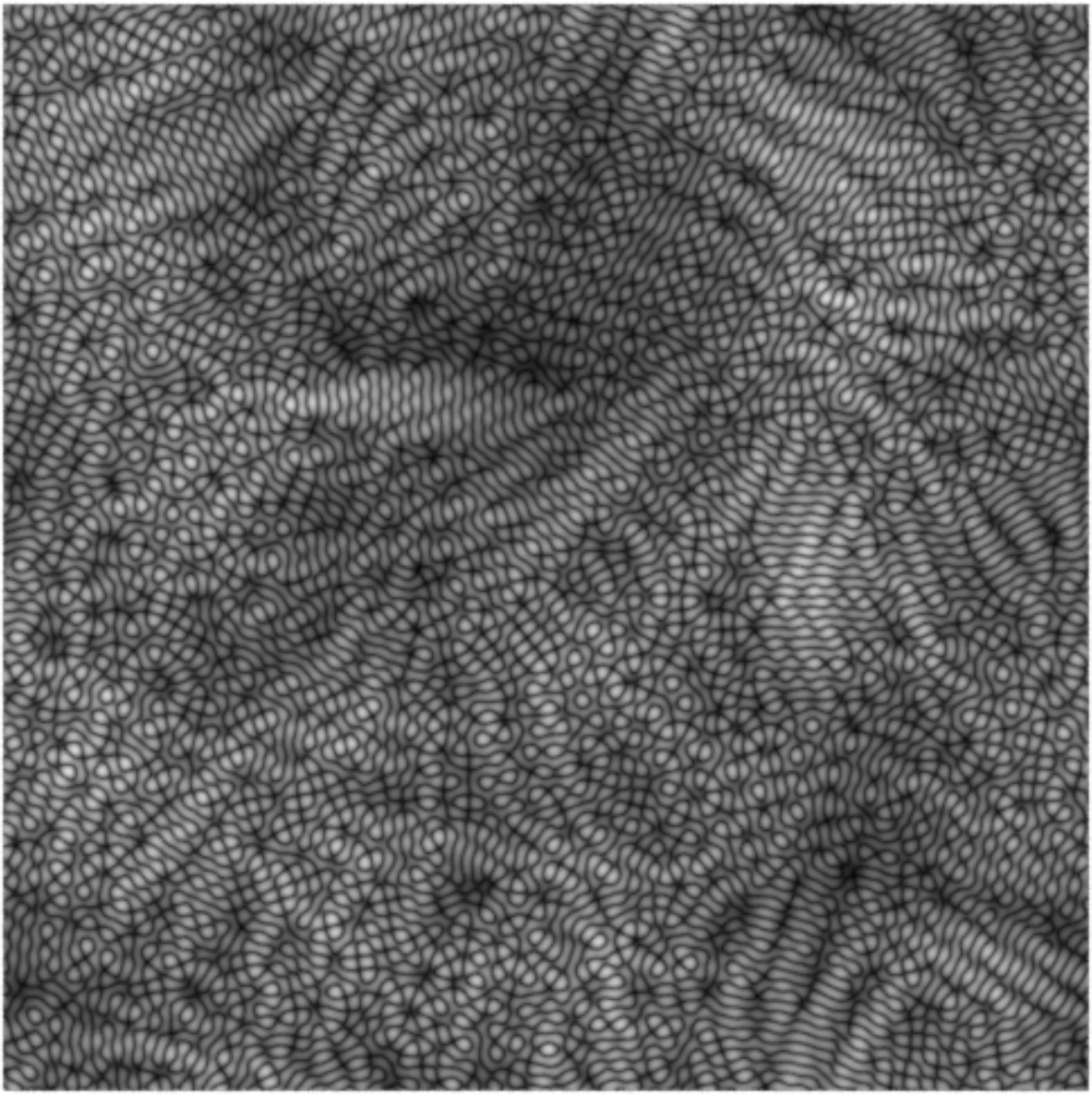




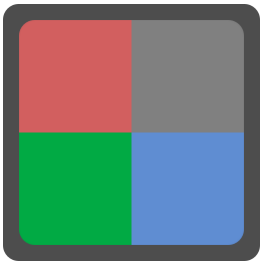








References



1.18 asTexture

A texture lookup node with full control over OSL's texture() call.

1.18.1 Parameters

Texture

Filename The texture filename

Atlas Type The texture atlas type, it can be one of

- None (ordinary texture, the *default*)
- ZBrush[1]
- Mudbox
- Mari[2]

Note: Though a full reference of UV tiles is outside the scope of this document, it suffices to say here that the ZBrush UV tiles pattern is in the form *u<N>_v<N>* with the tiles starting at 0, whilst Mudbox shares the same pattern but starts at 1. Mari uses the form *<UDIM>*. All a user needs to do is to point at the filename and choose the respective texture atlas type, and this node should take care of doing the appropriate string substitutions. See also the Maya documentation on UV tiles.

Color The default color to use if the texture lookup fails for any reason.

Alpha The default alpha value if there is none in the texture file, or if the lookup of the alpha channel fails for any reason.

Starting Channel The starting channel for the texture lookup. For an *RGBA* texture, the starting channel is 0, which is also the default.

S Blur Amount The amount of blur along the *s* texture coordinate, defaulting to 0.

[1] For the ZBrush and Mudbox case, the UV tiles are assumed to be separated by underscores.

[2] This note assumes however, that the UDIM pattern will always come last before the filename extension. That is, if you are using an animated sequence or frames of an animated sequence, then the padded frame numbers **must** come before the UDIM pattern. I.e, `<filename>.<padded frame numbers>.<UDIM>.<extension>.`

T Blur Amount The amount of blur along the *t* texture coordinate, defauling to 0.

S Filter Width A scaling factor for the size of the texture filter as defined by the differentials or implicitly by the differentials of the *s* texture coordinates, with a default value of 1.0. A value of 0.0 turns off texture filtering.

T Filter Width A scaling factor for the size of the texture filter as defined by the differentials or implicitly by the differentials of the *t* texture coordinates, with a default value of 1.0. A value of 0.0 turns off texture filtering.

S Wrap The texture wrapping mode along the *s* direction, which can be one of

- Default (the texture system default)
- Black
- Periodic
- Clamp
- Mirror

T Wrap The texture warpping mode along the *t* direction, which can be one of

- Default (the texture system default)
- Black
- Periodic
- Clamp
- Mirror

Interpolation Method The texture interpolation method, which can take the following values

- Smart Cubic (default)
- Cubic
- Linear
- Closest

See also:

This link on texture filtering for more details.

Color Management

Enable CMS Toggles the color management options *on* or *off*.

Input Transfer Function Applies an Electro-Optical Transfer Function, or EOTF, to the input texture, linearizing it. It can take the following values

- None/Raw
- sRGB
- Rec.709
- Gamma 2.2
- Gamma 2.4
- Gamma 2.6 (DCI)
- Rec.1886[3]

[3] See ITU-R BT.1886 recommendation for details on the electro-optical transfer function.

- Rec.2020

RGB Primaries It allows the user to set the RGB primaries that define the color space of the input texture, and can take the following values

- Raw[4]
- sRGB/Rec.709[5]
- AdobeRGB[6]
- Rec.2020 *[SCW14]*
- DCI-P3 *[Ini11]*
- ACES *[oMPAS12]*
- ACEScg *[DFD+15]*

Rendering RGB Primaries It allows the user to set the RGB primaries of the rendering or working space, and it should match the choice of rendering/working space of the renderer. It can take the following values

- sRGB/Rec.709
- Rec.2020
- DCI-P3
- ACES
- ACEScg

Texture Coordinates

UV Coords The *uv* texture coordinates.

UV Filter Size The computed filter size for the *uv* texture coordinates.

1.18.2 Outputs

Output Color The color resulting from the *Features Mode* choice.

Output Alpha The alpha resulting from the *Features Mode* choice, usually luminance of the color only.

Output Single Channel The output when the texture lookup is made on a greyscale image.

---

[4] Because it makes no sense whatsoever to use colorimetry on non-color information or data, such as normal maps, or Z depth, motion vectors, and so on.

[5] sRGB shares the same CIE xy chromaticity coordinates with ITU-R BT.709/Rec.709 , hence this node refers to the RGB primaries shared by these two color spaces as *sRGB/Rec.709*.

[6] See encoding characteristics of AdobeRGB specification.

References

1.19 asTexture3D

A node that performs 3D lookup of a volume texture, typically a Field3D file.

1.19.1 Parameters

3D Texture

Filename The volume texture, typically a **.f3d*[1] file.

Starting Channel The starting channel for the volume texture lookup.

Color The default color to use if the texture lookup fails.

Channel Fill The default scalar value to use for any channels requested but not present in the volume texture.

Time The time value to use if the texture specifies a time varying local transformation.

Wrapping

S Wrap The wrapping mode along the *s* coordinate, it can be one of

- Default
- Black
- Periodic
- Clamp
- Mirror

T Wrap The wrapping mode along the *t* coordinate, it can be one of

[1] These files are exported as a result of simulations done in other applications. In the case of Maya, one could for instance convert Maya's fluids to a Field3d file via the Field3D Maya Plugin. Other simulation applications also allow exporter volume data as **.f3d*.

- Default
- Black
- Periodic
- Clamp
- Mirror

R Wrap The wrapping mode along the *r* coordinate, it can be one of

- Default
- Black
- Periodic
- Clamp
- Mirror

Blur

Blur Width The amount of blur along the width of the volume, or the *s* direction.

Blur Height The amount of blur along the height of the volume, or *t* direction.

Blur Depth The amount of blur along the depth of the volume, or *r* direction.

Filter

Width Filter Scale for the size of the filter defined by the partial derivatives along the *s* direction or implicitly from *P*, with a value of 0 disabling filtering altogether.

Height Filter Scale for the size of the filter defined by the partial derivatives along the *t* direction or implicitly from *P*, with a value of 0 disabling filtering altogether.

Depth Filter Scale for the size of the filter defined by the partial derivatives along the *r* direction or implicitly from *P*, with a value of 0 disabling filtering altogether.

Coordinates

Surface Point The surface point being shaded.

Coordinate System The coordinate system to use for the volume, it can be

- Object Space
- World Space
- Camera Space

1.19.2 Outputs

Output Color The color resulting from the *Features Mode* choice.

1.20 asVoronoi2D

A procedural 2D Worley *[Wor96]* like noise shader, that outputs not only the resulting color, but the four nearest features to the evaluated point, their respective positions, and their cell color IDs. See also *[EMP+02]*.

1.20.1 Parameters

Color Parameters

Color 1 Primary cell color.

Color 2 Secondary cell color.

Contrast Contrast between primary and secondary cell color.

Recursion Parameters

Amplitude Controls the amplitute at each octave, including the starting iteration.

Octaves Number of iterations to perform, higher values lead to increasing detail, but increased computational cost as well.

Lacunarity Defines how large the gaps are in the cell noise with increasing octaves, higher values lead to higher gaps, lower values to small gaps.

Persistence The persistence of the fractal is a gain factor to apply to the amplitude at each iteration, but it only has an effect when the shader is set to the mode *pebbles*.

Cell Parameters

Density The density of the cells, with higher values resulting in a higher number of cells in the same area.

Jittering How random the placement of the cells is, with low values resulting in a ordered grid of cells, and higher values resulting in aleatory placement of cells.

Metric Which metric to choose to calculate the distance from cell to feature points. There are several to choose from, resulting in different types of patterns.

- Euclidian distance[1]
- Sum of square difference
- Tchebychev distance[2]
- Sum of absolute difference
- Akritean distance
- Minkowski metric[3]
- Karlsruhe metric[4]

The sum of the square difference is also known as the Manhattan metric[5].

The Minkowski metric is a generalized metric whose P parameter allows you to go from the Euclidian distance when P has a value of 2, to the Manhattan distance when P has a value of 1, and as P reaches infinity, it represents the Tchebychev metric.

The Akritean distance if a weighted mix of the Euclidian distance, and the Tchebychev distance.

The Karlsruhe metric, also known as Moscow metric, is a radial metric, returns radial sections from a cell at the center.

Minkowski Parameter Controls the metric, with a value of 1 being the Manhattan distance, 2 being the Euclidian distance, and higher values tending to the Tchebychev metric as the parameter approaches infinity.

Coverage The Akritean distance coverage, or the weighting mix between the Euclidian distance and the Tchebychev distance.

Features Mode The features mode to use when computing the output color.

- Feature 1, or nearest feature from the cell
- Feature 2, or second nearest feature from the cell
- Feature 3, or third nearest feature from the cell
- Feature 4, or fourth nearest feature from the cell
- F1 + F2, or sum of first and second nearest features
- F2 - F1, or difference between second and first nearest features
- F1 * F2, or product of first and second nearest features
- F1 / F2, or division of first nearest feature by second nearest feature

[1] The Euclidian distance or Euclidian metric, also known as L_2 norm, is the straight line distance between two points in Euclidian space.

[2] The Chebyshev (or Tchebychev) distance, also known as Chessboard distance or L_∞ norm, is a metric on a vector space where the distance between two vectors is the greatest of their differences along any coordinate dimension.

[3] The Minkowski distance, also known as L_P norm, is a metric which is a generalization of both the Euclidian distance and the Manhattan distance, being equal to the Manhattan distance when its *p* parameter is equal to 1, or equal to the Euclidian distance when its *p* parameter is equal to 2. On the limit as *p* approaches infinity, it is equal to the Chebyshev distance.

[4] In metric geometry, the Karlsruhe metric is a measure of distance that assumes travel is only possible along rays through the origin and circular arcs centered at the origin.

[5] The Manhattan distance, also known as *taxicab* metric or L_1 norm, is a metric in which the distance between two points is the sum of the absolute differences of their Cartesian coordinates.

- F1 ^ F2, nearest feature raised to the second nearest feature
- Pebbles, a mode that resembles pebbles
- Cell ID 1, the ID of the nearest feature to the cell
- Cell ID 2, the ID of the second nearest feature to the cell
- Cell ID 3, the ID of the third nearest feature to the cell
- Cell ID 4, the ID of the fourth nearest feature to the cell

Note: The unmodified features, points and their color IDs are also output from the shader, giving the user greater creative potential. The feature modes above are but a starting point.

Color Balance

The standard Maya color balance, gain, offset parameters. Please consult Maya's documentation for more information on these controls.

Effects

The standard Maya effects parameters. Please consult Maya's documentation for more information on these controls.

Coordinates

The input UV coordinates, typically from an upstream *placement2d* node.

1.20.2 Outputs

Output Color The color resulting from ghe *Features Mode* choice.

Output Alpha The alpha resulting from the *Features Mode* choice, usually luminance of the color only.

Output Features An array of 4 floats, containing the four nearest features to the cell.

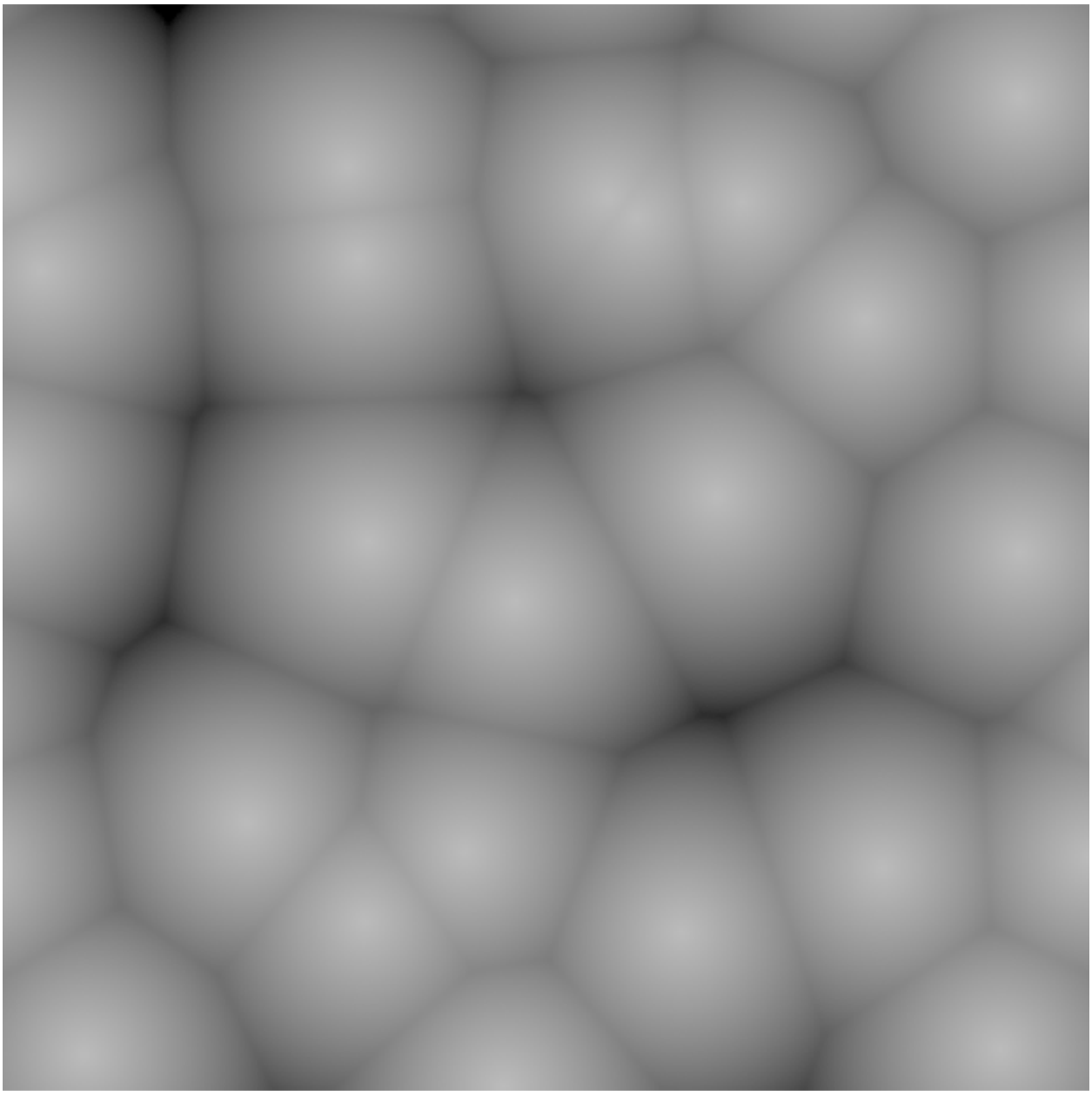
Output Positions An array of 4 points, containing the center of the four nearest features to the cell.

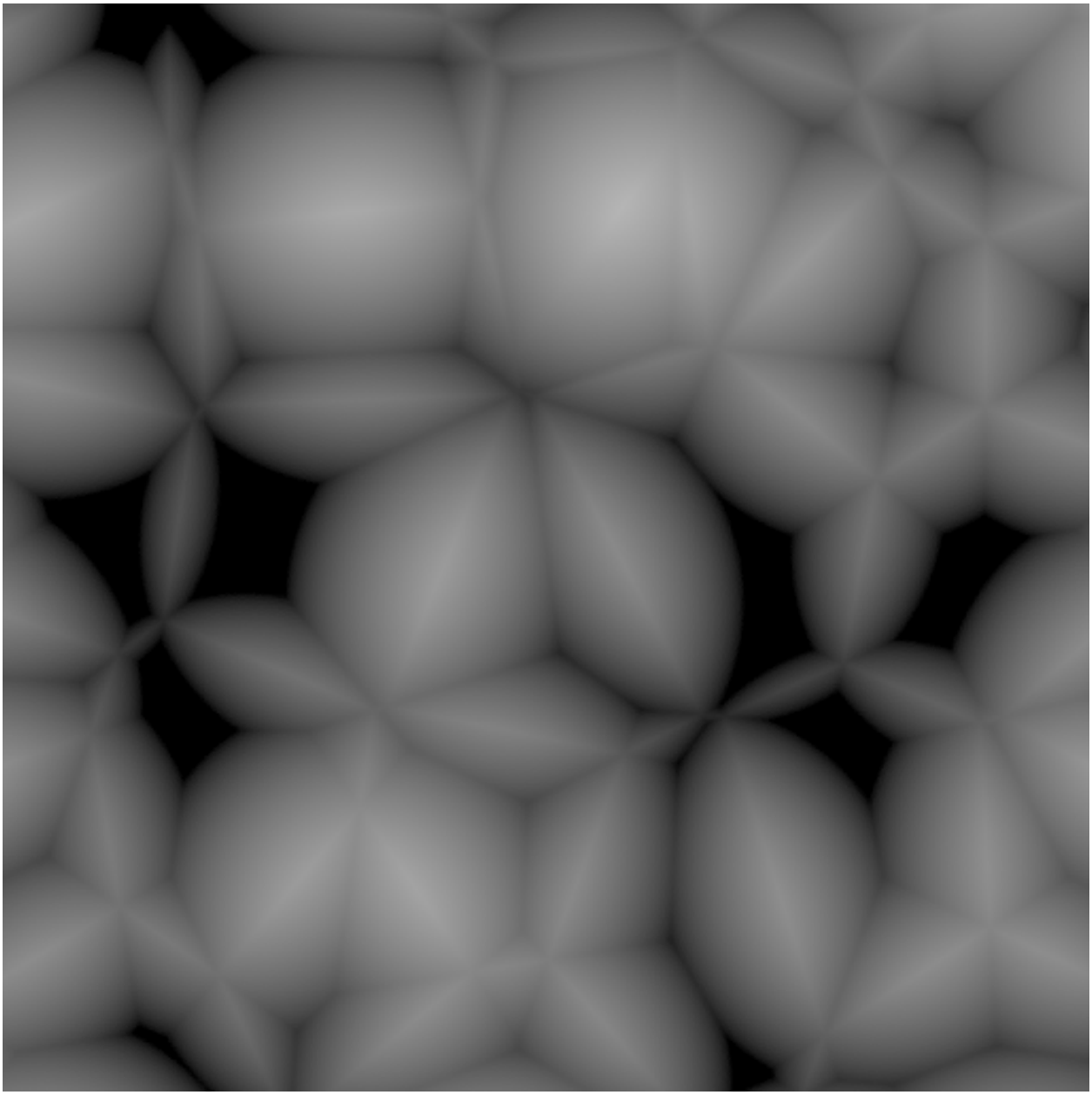
Output IDs An array of 4 colors, containing the color IDs of the four nearest features to the cell.

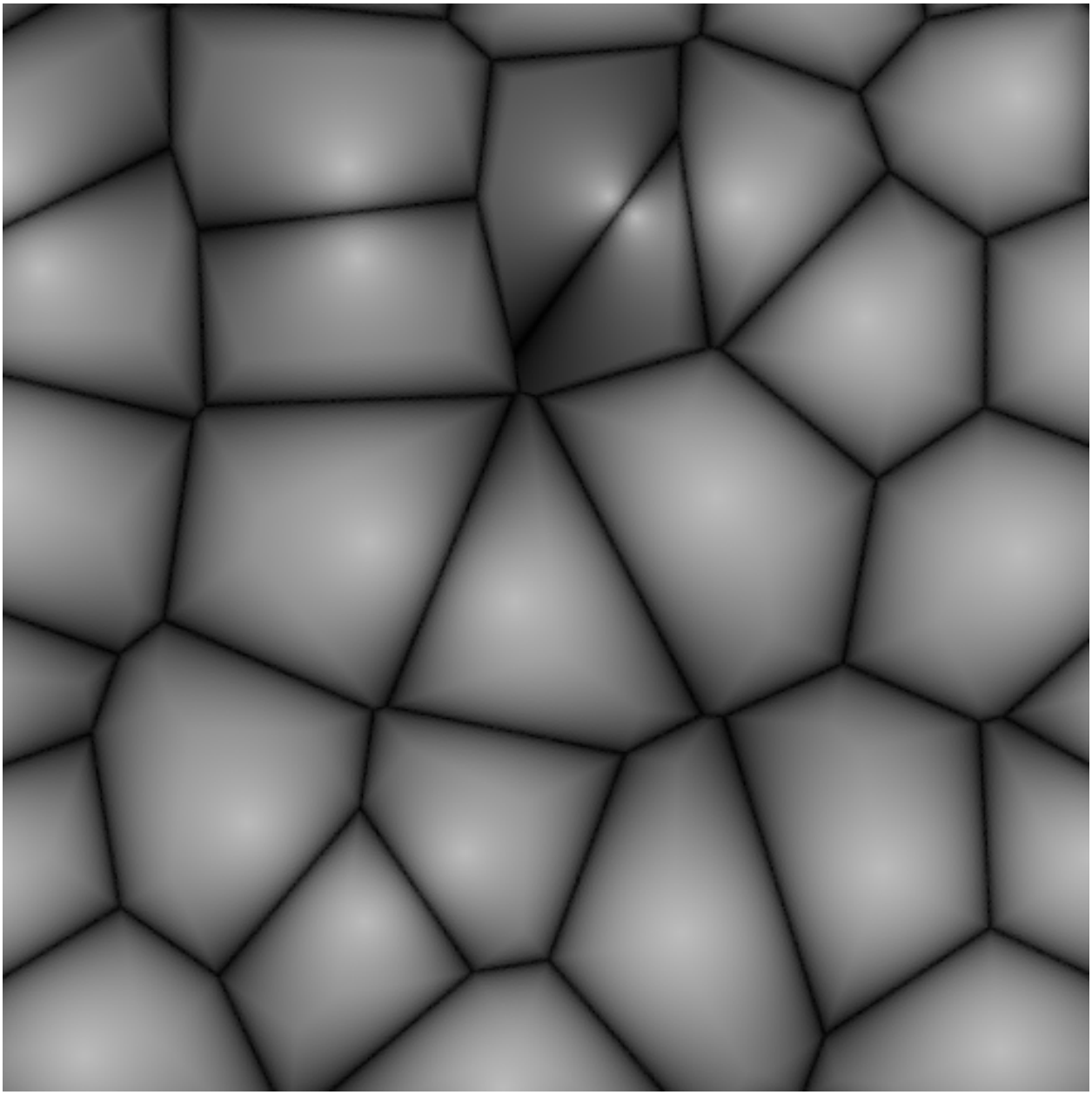
Warning: presently OSL does not allow connections from/to array elements, and appleseed-maya is not enabling the array outputs for now. This will be addressed in a future release.

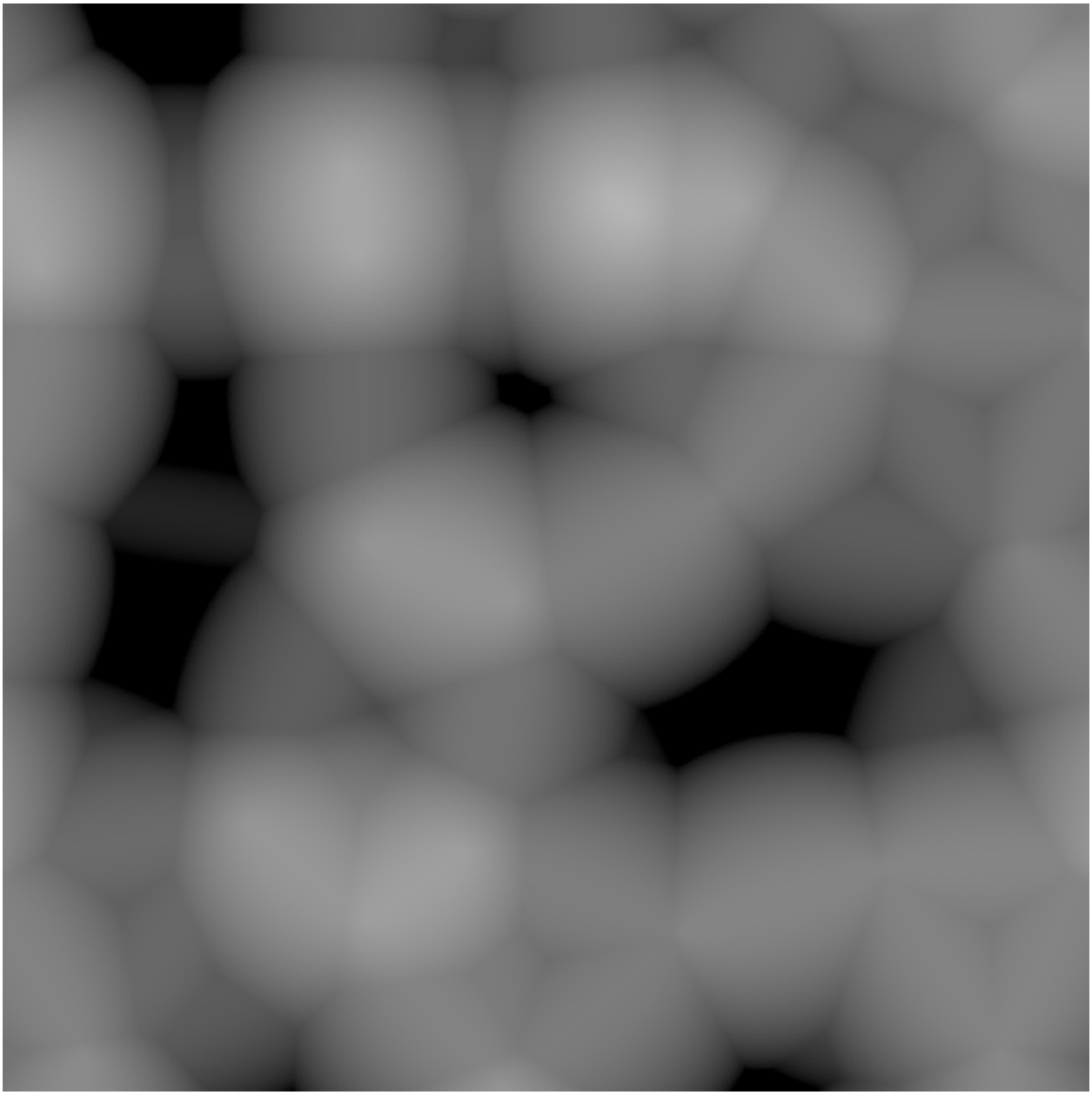
1.20.3 Screenshots

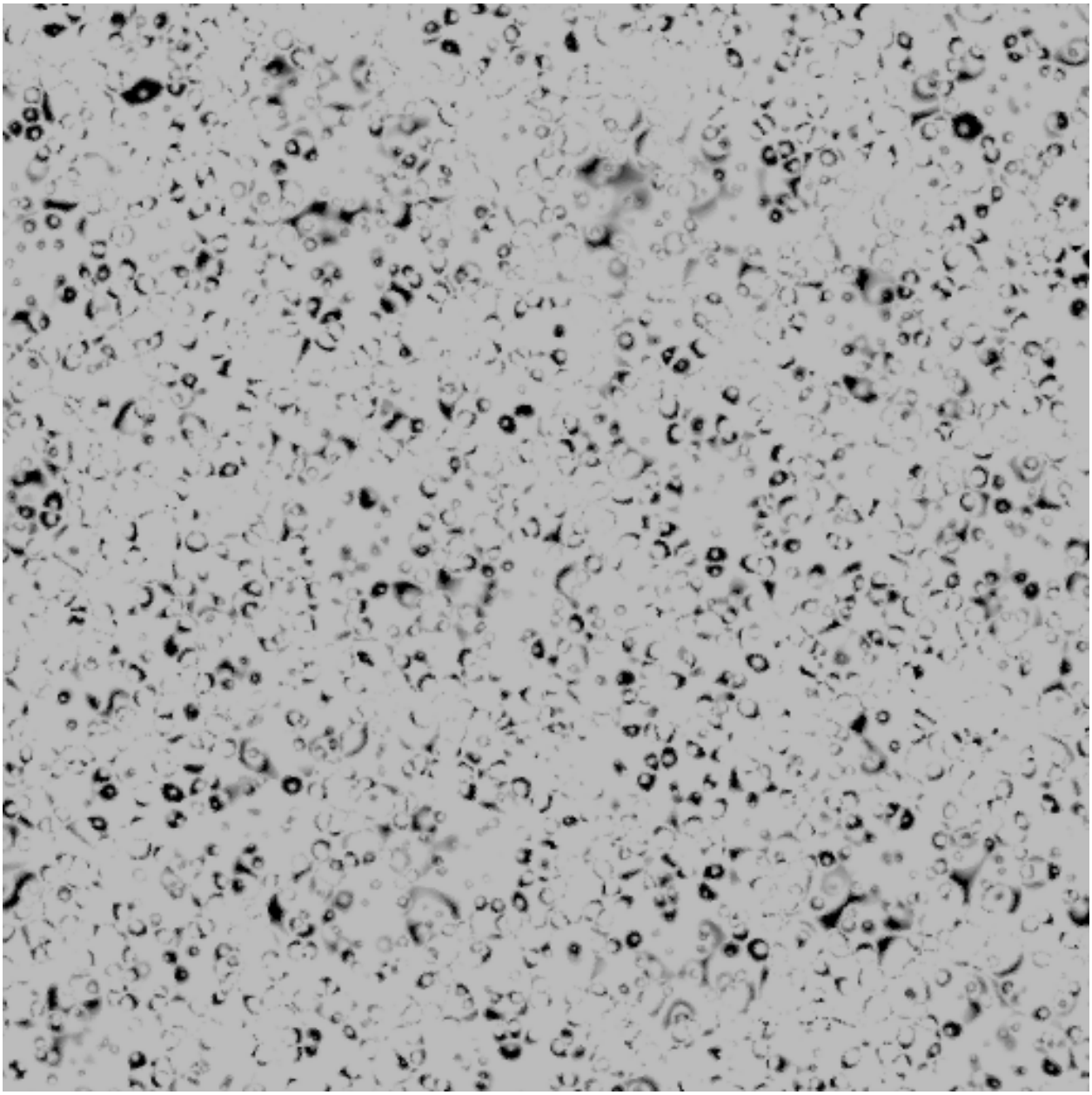
Some examples of feature output modes and metrics.

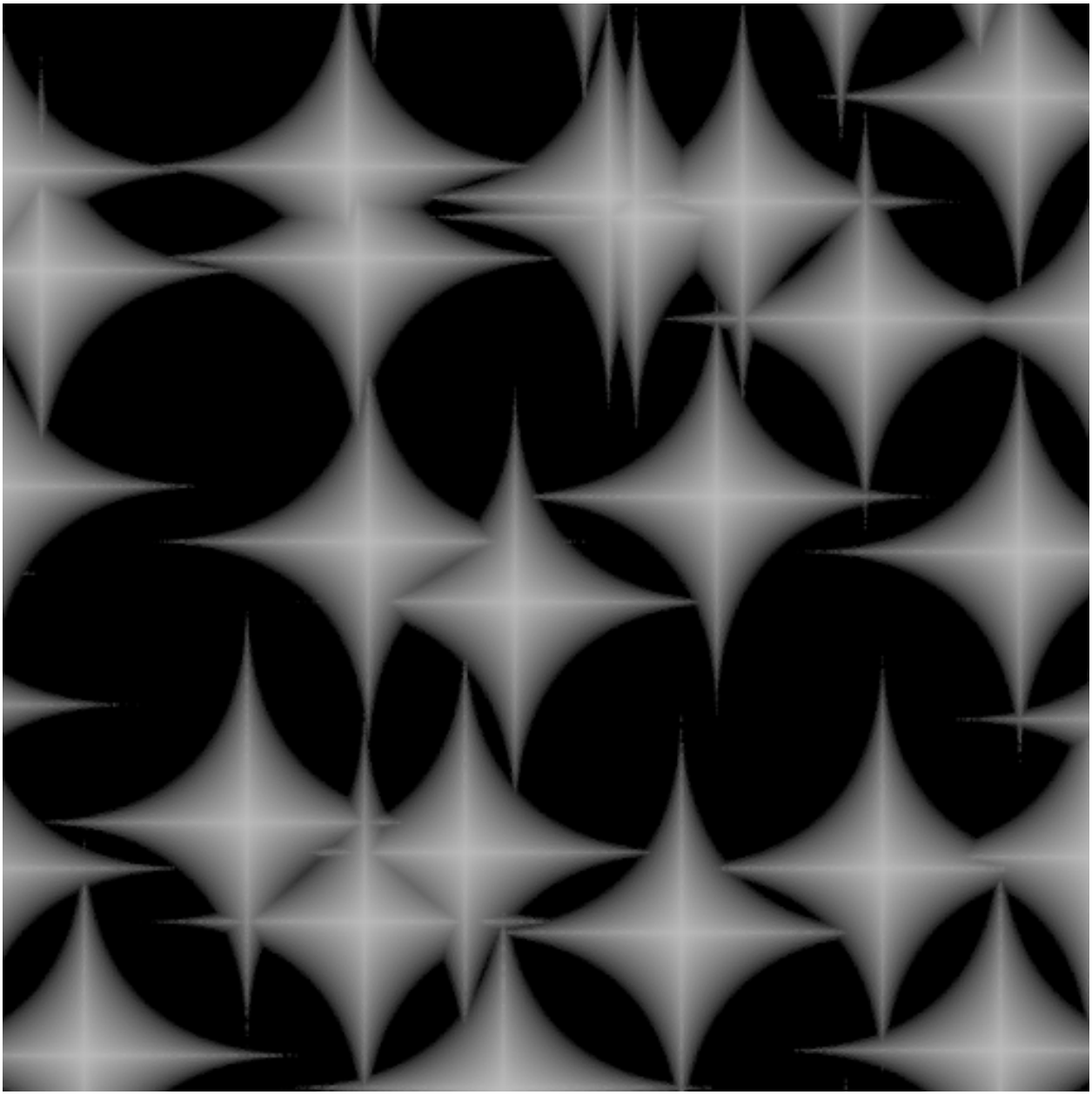


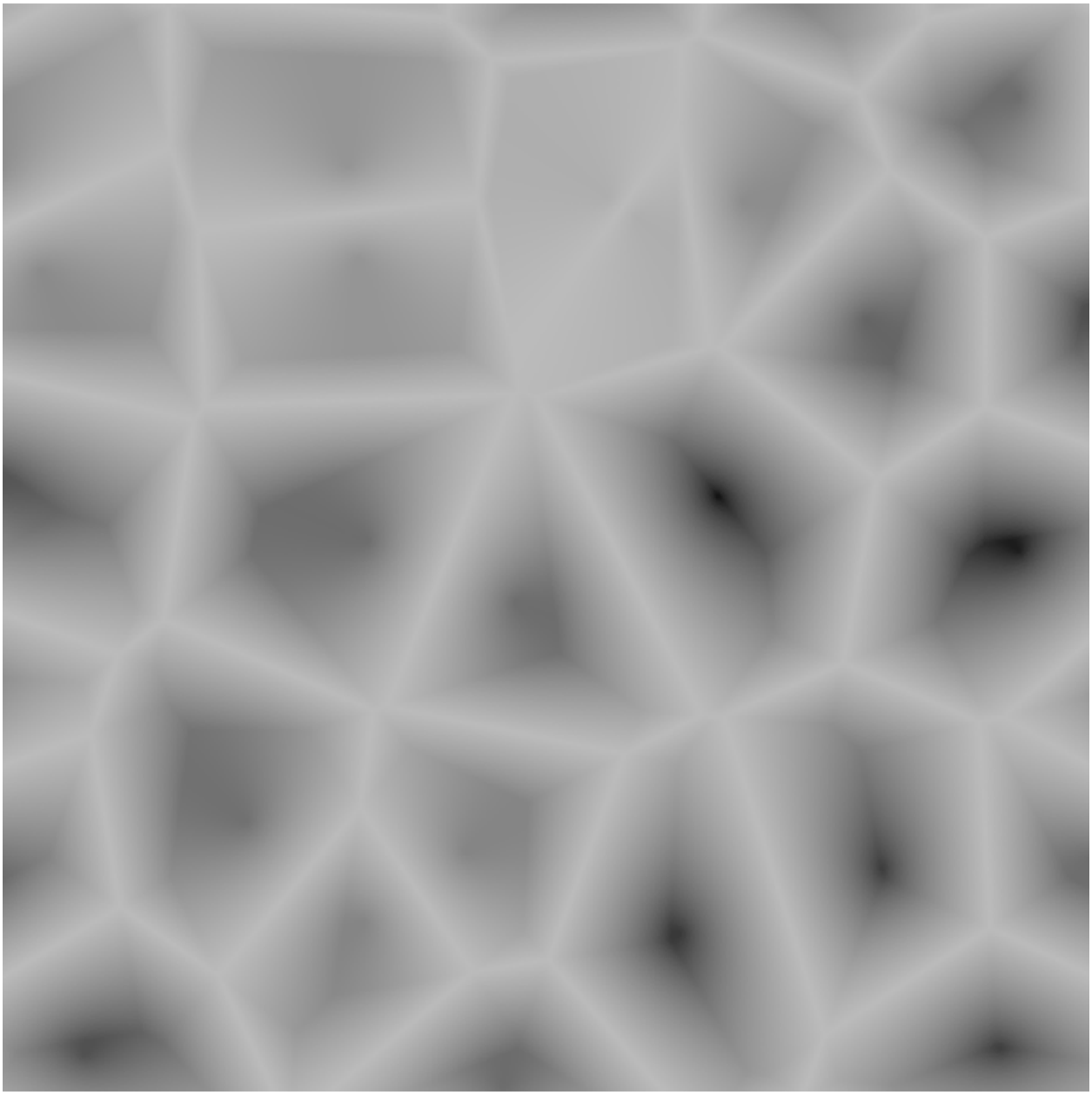


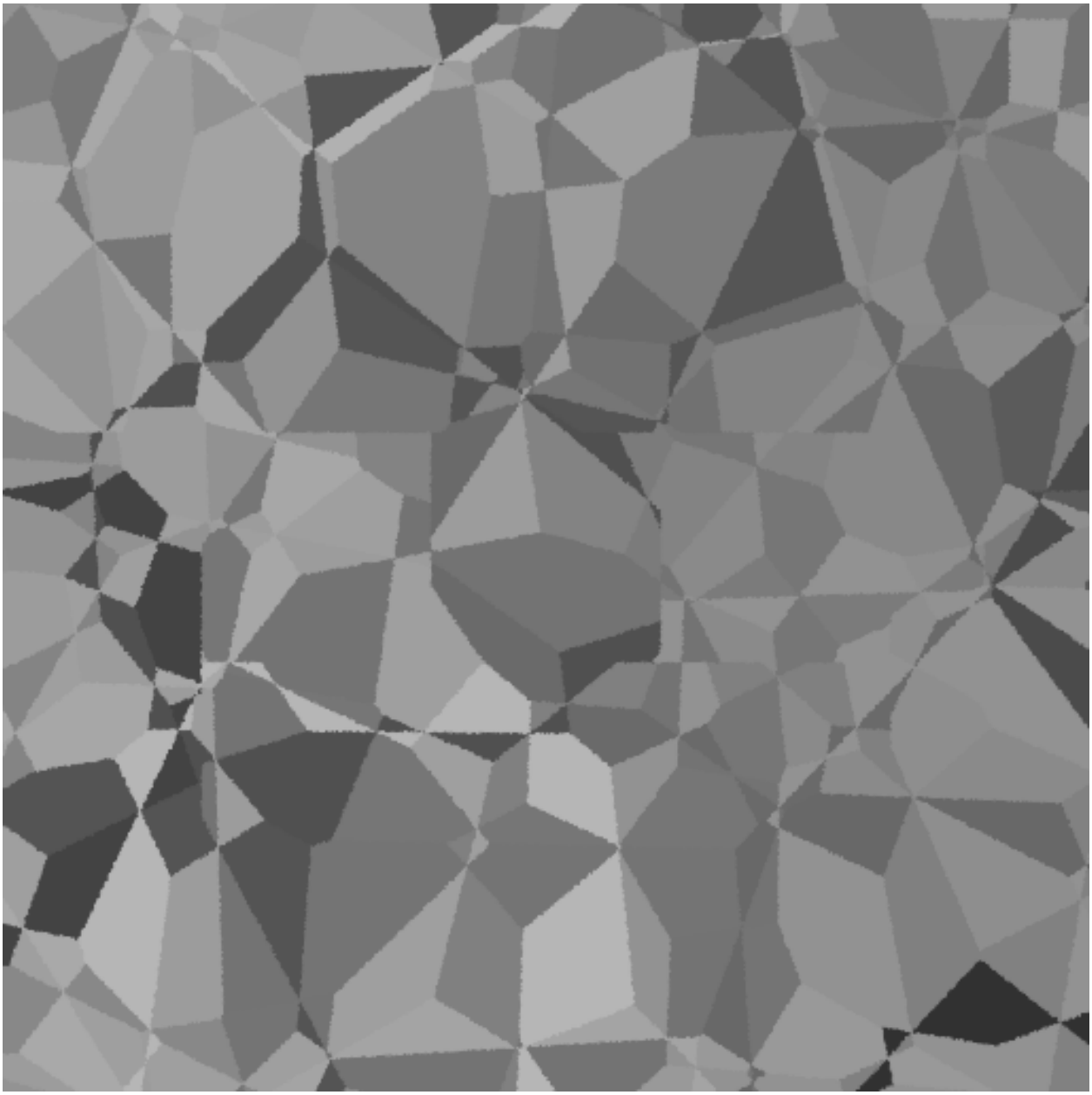












References

1.21 asVoronoi3D

A procedural 3D Worley *[Wor96]* like noise shader, that outputs not only the resulting color, but the four nearest features to the evaluated point, their respective positions, and their cell color IDs. See also *[EMP+02]*.

1.21.1 Parameters

Color Parameters

Color 1 Primary cell color.

Color 2 Secondary cell color.

Contrast Contrast between primary and secondary cell color.

Recursion Parameters

Amplitude Controls the amplitute at each octave, including the starting iteration.

Octaves Number of iterations to perform, higher values lead to increasing detail, but increased computational cost as well.

Lacunarity Defines how large the gaps are in the cell noise with increasing octaves, higher values lead to higher gaps, lower values to small gaps.

Persistence The persistence of the fractal is a gain factor to apply to the amplitude at each iteration, but it only has an effect when the shader is set to the mode *pebbles*.

Cell Parameters

Density The density of the cells, with higher values resulting in a higher number of cells in the same area.

Jittering How random the placement of the cells is, with low values resulting in a ordered grid of cells, and higher values resulting in aleatory placement of cells.

Metric Which metric to choose to calculate the distance from cell to feature points. There are several to choose from, resulting in different types of patterns.

- Euclidian distance[1]
- Sum of square difference
- Tchebychev distance[2]
- Sum of absolute difference
- Akritean distance
- Minkowski metric[3]
- Karlsruhe metric[4]

The sum of the square difference is also known as the Manhattan metric[5].

The Minkowski metric is a generalized metric whose P parameter allows you to go from the Euclidian distance when P has a value of 2, to the Manhattan distance when P has a value of 1, and as P reaches infinity, it represents the Tchebychev metric.

The Akritean distance if a weighted mix of the Euclidian distance, and the Tchebychev distance.

The Karlsruhe metric, also known as Moscow metric, is a radial metric, returns radial sections from a cell at the center.

Minkowski Parameter Controls the metric, with a value of 1 being the Manhattan distance, 2 being the Euclidian distance, and higher values tending to the Tchebychev metric as the parameter approaches infinity.

Coverage The Akritean distance coverage, or the weighting mix between the Euclidian distance and the Tchebychev distance.

Features Mode The features mode to use when computing the output color.

- Feature 1, or nearest feature from the cell
- Feature 2, or second nearest feature from the cell
- Feature 3, or third nearest feature from the cell
- Feature 4, or fourth nearest feature from the cell
- F1 + F2, or sum of first and second nearest features
- F2 - F1, or difference between second and first nearest features
- F1 * F2, or product of first and second nearest features
- F1 / F2, or division of first nearest feature by second nearest feature
- F1 ^ F2, nearest feature raised to the second nearest feature
- Pebbles, a mode that resembles pebbles
- Cell ID 1, the ID of the nearest feature to the cell
- Cell ID 2, the ID of the second nearest feature to the cell

[1] The Euclidian distance or Euclidian metric, also known as L_2 norm, is the straight line distance between two points in Euclidian space.

[2] The Chebyshev (or Tchebychev) distance, also known as Chessboard distance or L_∞ norm, is a metric on a vector space where the distance between two vectors is the greatest of their differences along any coordinate dimension.

[3] The Minkowski distance, also known as L_P norm, is a metric which is a generalization of both the Euclidian distance and the Manhattan distance, being equal to the Manhattan distance when its *p* parameter is equal to 1, or equal to the Euclidian distance when its *p* parameter is equal to 2. On the limit as *p* approaches infinity, it is equal to the Chebyshev distance.

[4] In metric geometry, the Karlsruhe metric is a measure of distance that assumes travel is only possible along rays through the origin and circular arcs centered at the origin.

[5] The Manhattan distance, also known as *taxicab* metric or L_1 norm, is a metric in which the distance between two points is the sum of the absolute differences of their Cartesian coordinates.

- Cell ID 3, the ID of the third nearest feature to the cell
- Cell ID 4, the ID of the fourth nearest feature to the cell

Note: The unmodified features, points and their color IDs are also output from the shader, giving the user greater creative potential. The feature modes above are but a starting point.

Color Balance

The standard Maya color balance, gain, offset parameters. Please consult Maya's documentation for more information on these controls.

Effects

The standard Maya effects parameters. Please consult Maya's documentation for more information on these controls.

Coordinates

Typically, the *placement 3d* node's placement matrix, which provides a placement matrix to transform the surface point providing the x,y,z coordinates. By default this point is the global primitive variable **P**, but the user can override this if needed.

1.21.2 Outputs

Output Color The color resulting from ghe *Features Mode* choice.

Output Alpha The alpha resulting from the *Features Mode* choice, usually luminance of the color only.

Output Features An array of 4 floats, containing the four nearest features to the cell.

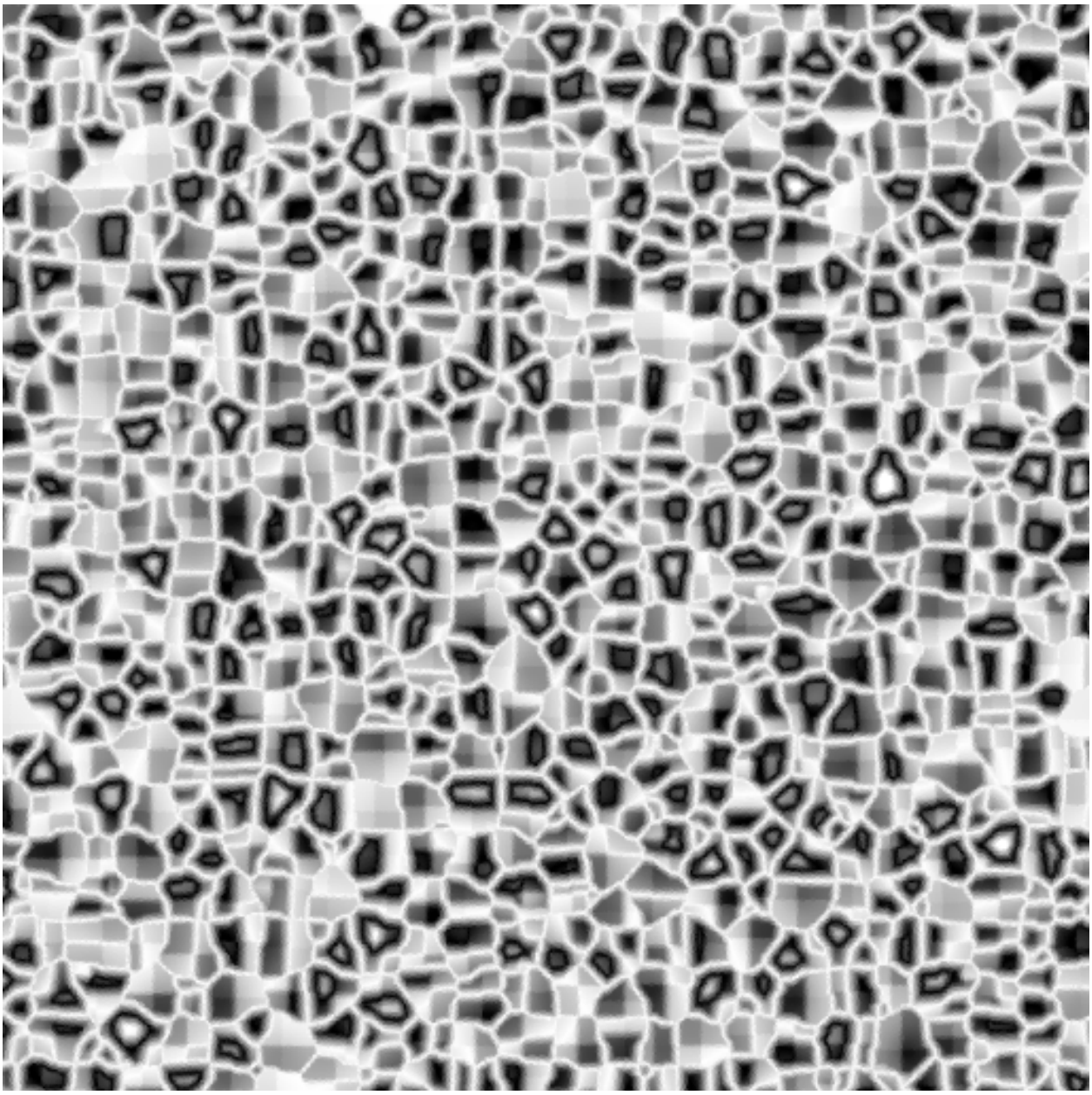
Output Positions An array of 4 points, containing the center of the four nearest features to the cell.

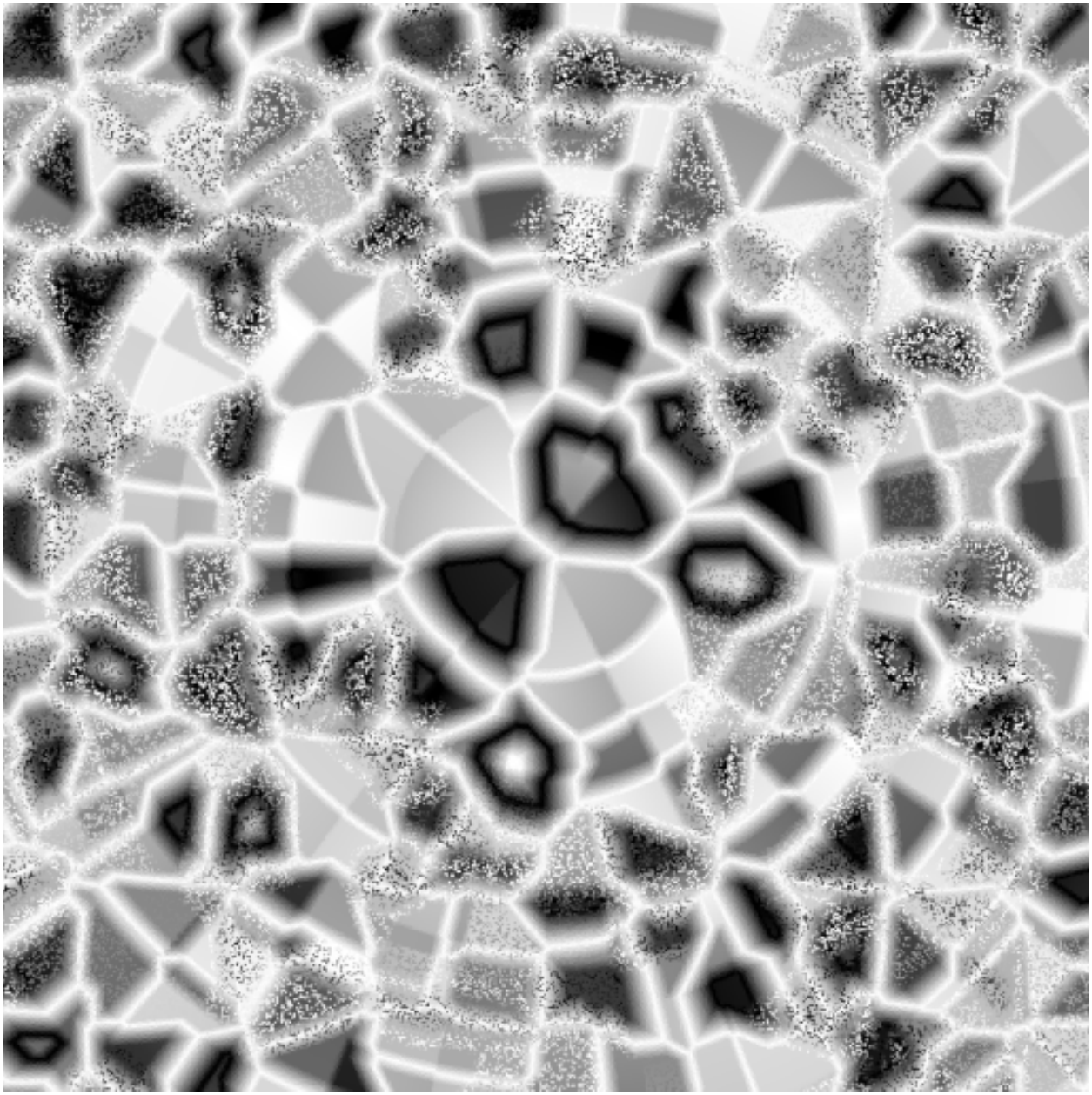
Output IDs An array of 4 colors, containing the color IDs of the four nearest features to the cell.

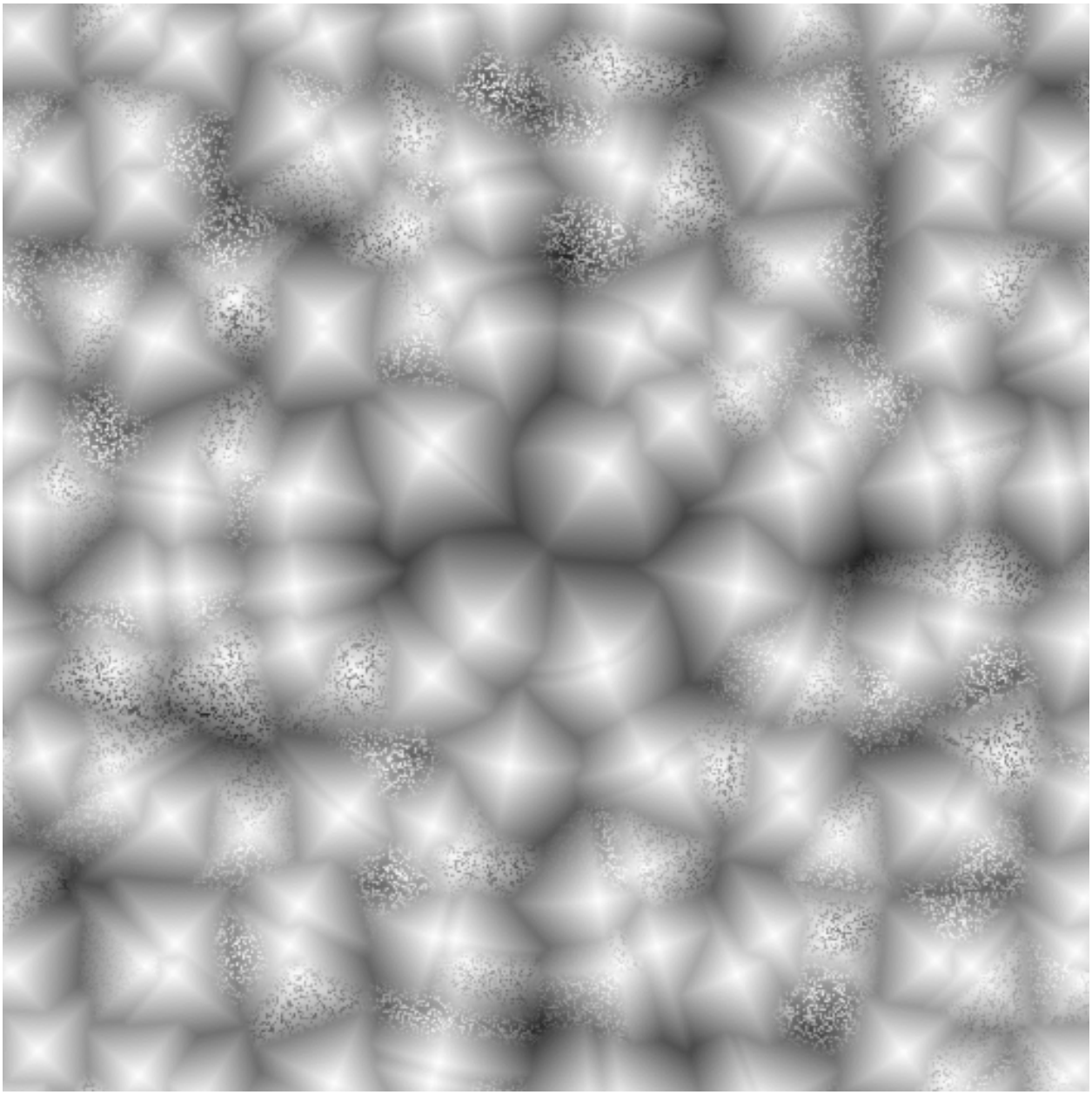
Warning: presently OSL does not allow connections from/to array elements, and appleseed-maya is not enabling the array outputs for now. This will be addressed in a future release.

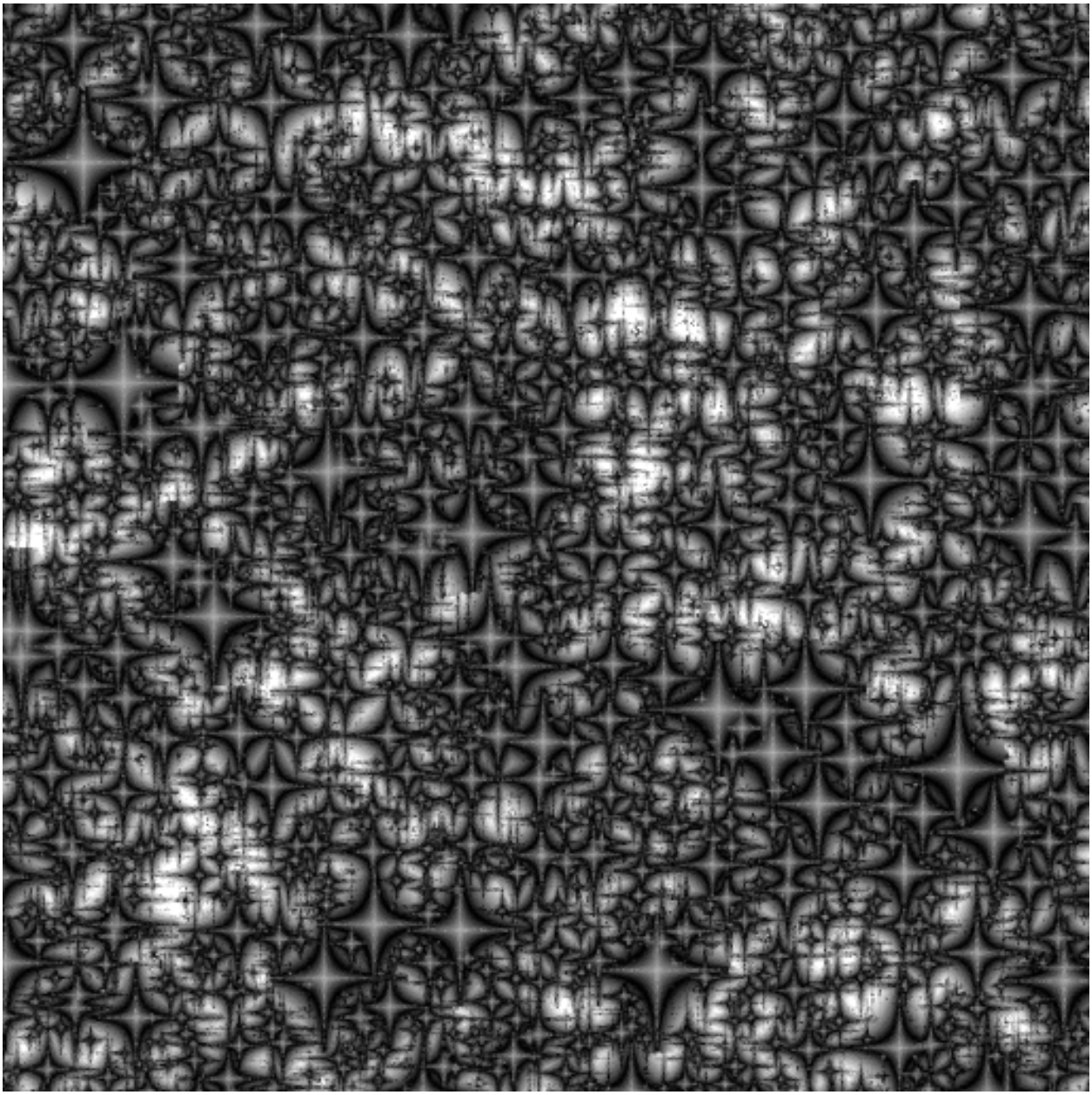
1.21.3 Screenshots

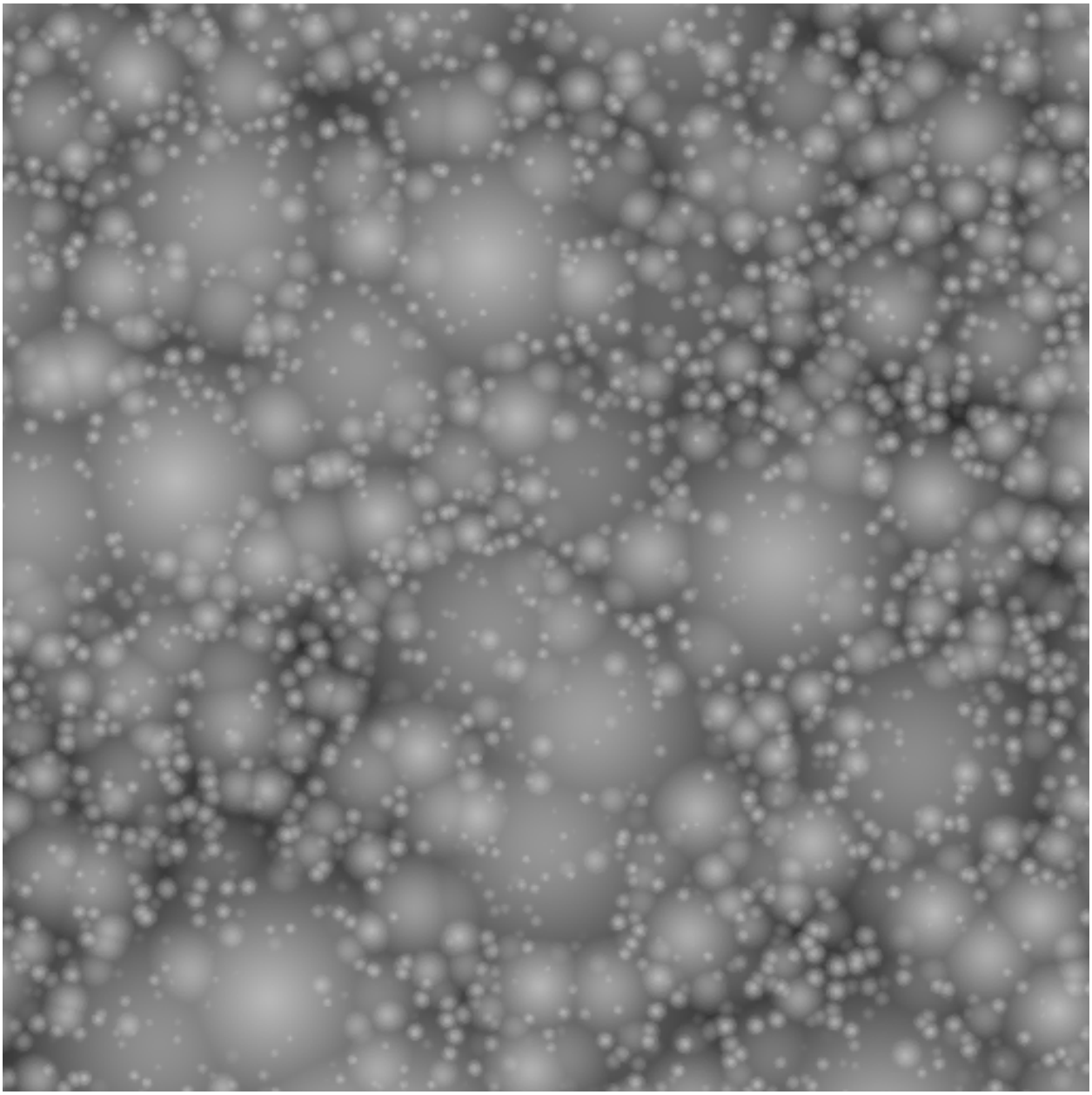
Some examples of different metrics and feature output combinations used.

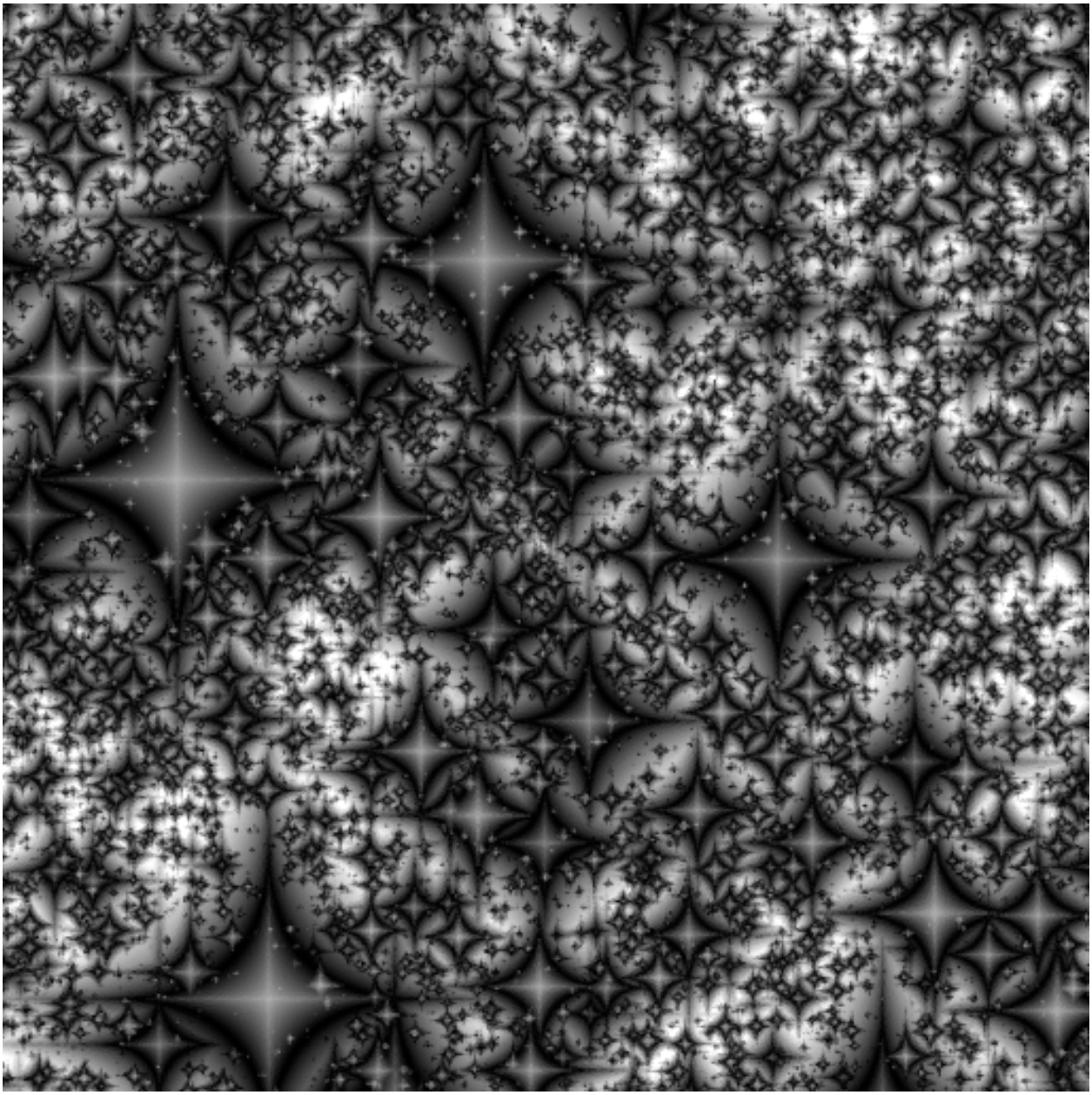


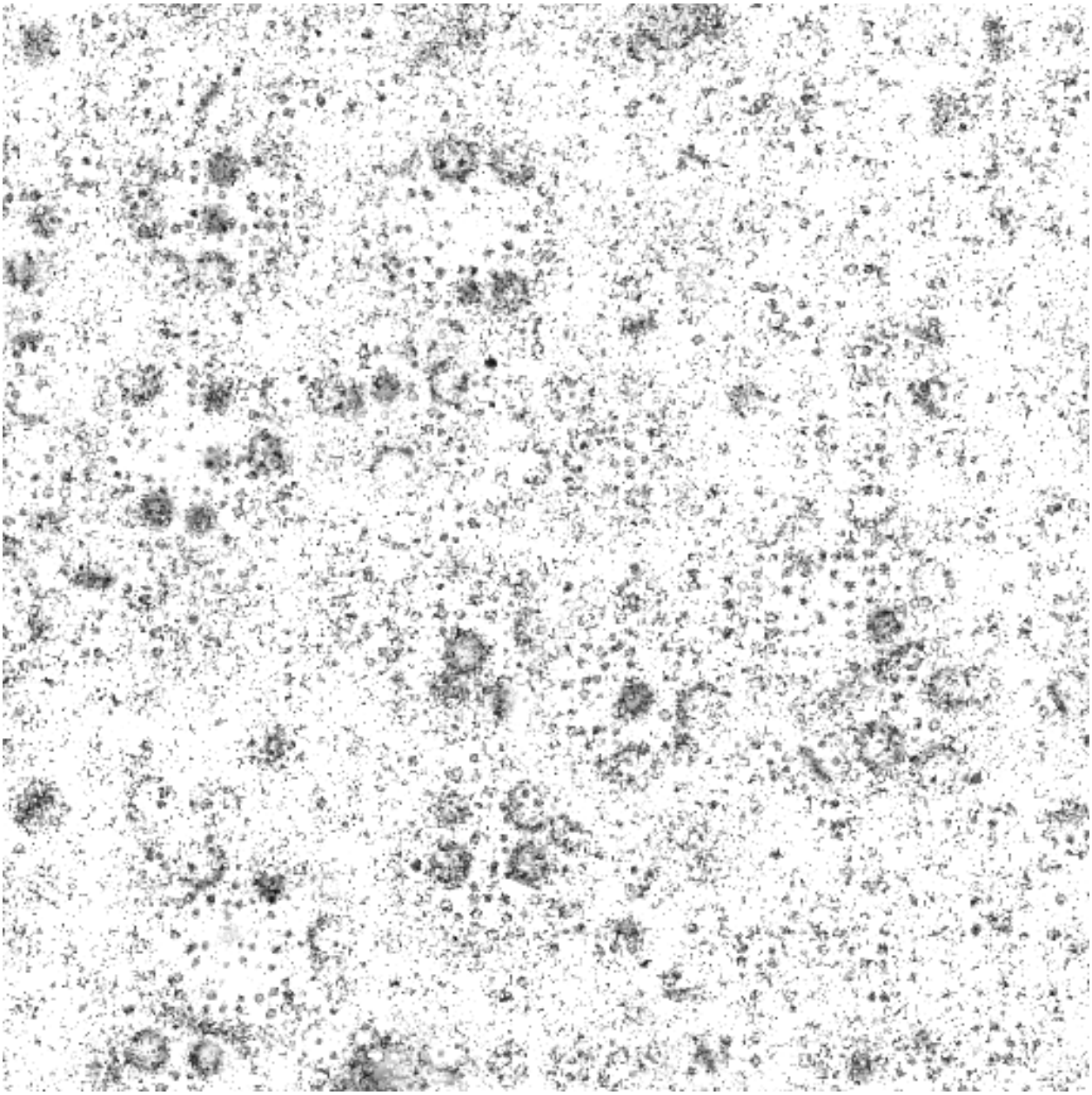


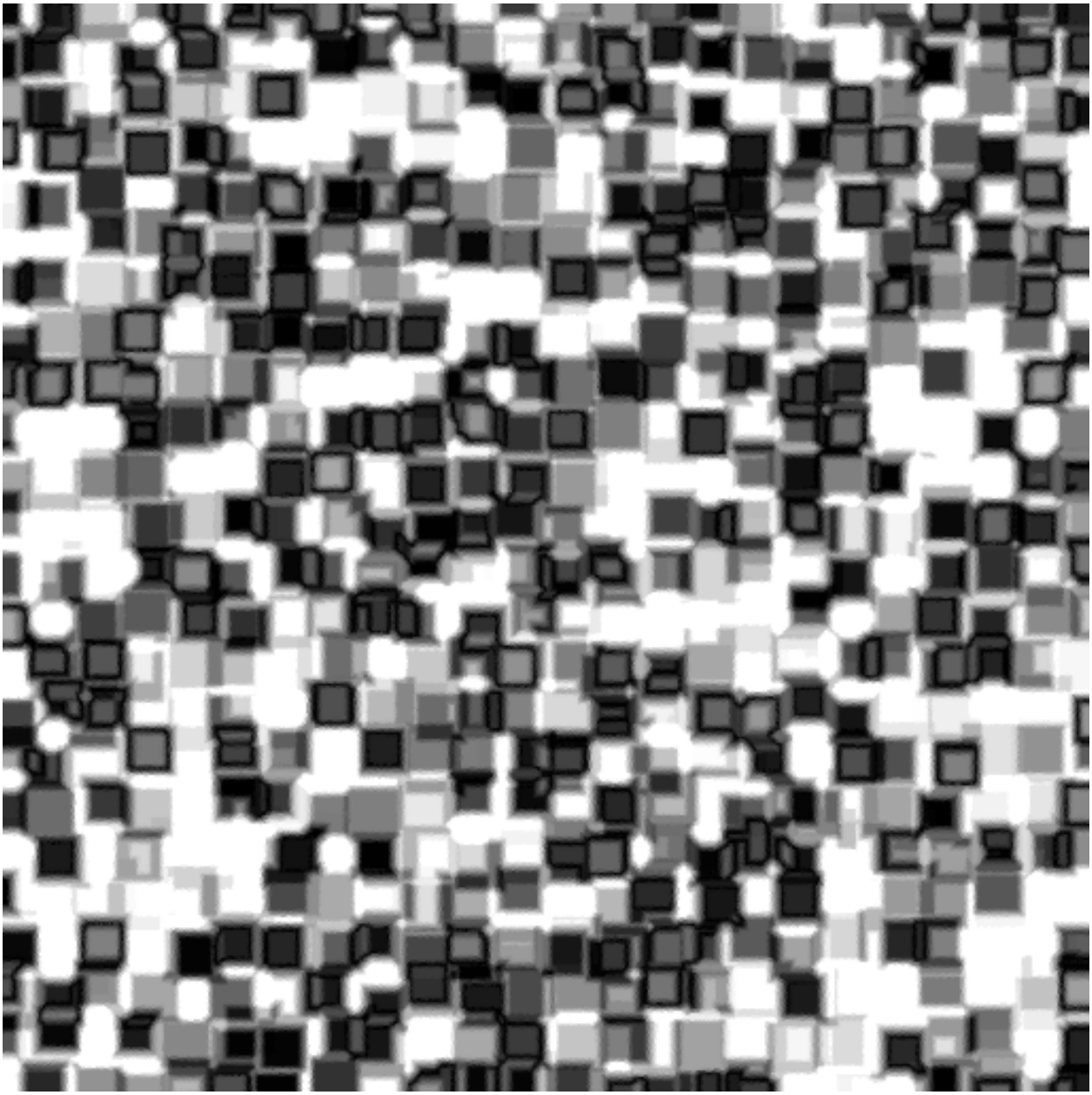




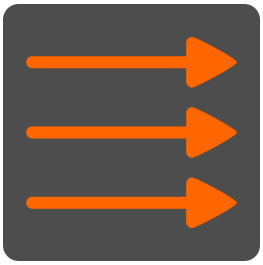








References



1.22 asAnisotropyVectorField

A node that manipulates anisotropy vector maps and outputs a anisotropy vector for use in BxDFs that support anisotropy.

1.22.1 Parameters

Field Parameters

Field Color The anisotropy vector map color.

Rotation Angle A rotation angle, in degrees, from -360 to 360 degrees, to rotate the input anisotropy vector map. This is an absolute value to re-orient the initial anisotropy vector map, non-texturable.

Field Mode The input anisotropy vector map can have the anisotropy directions along X and Y, encoded in the Red and Green channels, or in the Red and Blue channels. Typically, RG encoding is used in applications such as RenderMan[1], while RB encoding is used in applications such as Modo[2] or Softimage[3]. The values allowed are therefore

- Red/Green as XY
- Red/Blue as XY

Rotation Parameters

Rotation Value This parameter rotates the RG or RB vector field, and unlike the *Rotation Angle* parameter, this parameter takes as inputs values between 0.0 and 1.0, and is texturable. Internally this is mapped to [0,360] degrees.

Rotation Mode The rotation computed earlier in *Rotation Value* can be added to the vector field, or the rotation can be applied to both directions along a central axis. That is, the rotation can be mapped from the input [0,1] values to [0,360] degrees (Absolute), or to [-360,360] degrees (Centered) - where a value of 0.5 means no rotation. Values below 0.5 means counter-clockwise rotation, and values above 0.5 imply clockwise rotation. The allowed modes are therefore

[1] See RenderMan's tangent field for details on PRman's implementation.

[2] See Modo anisotropy help for details on the Modo implementation. Unlike PRman's, the XY data is encoded in R and B channels, not R and G.

[3] See Softimage anisotropy patterns. Like Modo, the anisotropy data is encoded in the R and B channels.

- Centered
- Absolute

Output Parameters

Normalize Output Parameter that forces a normalization of the final anisotropy vector. The default is to enable it.

Surface Normal

Surface Normal The world space, unit length shading normal, bump mapped. **Not** a tangent space nor a object space normal.

1.22.2 Outputs

Anisotropy Vector The final re-oriented, RG or RB encoded anisotropy vector.

1.23 asAttributes

A node that exposes to the user the general attributes specified in OSL[1] and the appleseed renderer specific attributes.

1.23.1 Parameters

This node has no input parameters.

---

[1] OSL or Open Shading Language.

1.23.2 Outputs

Assembly Instance ID An integer value containing the ID of an assembly[2] instance.

Assembly Instance Name A string containing the name of the assembly instance.

Assembly Name A string containing the name of the assembly.

Camera Clip Far The farthest camera clipping plane.

Camera Clip Near The nearest camera clipping plane.

Camera FOV The camera Field Of View[3].

Camera Pixel Aspect The camera pixel aspect.

Camera Projection The camera projection.

Camera Resolution X The camera X resolution.

Camera Resolution Y The camera Y resolution.

Screen Window X Max The screen window *x* maximum value.

Screen Window X Min The screen window *x* minimum value.

Screen Window Y Max The screen window *y* maximum value.

Screen Window Y Min The screen window *y* minimum value.

Shutter Close Time The shutter closing time.

Shutter Open Time The shutter opening time.

Object Instance ID An integer containing an object instance's ID.

Object Instance Index An integer containing an object instance's index.

Object Instance Name A string containing the object instance's name.

Object Name A string containing an object's name.

Ray Differentials An boolean denoting if the path being traced has ray differentials.

Ray Depth A integer denoting the present ray depth.

Ray IOR The absolute index of refraction of the medium the present ray is travelling in. This is affected by nested dielectrics *[SB02]*.

Ray Length The current path's length.

References

[2] See appleseed's wiki entry for assemblies concepts and procedural assemblies.
[3] See *field of view wikipedia page <https://en.wikipedia.org/wiki/Field_of_view>* for more details.

1.24 asBlendColor

A node that allows the user to blend an input color with a second input color according to preset modes, similar to what is commonly found in image editing applications.

1.24.1 Parameters

Color Parameters

Source Color The input RGB color.

Source The input color weight.

Blend Color The second input color.

Blend Weight The second input color weight.

Blend Mode The blend mode[1] to use for the respective colors. Users of applications such as GIMP or Adobe® Photoshop® should be familiar with these blend operations. They can take the following values

- Darken
- Multiply
- Color Burn
- Linear Burn
- Lighten
- Screen
- Color Dodge
- Linear Dodge
- Overlay
- Soft Light
- Hard Light
- Vivid Light

[1] See this page on blend modes for more information.

- Linear Light
- Pin Light
- Difference
- Exclusion
- Subtract
- Divide
- Hue
- Saturation
- Color
- Luminosity

See also:

Merging and Transformation of Raster Images for Cartoon Animation [Wal81].

Clamp Output Checking this checkbox will clamp the output to the [0,1] range.

1.24.2 Outputs

Output Color The blended color result.

References

1.25 asBump

A node that allows the user to apply scalar bump mapping, or normal mapping.

1.25.1 Parameters

Mode Parameters

Mode The choice of bump mapping algorithm to use. It can be one of:

- Bump *[Bli78]*
- Normal Map *[COM98]*, *[CMSR98]*

Bump Parameters

Bump Value A scalar value controlling the magnitude of the bump effect.

Bump Depth A scalar value controlling the bump depth of the bump effect. Unlike the *bump value* which expects a value in [0,1] range, the *bump depth* can be a positive or negative value, in which case it will apply the bump effect outwards or inwards respectively.

Normal Map

Normal Map Weight A scaling factor that defines the contribution weight of the normal map. With a value of 0.0, no contribution takes place and the regular surface normal is used. A value of 1.0 defines full contribution of the normal map input.

Normal Map The input normal map color.

Map Coordinate System The coordinate system of the input normal map used[1]. It can be one of:

- *Tangent Space*
- *Object Space*
- *World Space*

Advanced Parameters

Map Signedness The signedness of the input normal map. If your map is in [-1,1] range, use *Signed*. If your map is in [0,1] range, used *Unsigned*.

Flip R Flip the *red* channel of the input tangent space normal map.

Flip G Flip the *green* channel of the input tangent space normal map.

Swap RG Swap the *red* and *green* channels of the input tangent space normal map.

Note: The channel flipping and swapping options **only** have effect on the tangent space normal maps. They are ignored when the *Map Coordinate System* is *Object Space* or *World Space*.

---

[1] Usually one uses *tangent space* normal maps, but the option is provided here to use *object* and *world* space normal maps, which sometimes can be exported from other applications.

Surface Parameters

Surface Normal The base surface normal to use. It can be the result of a previous bump node, or the global variable *N* from the *asAttributes node*. If not set, it defaults to the (world space) surface normal N.

1.25.2 Outputs

Result The unit length world space bumped normal.

References

1.26 asColorTransform

A node that transforms a color from **any** input color model, to **any** output color model, respecting the color space definitions set in synColor CMS options. An option will be added later to override this, in a way similar to what is presently done in *the luminance shader*.

1.26.1 Parameters

Color Attributes

Input Color The color being transformed.

Input Space The input color space to transform from. It can be one of

- RGB
- HSV[1]
- HSL[2]

[1] Hue, Saturation, Value color space, https://en.wikipedia.org/wiki/HSL_and_HSV
[2] Hue, Saturation, Lightness color space, https://en.wikipedia.org/wiki/HSL_and_HSV

- CIE XYZ[3]
- CIE xyY[4]
- CIE 1976 L*a*b* *[27]*
- CIE 1976 L*u*v* *[34]*
- CIE 1976 LCh $_{ab}$[5]
- CIE 1976 LCh $_{uv}$[6]

Output Space The output color space to transform to. It can be one of

- RGB
- HSV
- HSL
- CIE XYZ
- CIE xyY
- CIE 1976 L*a*b*
- CIE 1976 L*u*v*
- CIE 1976 LCh $_{ab}$
- CIE 1976 LCh $_{uv}$

See also:

The Colour Python colour science package for extensive information on the topic :cite:‘ *[25]*

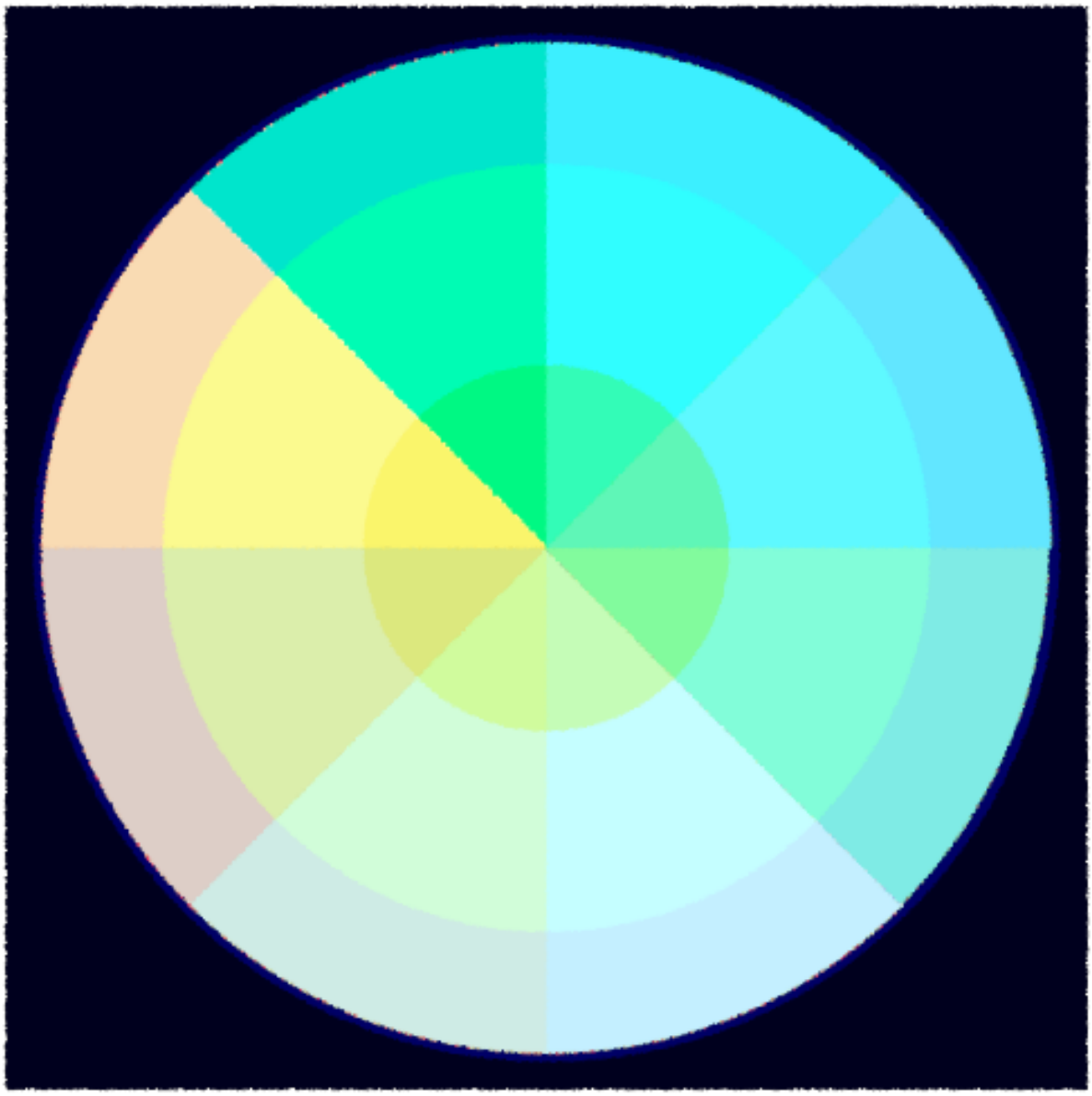
1.26.2 Outputs

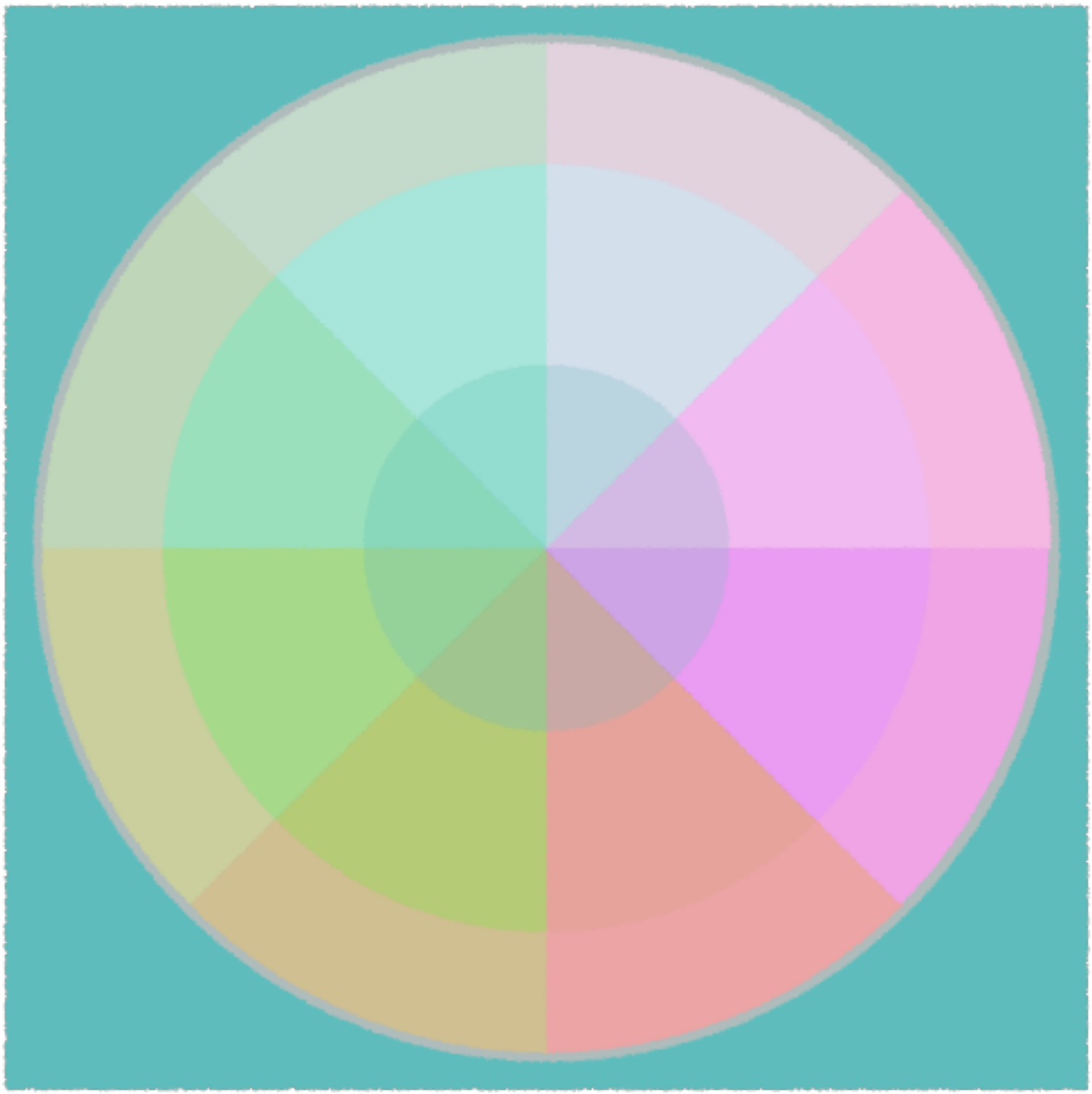
Result The transformed color. For usability, the range of some spaces are remapped to the [0,1] range. As an example, the hue could be mapped to a greyscale texture, or if the space is set to CIE 1976 L*a*b*, the *a* variable would control the green/magenta oposition, while the *b* variable would control the blue/yellow oposition, with respective neutral/grey values at 0.5.

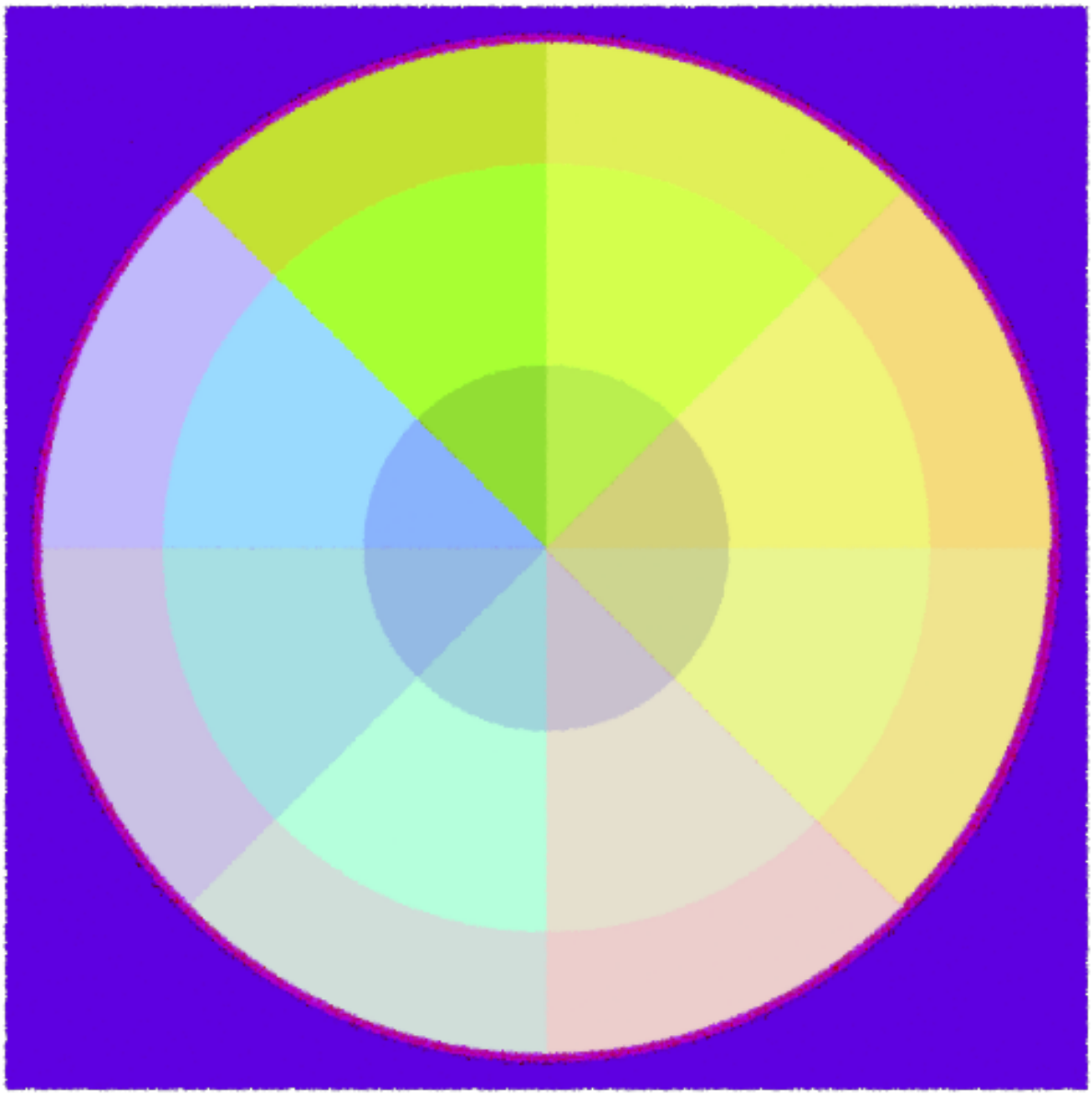
1.26.3 Screenshots

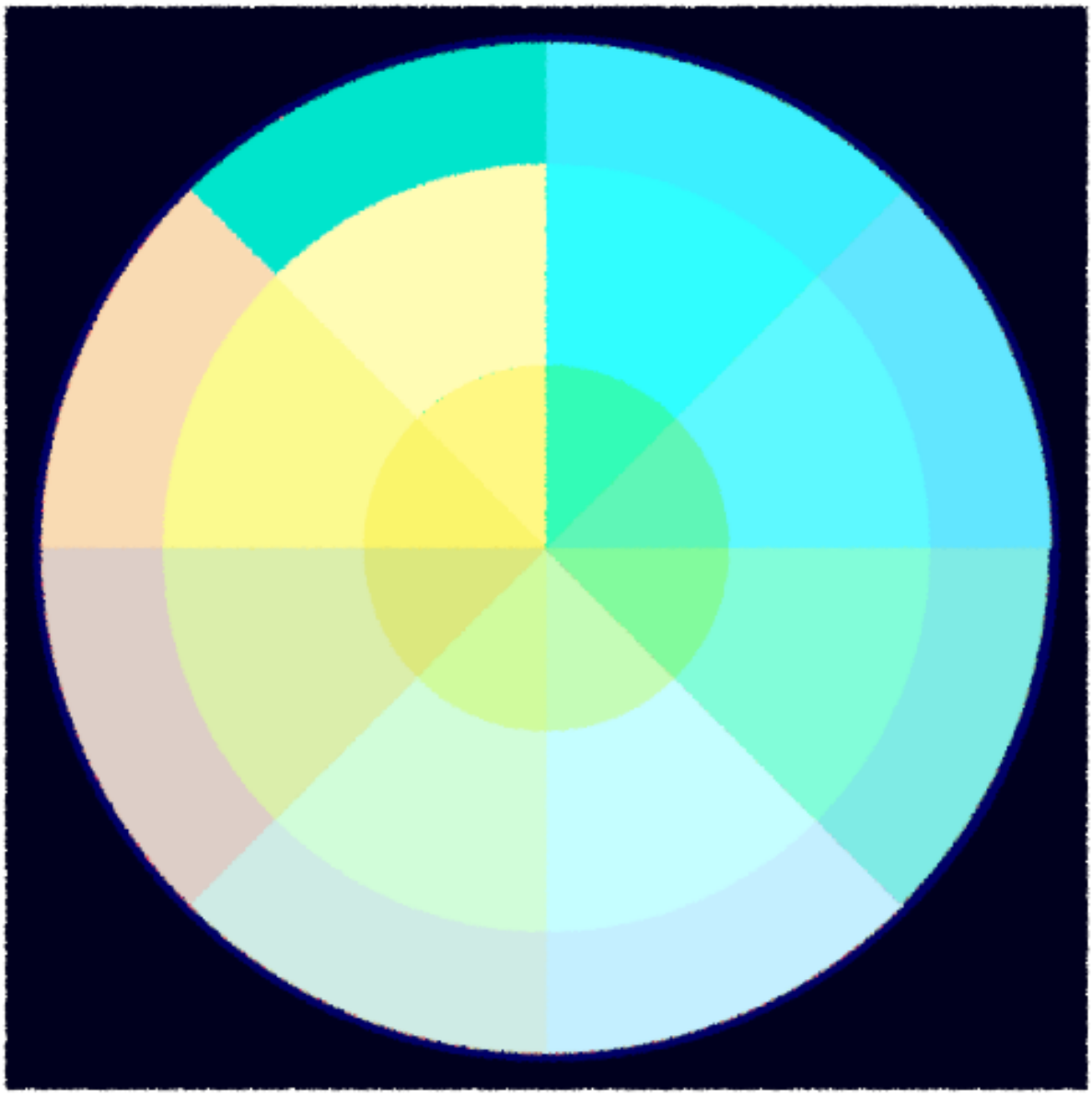
Some examples of color transformations.

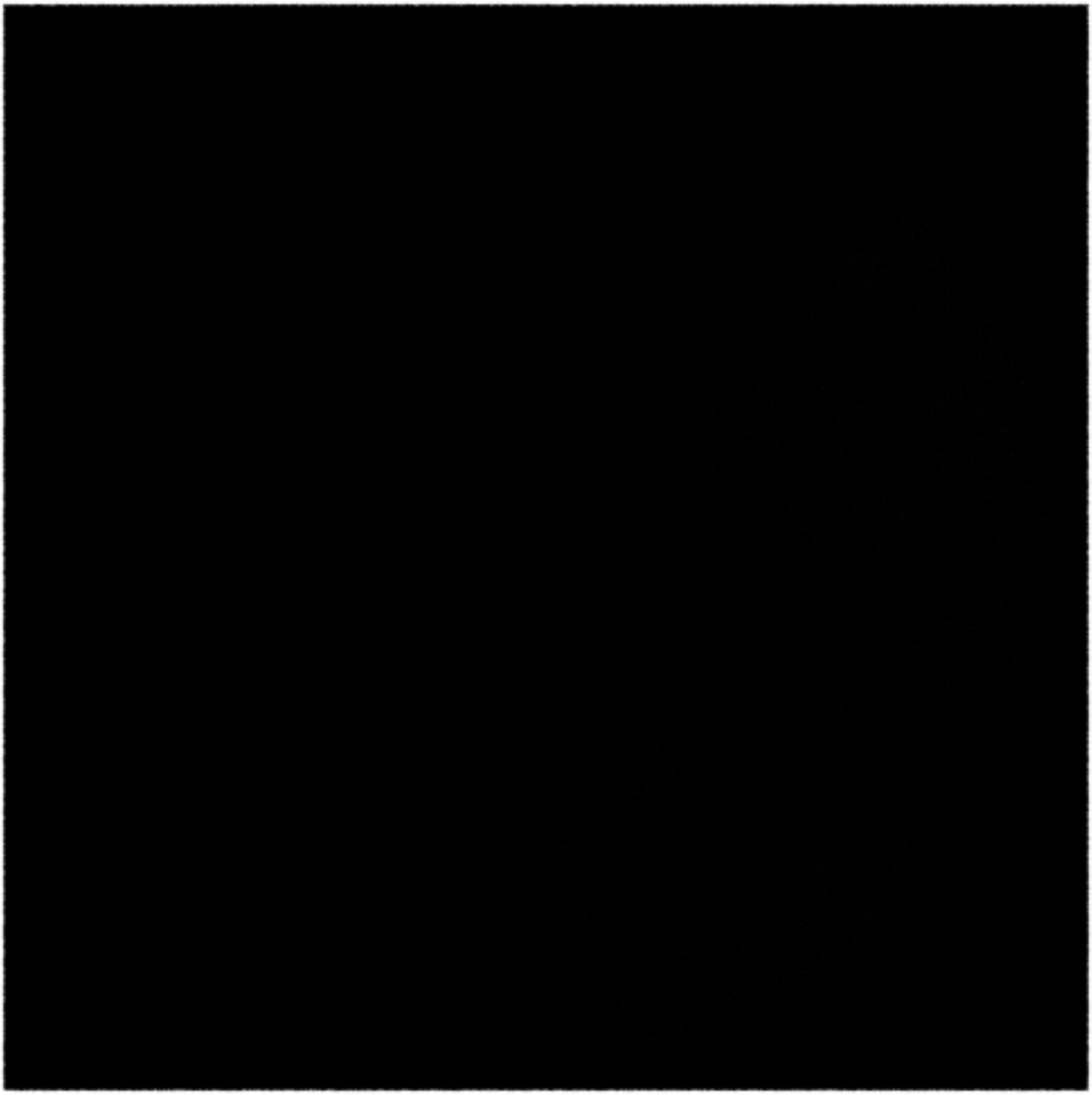
[3] The CIE XYZ color space, https://en.wikipedia.org/wiki/CIE_1931_color_space
[4] CIE xyY, https://en.wikipedia.org/wiki/CIE_1931_color_space
[5] Cylindrical representation of the CIELAB color space, LCh $_{ab}$ co, https://en.wikipedia.org/wiki/Lab_color_space#Cylindrical_representation:_CIELCh_or_CIEHLC
[6] Cylindrical representation of the CIELUV color space, CIE LCh $_{uv}$, https://en.wikipedia.org/wiki/Lab_color_space#Cylindrical_representation:_CIELCh_or_CIEHLC











References

1.27 asCompositeColor

A node to composite two RGBA inputs using Porter-Duff *[PD84]* compositing operators.

1.27.1 Parameters

Color Parameters

Source Color The source color.

Source Alpha The source alpha channel.

Destination Color The destination color.

Destination Alpha The destination alpha channel.

Composite Mode The composite operator[1] to use. It can take the following values

- Source
- Destination
- Over
- Under
- In
- Mask
- Out
- Stencil
- Atop
- Dst-Atop
- Xor
- Matte

See also:

Merging and Transformation of Raster Images for Cartoon Animation [Wal81] and the W3.org webpage for a detailed view on compositing algebra in general.

Clamp Output Checking this checkbox will clamp the output into the [0,1] range.

[1] The original Porter-Duff paper (PDF file) can be found here.

1.27.2 Outputs

Output Color The composited color.

Output Alpha The alpha with the result of the compositing operation.

References

1.28 asCreateMask

A node that allows the user to create a greyscale mask of a input color or texture.

1.28.1 Parameters

Color Parameters

Input Color The color used to create the mask from. It's expected to be *scene-linear*.

Input Alpha The alpha channel of the color used to create the mask from.

Threshold Channel The channel to use when applying a threshold in order to build a mask. It can be one of

- Red
- Green
- Blue
- Alpha

- Hue
- Saturation
- Value
- CIELAB L*[1]
- CIELAB a*
- CIELAB b*
- Average
- Luminance[2]

Threshold Value The threshold value to apply to the channel chosen.

Threshold Function The function to use for the threshold. It can be one of

- None
- Step
- Linear Step
- Smooth Step
- Exponential
- Double Circled Seat[3]
- Double Circled Sigmoid[4]
- Smoother Step[5]
- Smoothest Step[6]

Threshold Contrast The contrast to apply when the function is *Double Circled Seat* (a contrast flattening curve) or *Double Circled Sigmoid* (a contrast increasing curve).

Threshold Lower Bound The lower bound for the *smoothstep*, *smootherstep* and *smootheststep* functions.

Threshold Upper Bound The upper bound for the *smoothstep*, *smootherstep* and *smootheststep* functions.

1.28.2 Outputs

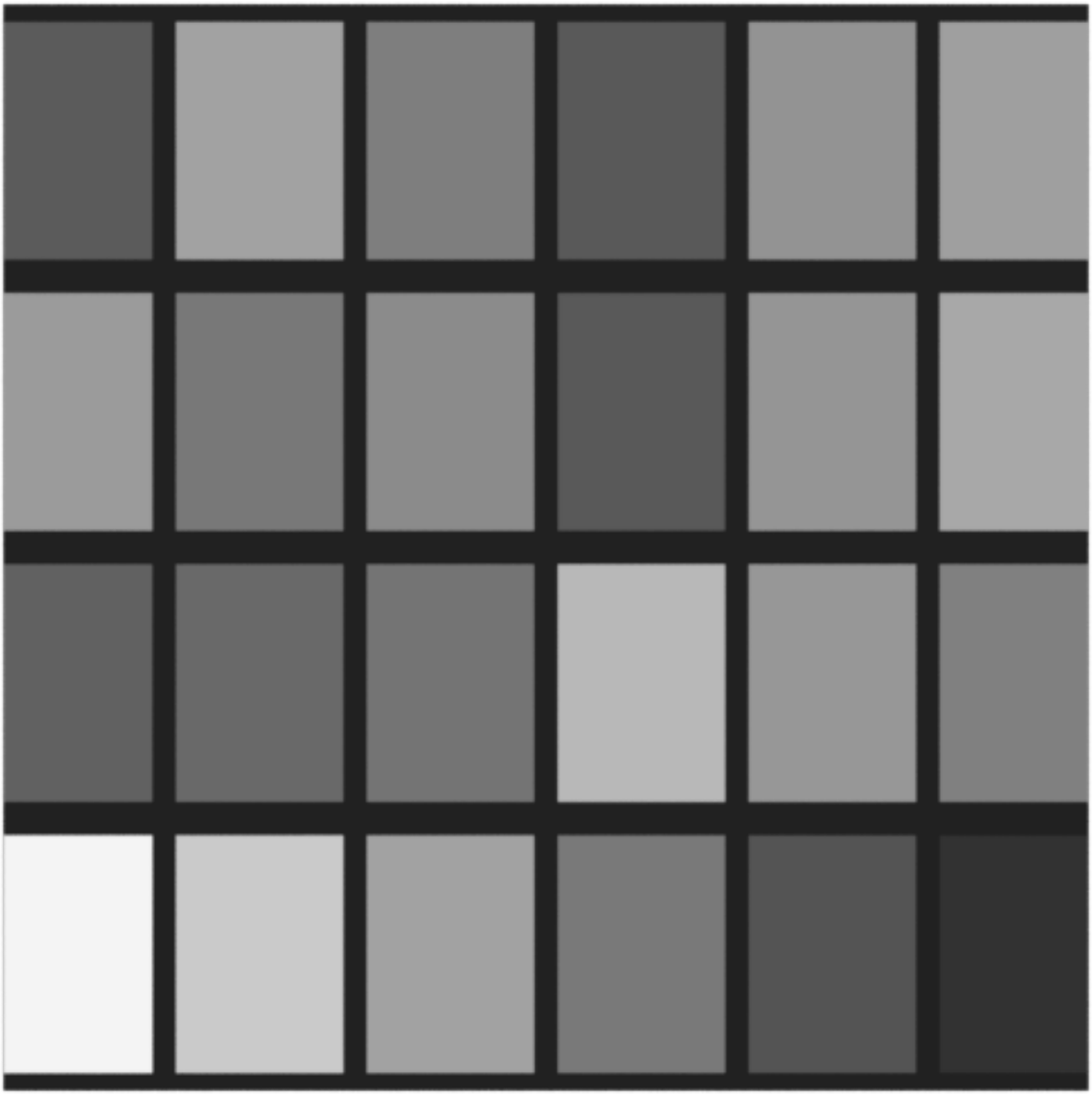
Result The new mask.

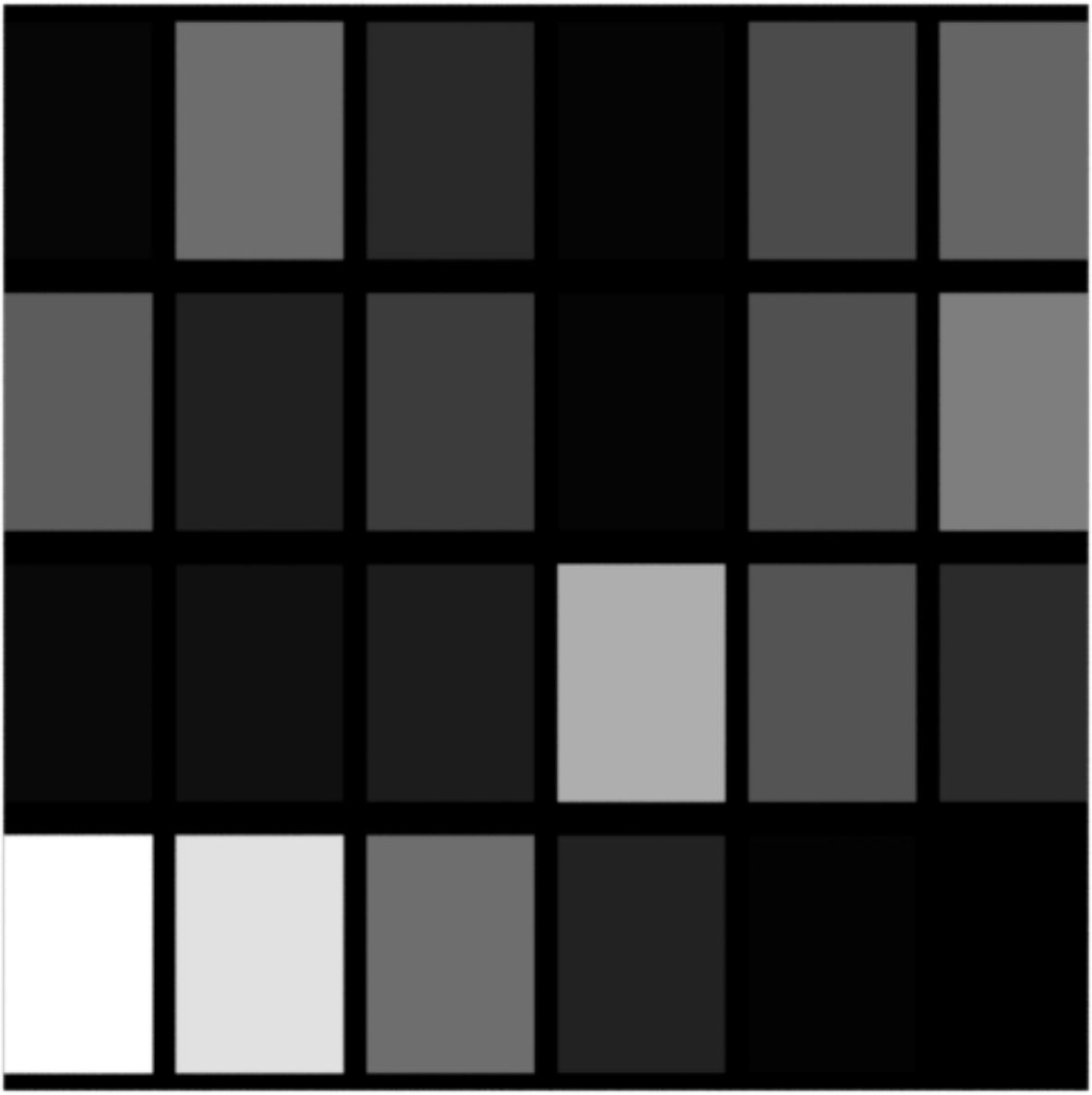
---

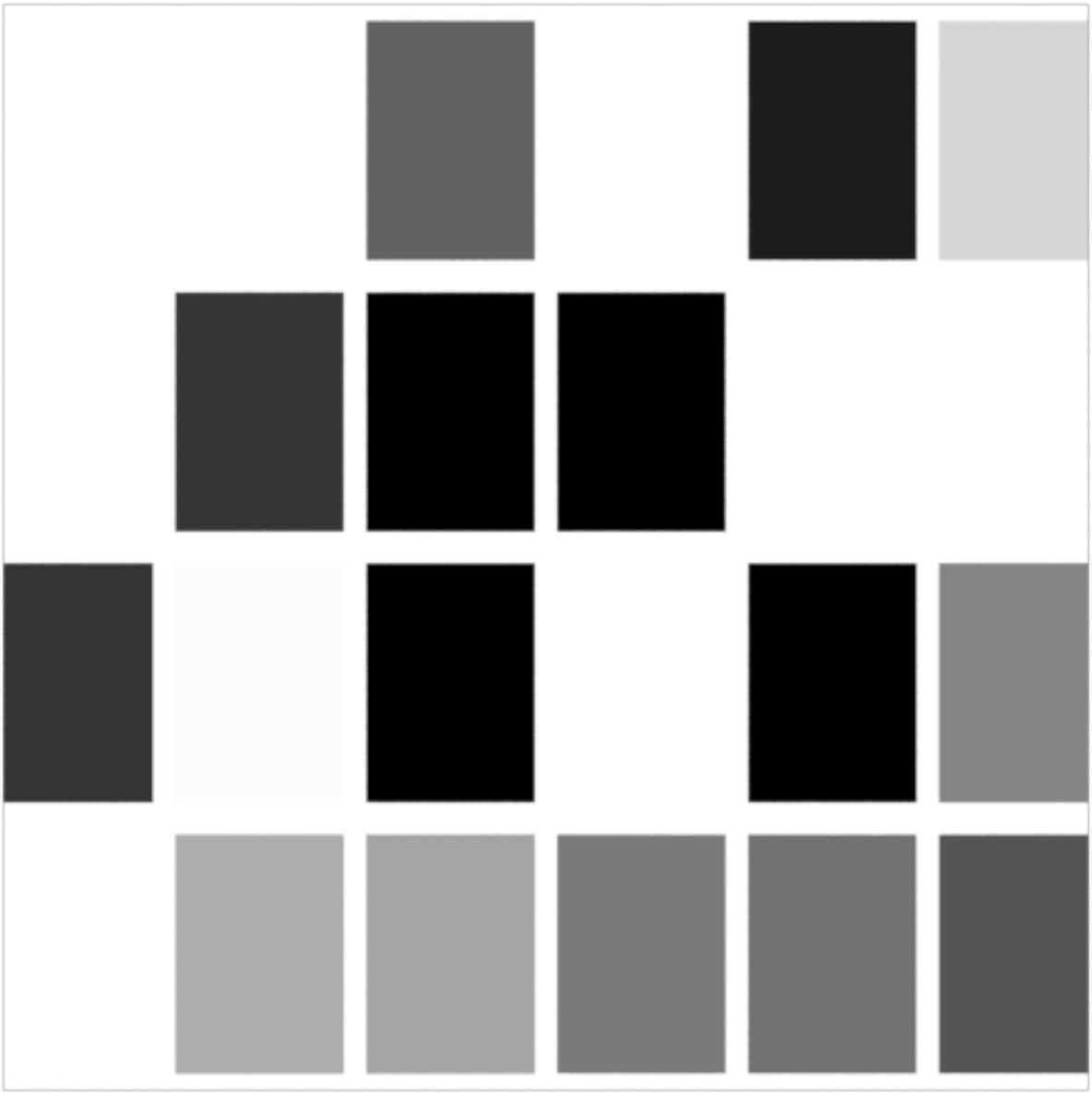
[1] CIELAB or CIE 1976 L*a*b* color space.
[2] For this this assumes the input color is using the ITU-R BT.709/Rec.709 RGB primaries. Once support for working or rendering space using other primaries other than Rec.709 is added to appleseed, this will be extended.
[3] A contrast flattening function, see Double Circled Seat function.
[4] A contrast increasing function, see Double Circled Sigmoid function.
[5] A smoother *smoothstep* function, with 0 first and second derivatives at x=0 and x=1.
[6] Like *smootherstep*, but with 0 third derivatives at x=0, and x=1.

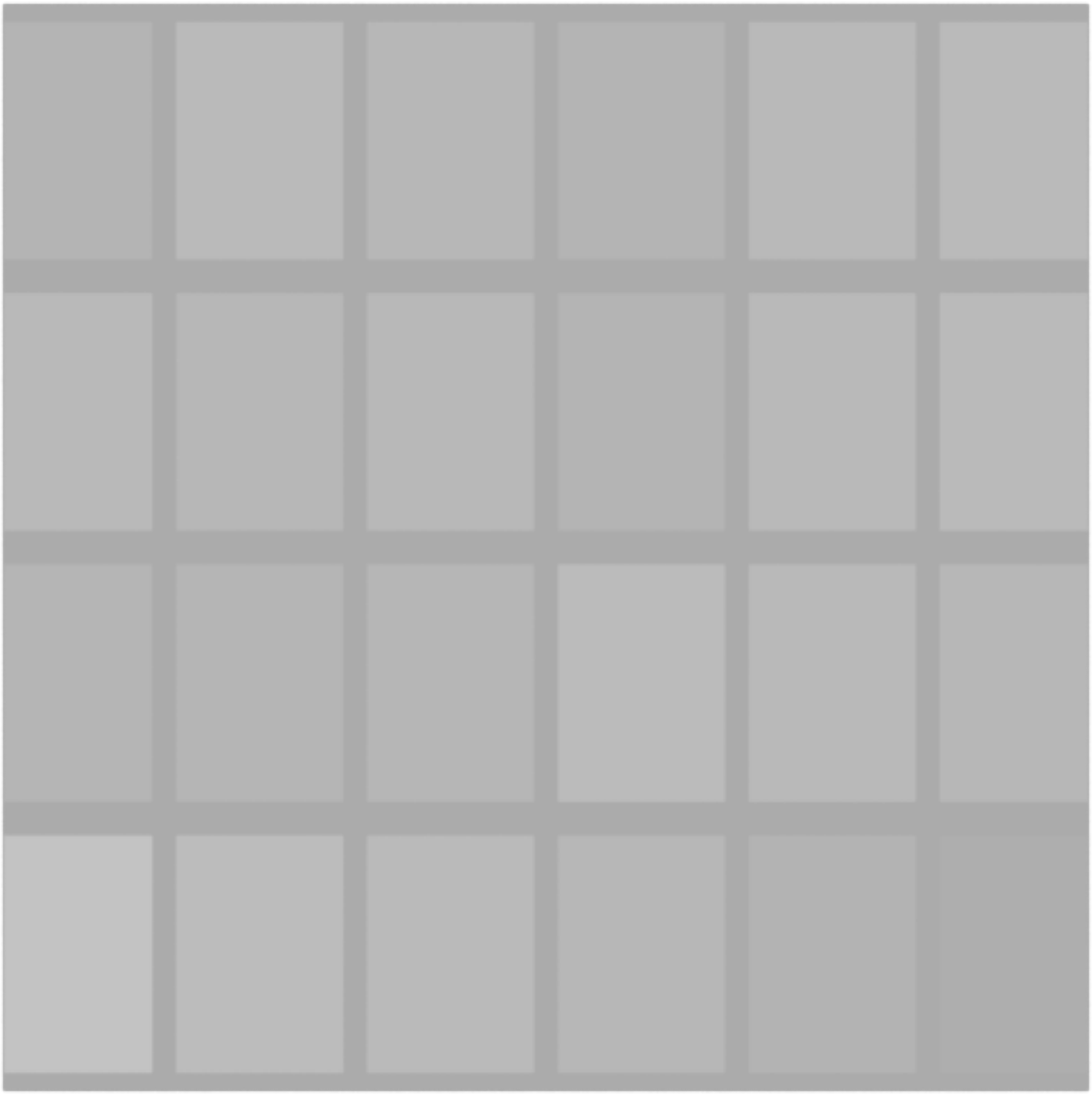
1.28.3 Screenshots

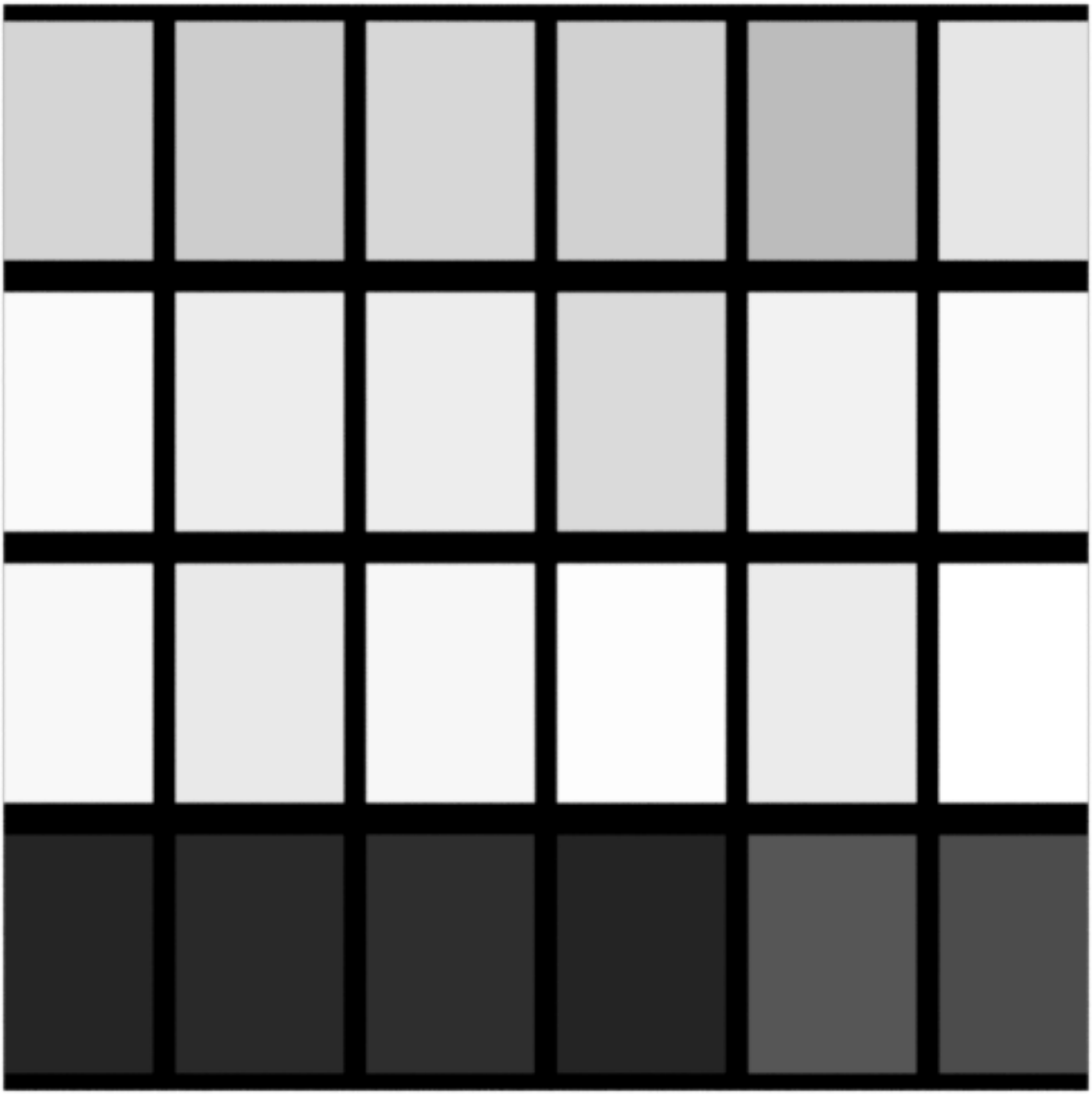
Some examples showing some of the masks created with the modes outlined above.

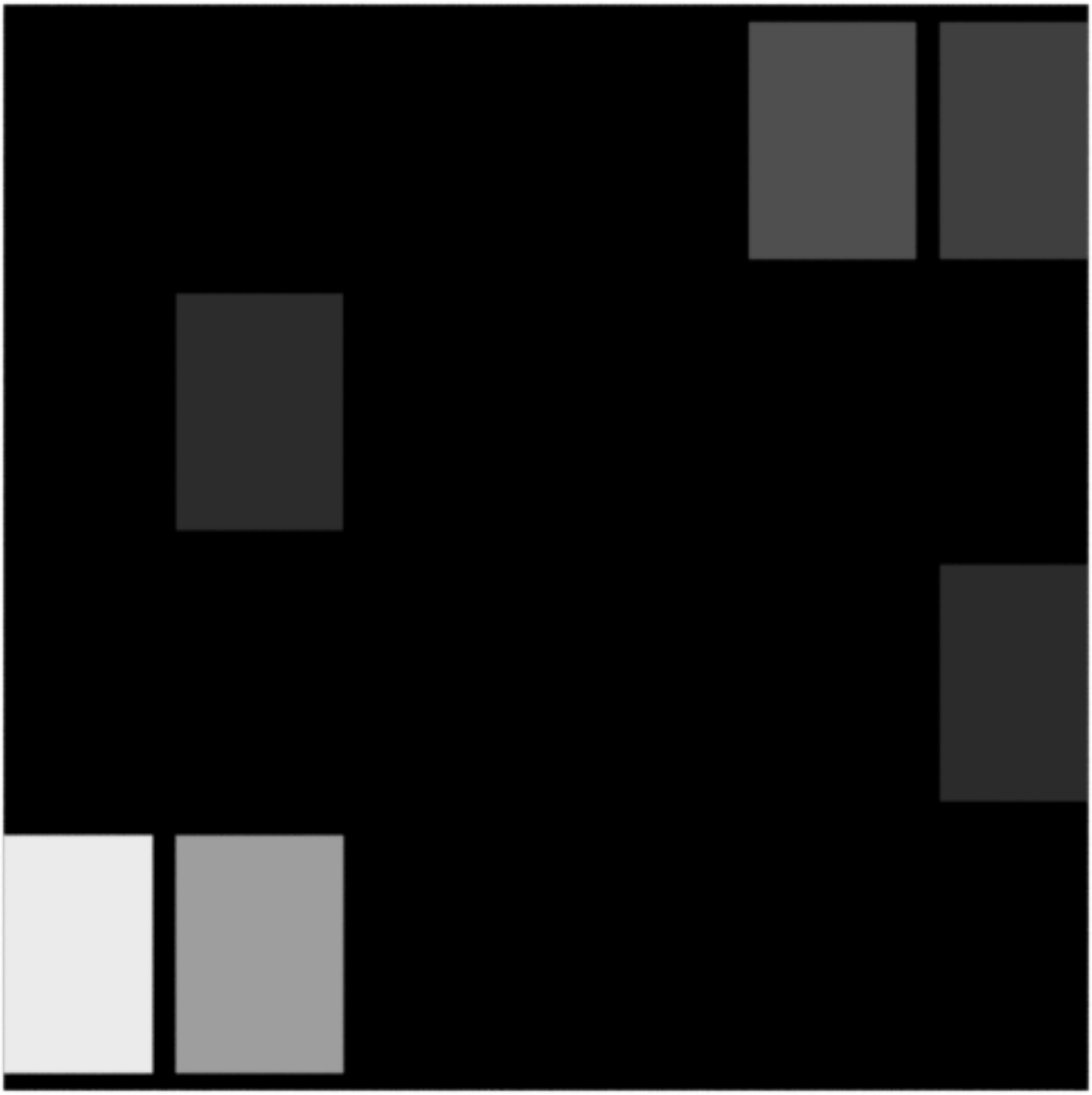


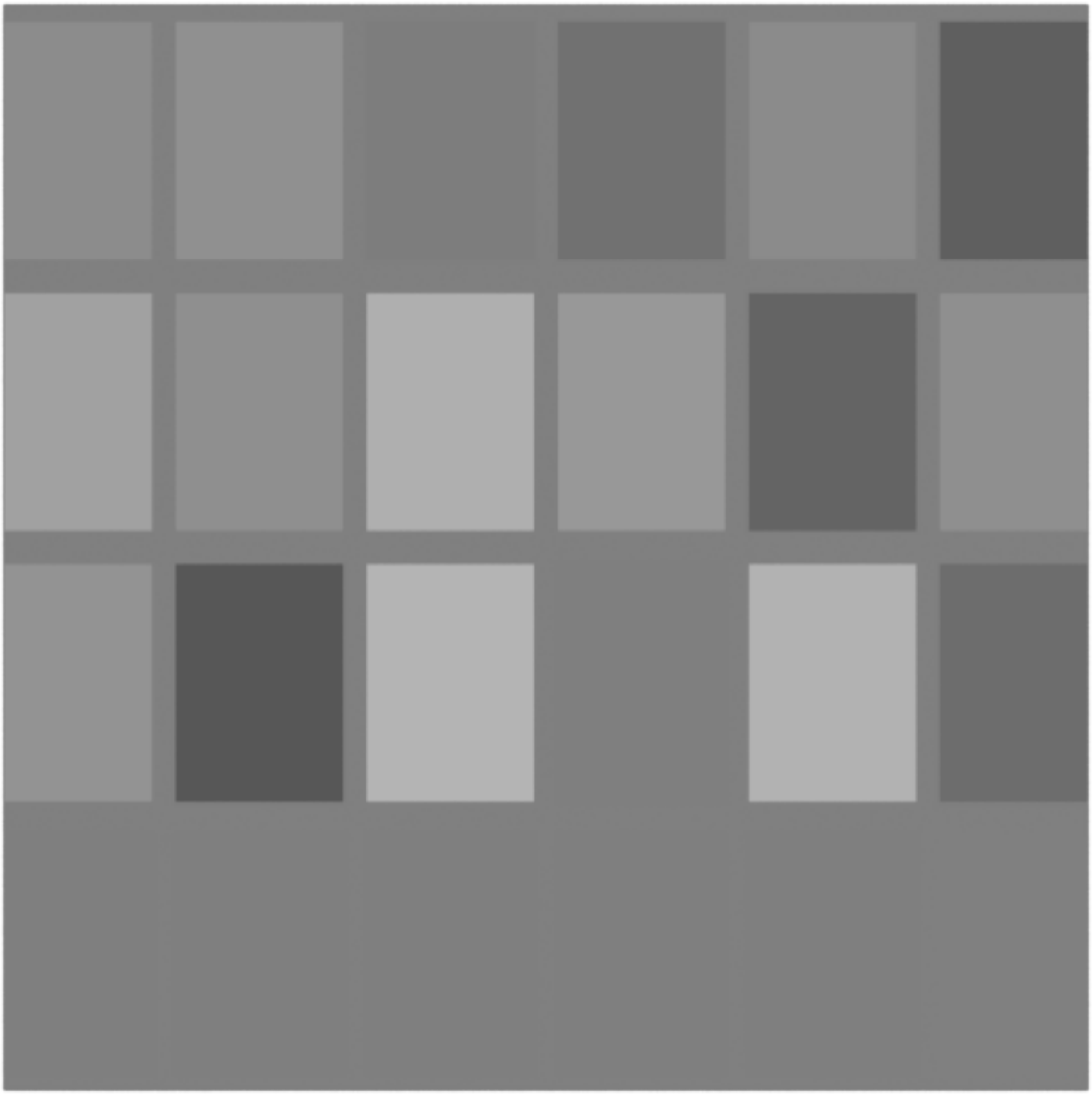


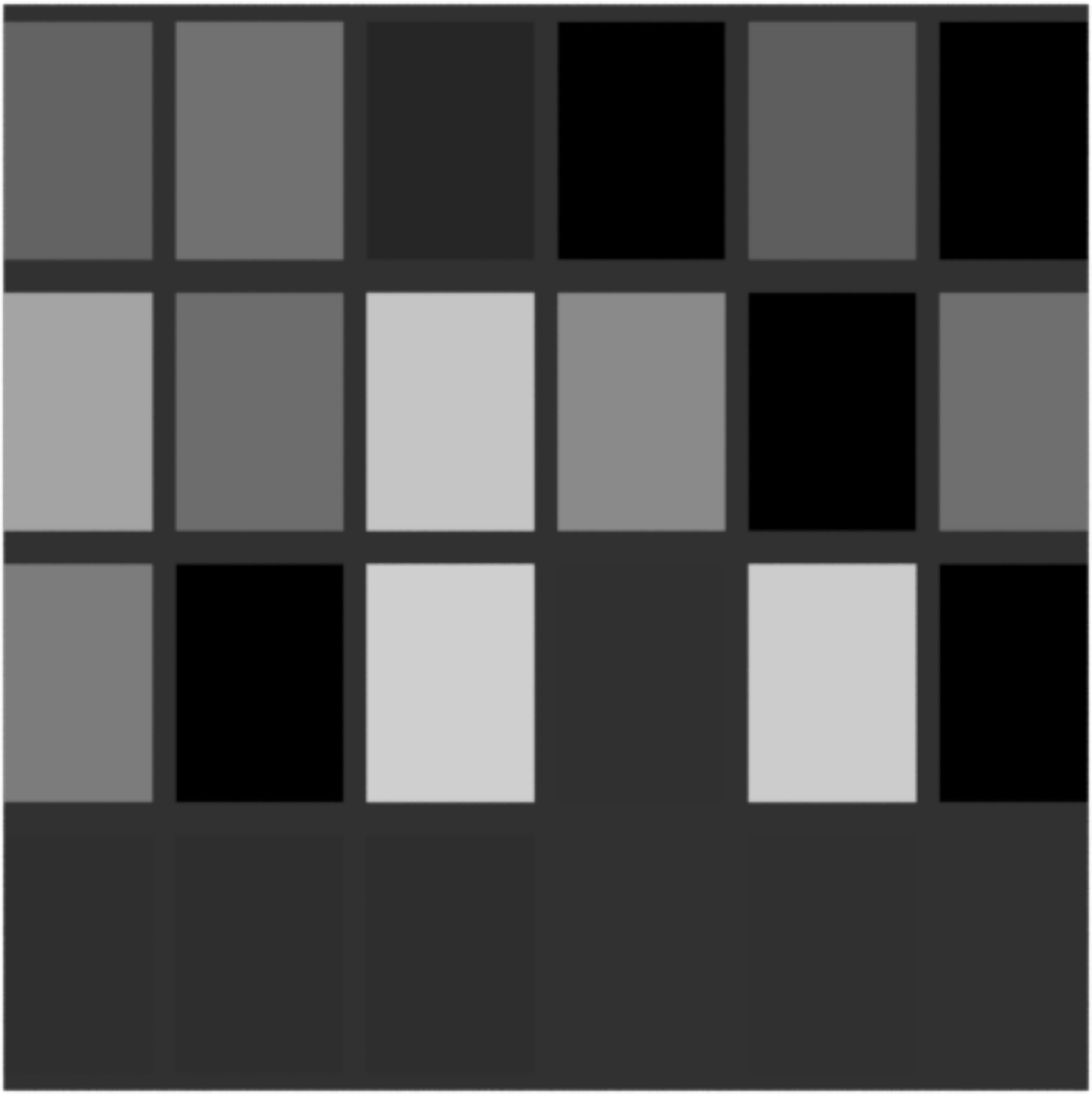












1.29 asDoubleShade

A node that allows the user to set a color for forward facing faces and for back facing faces.

1.29.1 Parameters

Color Parameters

Front Color The color or input for the front facing geometry.

Back Color The color or input for the back facing geometry.

1.29.2 Outputs

Output Color The resulting color.

1.30 asFalloffAngle

A node that takes as input two vectors, and returns a smoothly interpolated value between a user set lower and upper bounds for the angle between them.

1.30.1 Parameters

Input Parameters

Vector 1 The first input vector to consider.

Normalize Input 1 If the input vector is not unit length, enabling this checkbox will normalize it.

Vector 2 The second input vector to consider.

Normalize Input 2 If the input vector is not unit length, enabling this checkbox will normalize it.

Interpolation Parameters

Smoothstep Lower Bound The lower bound for the smooth interpolation for the angle between the two input vectors.

Smoothstep Upper Bound The upper bound for the smooth interpolation for the angle between the two input vectors.

Smoothstep Function The smooth interpolation[1] function to use, it can be one of

- *Smooth Step*
- *Smoother Step*
- *Smoothest Step*

1.30.2 Outputs

Result The smoothly interpolated value.

See also:

The wikipedia page on smoothstep for more information.

[1] Where *smootherstep* has 0 first and second derivatives, and *smootheststep* has 0 third derivatives as well.

1.31 asFresnel

A node that gives the user the reflection amount due to a viewer Fresnel term for dielectrics or conductors.

1.31.1 Parameters

Fresnel Parameters

Fresnel Type The type of Fresnel function to use. It can be one for dielectric[1] materials, or for conductors[2], with a physically based or an artist friendly *[Gul14]* parameterization. The values it can take are therefore

- Simple Dielectric
- Artist Friendly
- Physically Based

Hint: You can find physically based values for the complex index of refraction[3] in literature or in online resources such as this RefractiveIndex.Info or LuxPop.com.

Index Of Refraction The (monochromatic) absolute index of refraction for a dielectric, considered only when *Fresnel Type* is set to *Simple Dielectric*.

Facing Tint The R,G,B reflectance at normal or facing incidence for a conductor Fresnel. Considered only when the *Fresnel Type* parameter is set to *Artist Friendly*.

Edge Tint The R,G,B reflectance at edge or grazing incidence for Fresnel. Considered only when the *Fresnel Type* parameter is set to *Artist Friendly*.

Complex IOR The R,G,B index of refraction for the conductor Fresnel[4]. Considered only when in *Physically Based* mode.

Extinction Coefficient The R,G,B extinction coefficient for the conductor Fresnel[5]. Considered only when in *Physically Based* mode.

Warning: In order to be physically correct, the Fresnel term would need to provide the amount of light reflected off the surface of the object, but the object's surface might be described (and typically is) by a statistical distribution of normal vectors. Therefore, the correct Fresnel term would depend on this distribution's microfacet normal, and subsequently on the surface roughness.

This node however does **not** provide the Fresnel reflection amount off a microfacet normal, but from the true surface normal, a *viewer Fresnel term*. This is provided for creative freedom (i.e: creative blending of materials with *asBlendShader* node.

[1] See dielectric definition and Fresnel Equations for details.
[2] See conductor definition for details.
[3] Complex index of refraction, where the real part η is the index of refraction and describes the phase velocity of the wave, and the imaginary part κ is the extinction coefficient and indicates the amount of attenuation when the electro-magnetic wave propagates through the material. See Complex refractive index for more details.
[4] More precisely the real part of the complex index of refraction of the conductor.
[5] More precisely, the imaginary part of the complex index of refraction for a conductor.

Globals Parameters

Surface Normal The unit length, world space shading normal. You can use the bump normal here as well, as long as it's normalized and in *world* space.

Viewer Vector The unit length, world space vector pointing from the eye position to the surface point P being shaded.

1.31.2 Outputs

Output Color The output RGB Fresnel reflection amount.

Output Alpha The output dielectric Fresnel amount when *Fresnel Type* is set to *Simple Dielectric*, or the luminance[6] of the *Output Color* when set to the other modes.

References

1.32 asGlobals

A node that exposes the OSL[1] global variables to the user, **and** the appleseed specific global variables as well.

1.32.1 Parameters

This node has no input parameters.

[6] For now, the luminance is set to use the ITU-R BT.709/Rec.709 Y coefficients, since that is the working space appleseed is using. In the future, this will automatically reflect the choice of working or rendering space chosen by the user.

[1] OSL or Open Shading Language.

1.32.2 Outputs

Surface Position The surface point *P* in world space.

Reference Position The surface point *Pref* of a reference or pose mesh, in world space.

Light Point Position The position in the light source, *Ps*, in world space.

Shading Normal The surface normal *N* in world space, not necessarily normalized.

Reference Normal The surface normal *Nref* of a reference or pose mesh, in world space.

Geometric Normal The true geometric normal *Ng* in world space, not necessarily normalized.

Viewer Vector The vector *I* pointing from the eye position into the surface point being shaded, unit length.

Bitangent Vector The bitangent or binormal vector *Bn*, unit length.

Tangent Vector The tangent vector *Tn*, unit length.

Shutter Time The *time* global variable, denoting shutter time.

Time Amount The amount of time *dtime* covered by the shading sample.

dP/dtime Derivative of *P* in regard to *time*, or how *P* changes per unit *time*.

U Coordinate Texture coordinate *u*.

V Coordinate Texture coordinate *v*.

UV Coordinates A float array containing both the *u* and *v* coordinates, sometimes required for some applications that expect this specific data type.

dN/du The global variable *dNdu* is a appleseed specific global variable, and denotes the change of *N* in regard to *u*.

dN/dv The global variable *dNdv* is a appleseed specific global variable, and denotes the change of *N* in regard to *v*.

dP/du The global variable *dPdu*, denoting the rate of change or partial derivative of *P* in regard to *u*.

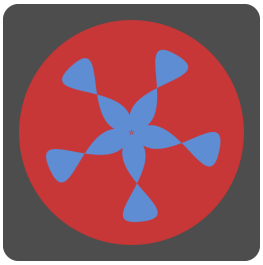
dP/dv The global variable *dPdv*, denoting the rate of change or partial derivative of *P* in regard to *v*.

dP/dx The rate of change or partial derivative of *P* in regard to *x*, or `Dx(P)`.

dP/dy The rate of change or partial derivative of *P* in regard to *y*, or `Dy(P)`.

dP/dz The rate of change or partial derivative of *P* in regard to *z*, or `Dz(P)`.

Attention: Some of the global variables exposed are specific to appleseed, hence shaders using these variables will not be entirely portable, unless the target renderers have some measure of equivalency.



1.33 asIdManifold

A utility node that generates a integer hash, a color ID, and a greyscale ID from the input geometry, based on a number of user-set criteria. Coupled with color variation or randomization nodes, this allows the user to shade effortlessly large scenes with natural looking variation.

1.33.1 Parameters

Manifold Parameters

Manifold Type The criteria to use when generating the outputs. It can create the randomization outputs based on the object, instance or assembly names or their IDs, and based on string matching via regular expressions[1]. It takes the following values

- Object Name
- Object Instance Name
- Assembly Name
- Assembly Instance Name
- Face ID
- String Prefix
- String Suffix
- Find String[2]

String Parameters

Expression The string expression to search for in the object or instance name. This string can be a regular expression.

Domain The domain of the input for the string matching with the *Expression* parameter. It can take the following values

- Object Name
- Object Instance Name
- Assembly Name
- Assembly Instance Name

Seed The seed[3] to use when generating the randomization outputs.

---

[1] Regular expressions, or regex. If you're unfamiliar with it, it allows the creation of complex patterns for string and substring matching. You can validate your expressions here at regex101.

[2] This mode tries to match the pattern anywhere in the string, while the two previous modes (*string suffix* and *string prefix*) try to match the beginning and the end of the string that was passed.

[3] A number used to initialize a pseudo random number generator, to allow some degree of determinism in a system. See random seed for more information.

Output Mode Parameters

Output Mode The type of outputs created with the node. It can take the values

- Hash Only
- Hash & Greyscale Value
- Hash, Greyscale & Color ID

1.33.2 Outputs

Output Hash The resulting integer hash.

Output ID The resulting color ID.

Output Greyscale The resulting greyscale ID.

1.34 asLuminance

A node that returns the luminance of a color, respecting the color space definitions (that is, the chromaticity coordinates of the primaries and the white point).

1.34.1 Parameters

Color Attributes

Input Color The color being evaluated

Color Space

Derive From Maya CMS Uses the render space definitions from Maya's synColor. This will set the chromaticity coordinates of the RGB primaries, and the white point, standardized in the color space chosen in the synColor configuration. When this is not set, the user **must** set the appropriate options matching its choice of rendering/working space.

Important: appleseed and appleseed-maya don't yet take OpenColorIO into account, so this parameter considers the working space definitions from synColor **only**. If you wish to use OCIO you **must** set the appropriate color space and white point settings. The default is (scene-linear) sRGB/Rec.709 primaries, with D65 whitepoint.

Input Color Space The color space chosen as render/working space. It allows the user to choose one of the following

- ACES 2065-1 AP0 (D60) *[oMPAS12]*
- ACEScg AP1 (D60) *[DFD+15]*
- Rec.2020 (D65) *[SCW14]*
- DCI-P3 (DCI) *[Ini11]*
- sRGB/Rec.709 (D65)
- Chromaticity Coordinates

Hint: When choosing *Chromaticity Coordinates*, the user **must** enter the xy chromaticity coordinates of the R,G,B primaries, and **must** choose the whitepoint, either in the form of one of the available standard illuminants, in the xy chromacity coordinates of the whitepoint, or via the correlated color temperature.

R xy Coordinates The xy chromaticity coordinates of the red primary.

G xy Coordinates The xy chromaticity coordinates of the green primary.

B xy Coordinates The xy chromaticity coordinates of the blue primary.

White Point The white point definition, which can be one of the following options:

- Standard Illuminant D50
- Standard Illuminant D55
- Standard Illuminant D60
- Standard Illuminant D65
- Standard Illuminant D75
- DCI White Point
- White Point E
- Correlated Color Temperature
- White Point Chromaticity Coordinates

Color Temperature The input color temperature value in Kelvin degrees, from 1667K to 25000K.

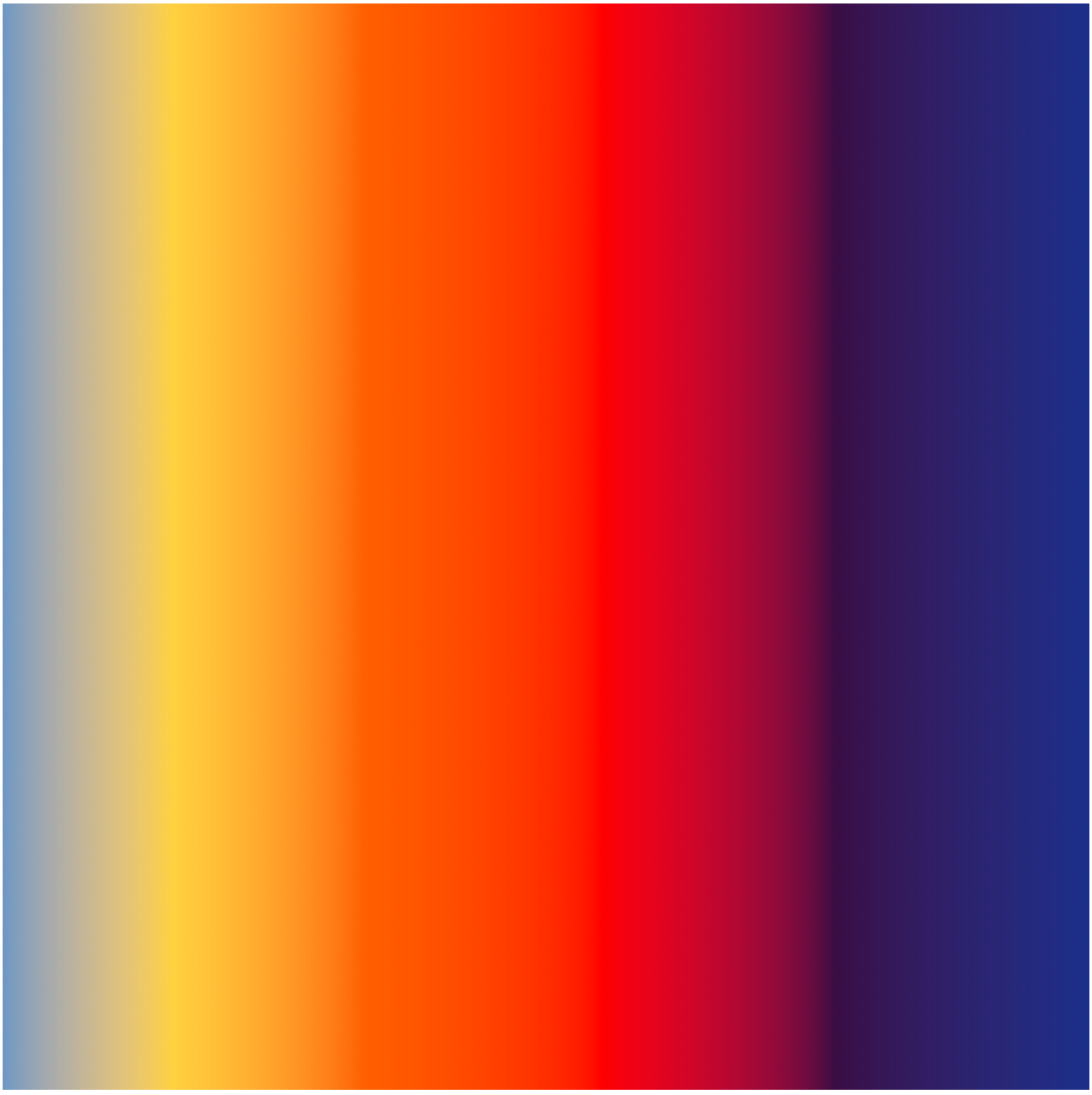
W xy Coordinates The xy chromacity coordinates of the white point.

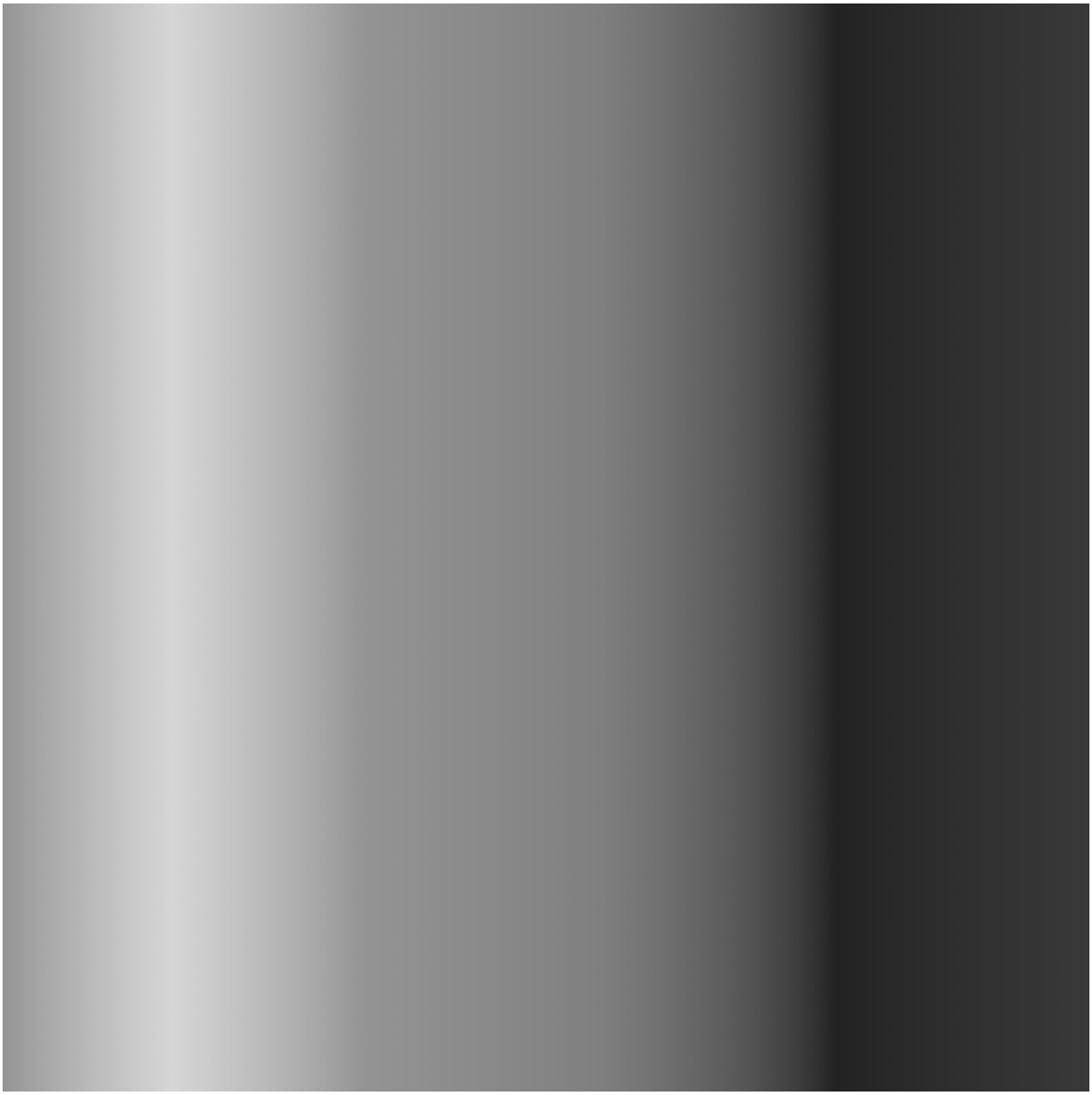
1.34.2 Outputs

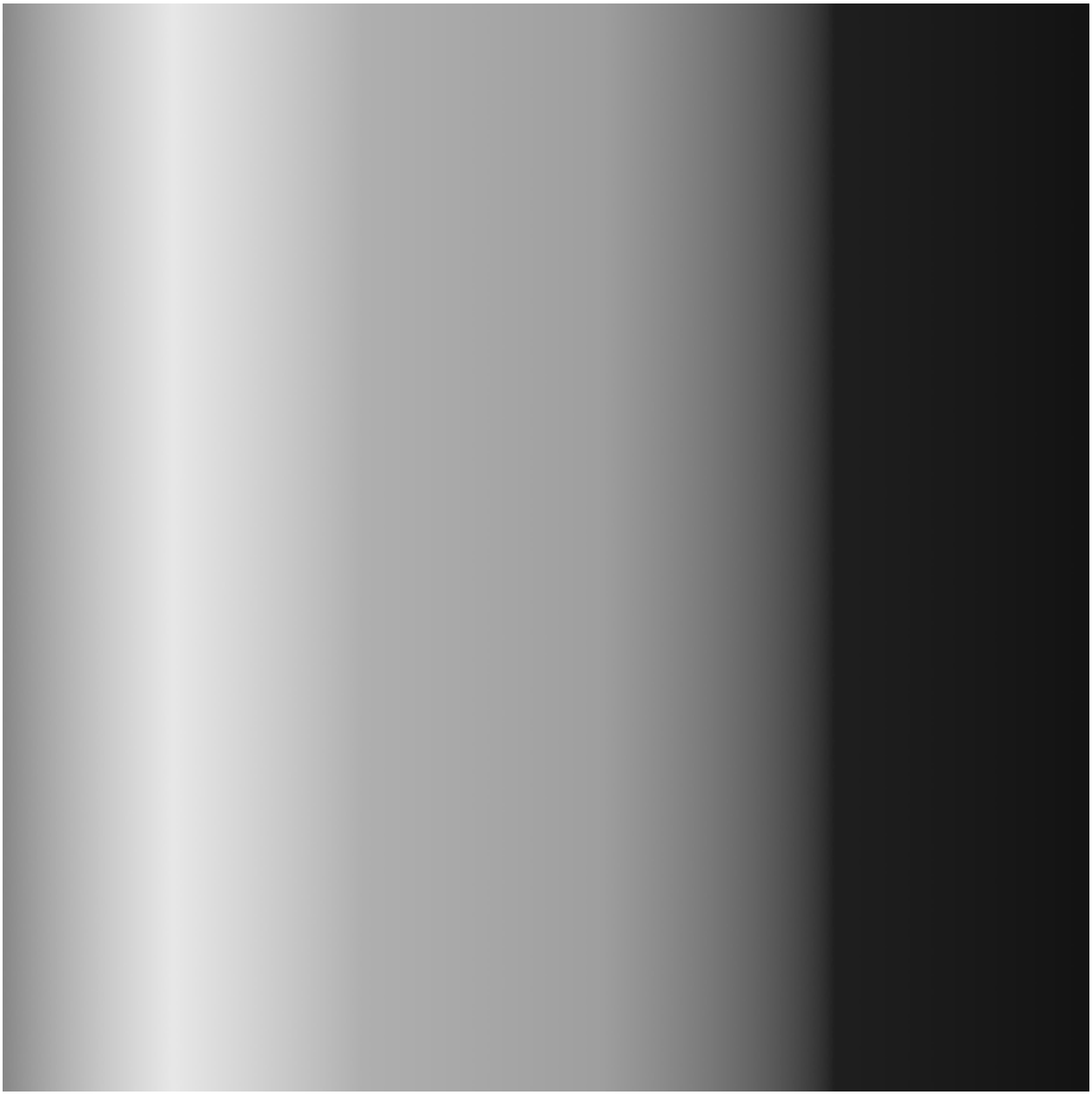
Result The luminance of the input color.

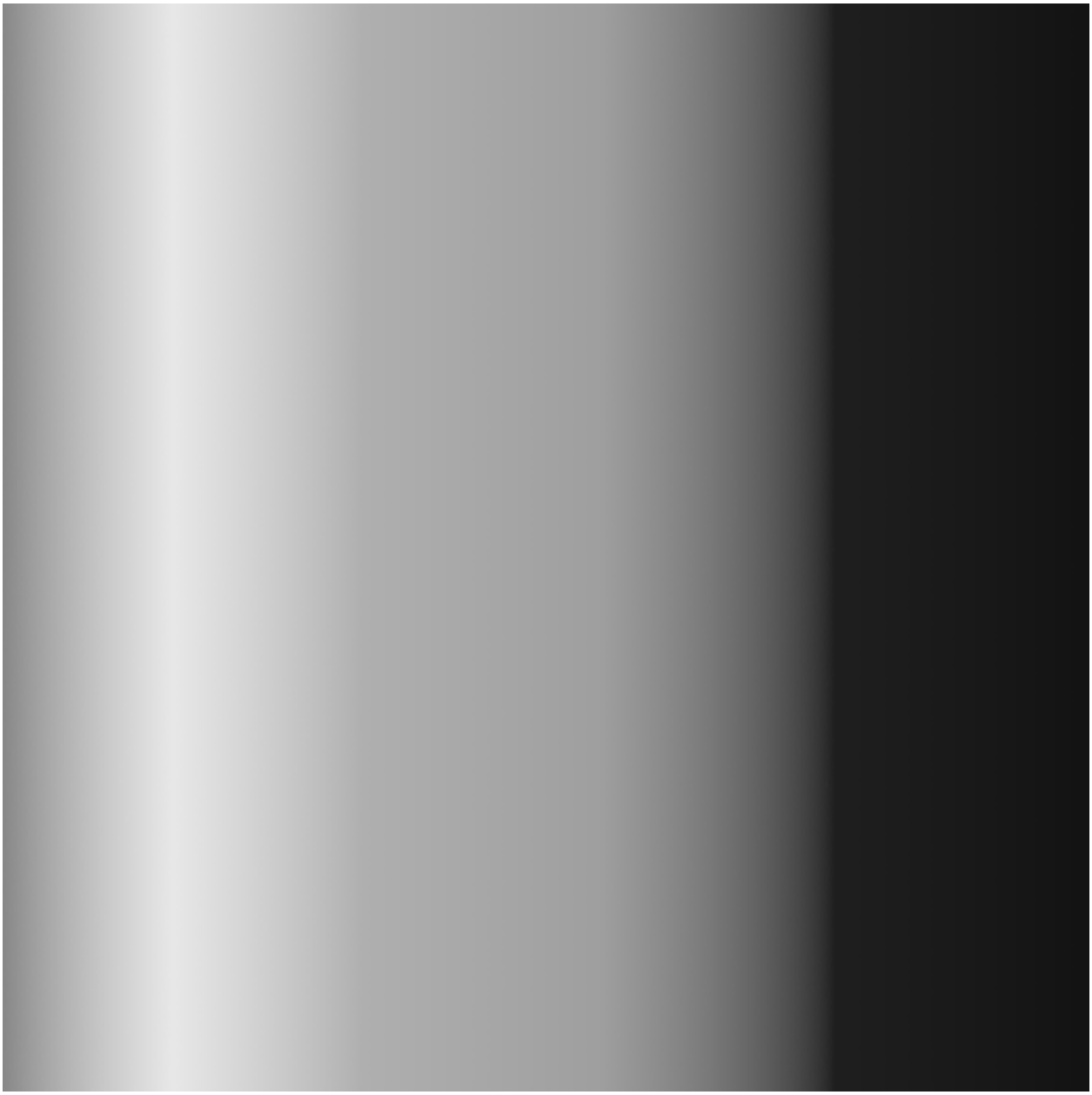
1.34.3 Screenshots

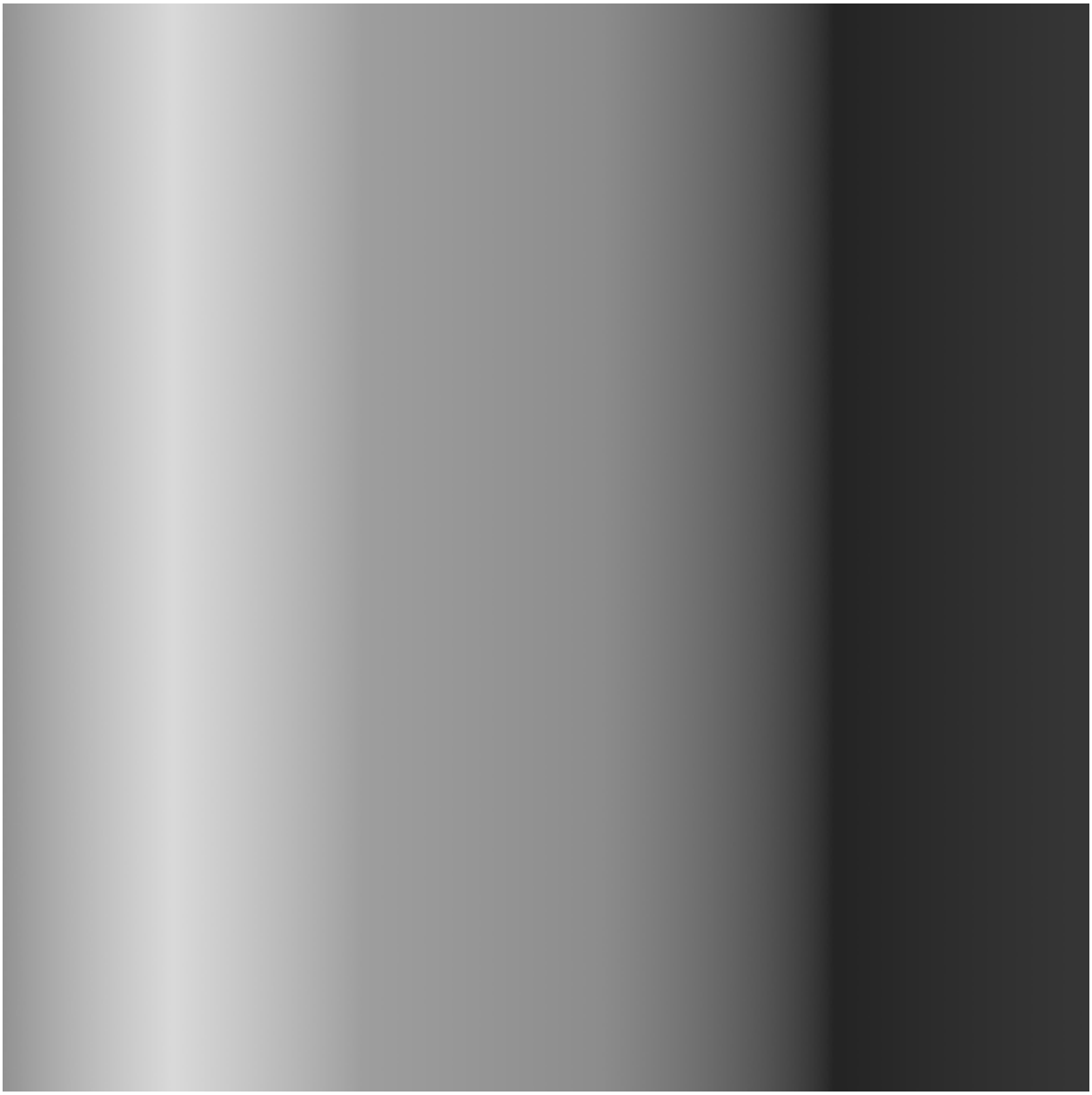
Some examples of the output luminance of the input color ramp, rendered in (scene linear) Rec.709 space, standard illuminant D65, with different color spaces and whitepoints chosen. The mismatches in color spaces are for illustration purposes. If the settings cannot be derived automatically from your DCC application, then the choice of color space should match your choice or render/working space.

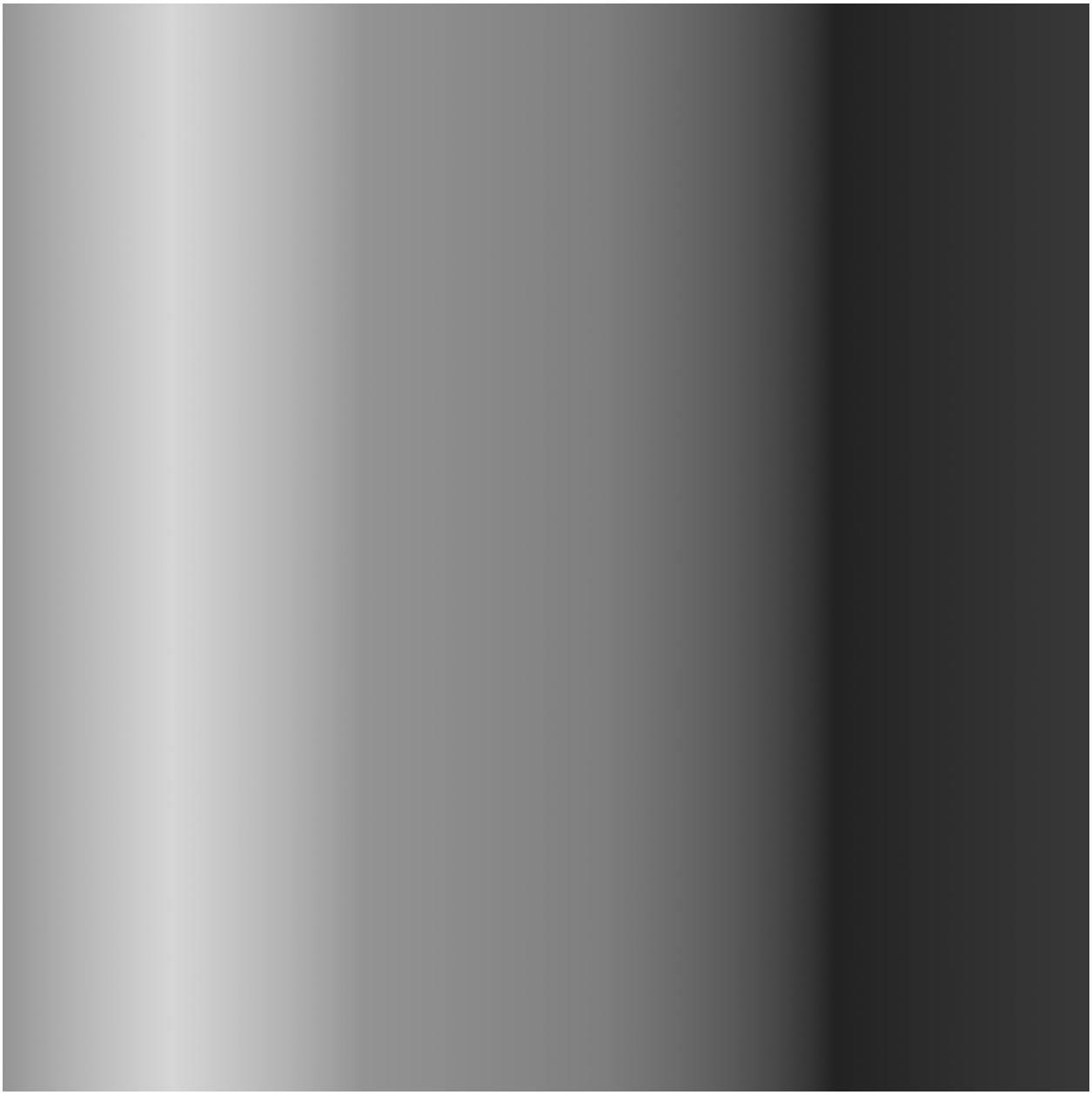


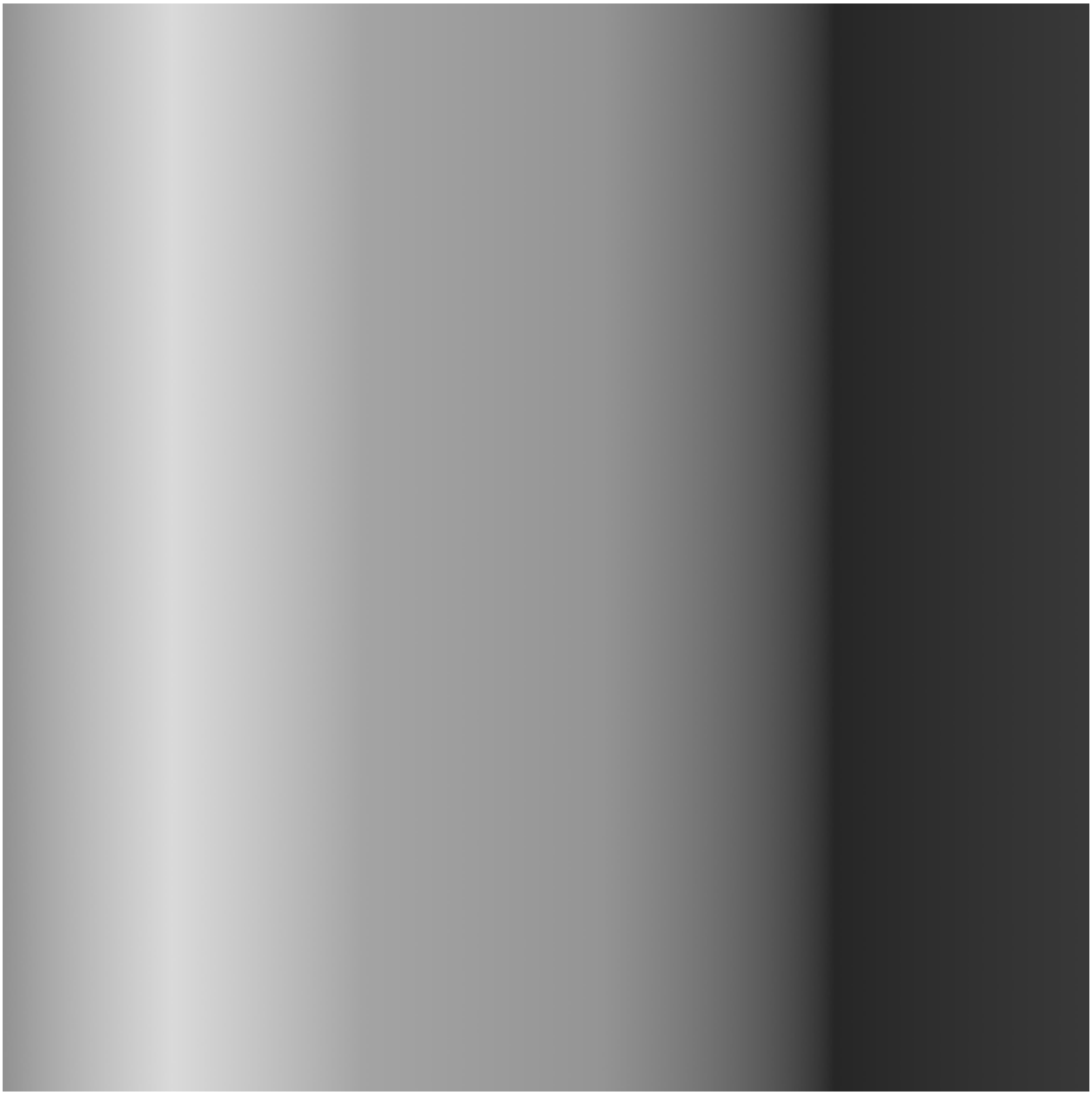


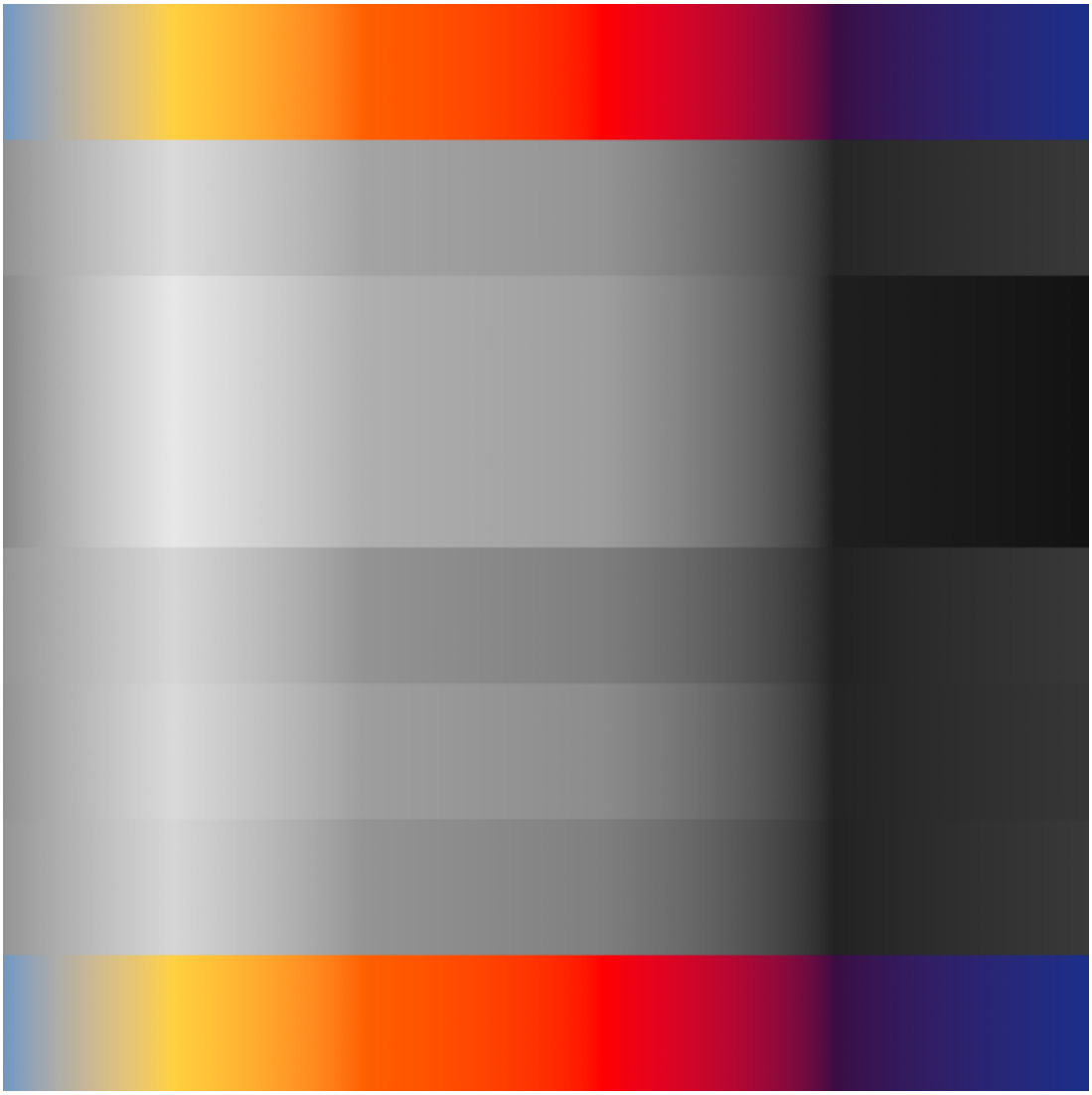




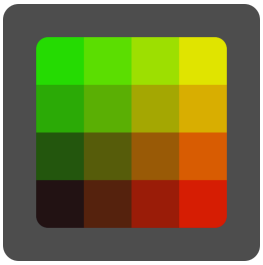








References



1.35 asManifold2D

A node that allows the user to transform UV coordinates, compute the filter sizes, define tiling, behavior outside of UV frame area, and more.

1.35.1 Parameters

UV Coordinates

UV Coordinates The input UV coordinates to transform. If not set, they default to the U and V global primitive variables.

UV Filter Size Filter widths for the input UV coordinates. If not set, they might be computed if required further below in the shader.

Bypass UV Bypass UV transformations, but still allowing you to compute the UV filter widths if required.

Compute Filters Flag that toggles computing the UV coordinates filter widths.

UV Frame

Scale UV Frame Scale the UV frame horizontally and vertically.

Translate UV Frame Translate the UV frame horizontally and vertically.

Rotate UV Frame Rotate the UV frame, where the unrestricted interval [0,1] maps to [0,360] degrees.

Frame Center Set the center of rotation X and Y coordinates.

Wrap Along U Set the U wrap mode for values outside the UV frame. These can be:

- *Periodic*: where you have repeating tiles
- *Mirror*: where you have mirrored repeating tiles
- *Clamp*: coordinates clamped to the limits of the UV frame
- *Default Color*: areas outside the UV frame take the default color set in nodes such as the *asTexture* node, in the form of its lookup failure color, or specific default color in other 2d texturing nodes.[1]

Wrap Along V Set the V wrap mode for values outside the UV frame. These can be:

- *Periodic*: where you have repeating tiles
- *Mirror*: where you have mirrored repeating tiles
- *Clamp*: coordinates clamped to the limits of the UV frame
- *Default Color*: areas outside the UV frame take the default color set in nodes such as the *asTexture* node, in the form of its lookup failure color, or specific default color in other 2d texturing nodes.

---

[1] In the case of the *asTexture node* the color used outside the [0,1] UV frame is the lookup failure color. In other nodes, you might have the access in some nodes, to set a specific *default color*, in a way similar to what Maya does for example.

Transform Parameters

UV Tiles The number of UV tiles along U and V directions.

Offset Tiles An offset to apply to the tiles, along the U and V directions.

Rotate Tiles Rotate tiles along a center of rotation, where the input values on the unrestricted interval [0,1] map to [0,360] degrees rotation.

Tiles Center Center of rotation for tiles rotation.

Tiles Coverage Defines how much of the UV frame is covered by the texture.

Tile Parameters

Mirror U Mirror the tiles along the U direction.

Mirror V Mirror the tiles along the V direction.

Wrap U Wrap the tiles along the U direction.

Wrap V Wrap the tiles along the V direction.

Stagger UV Similar to Maya's *stagger*, it offsets alternate rows of tiles by half the tile width, creating a brickwall kind of effect.

Swap UV Swap the U and V coordinates.

Noise

Distort UV Applies a signed Perlin noise to the input UV coordinates, where the texturing coordinates are the original unmodified UV coordinates scaled by a factor of 15.0 to increase the noise frequency.

1.35.2 Outputs

UV Coords The resulting UV coordinates.

UV Filter Size The filter widths for the UV coordinates, either bypassed, or final transformed UV coordinates.

References

1.36 asRaySwitch

A node that returns a specific input color according to the ray type being evaluated by the renderer *[18]*.

1.36.1 Parameters

Color Attributes

Camera Ray Color The color that passes through for primary visibility rays, also known as *camera* rays.

Light Ray Color The color that passes through *light* rays, involved in light emitting surfaces and light sources (via EDF or *emittance distribution functions*). This is also used in Bi-Directional Path Tracing[1] and in Stochastic Progressive Photon Mapping[2].

Shadow Ray Color The color that passes through for *shadow* rays, that compute the visibility between two points.

Transparency Ray Color The color that passes through for rays of type *transparency*, for matte holdouts, and when the transparency closure is evaluated.

Diffuse Ray Color The color that passes through for rays of type *diffuse*, typically involved in indirect diffuse lighting.

Glossy Ray Color The color that passes through for *glossy* rays, typically involved in glossy indirect specular lighting, such as blurry/soft reflections or refractions.

Specular Ray Color The color that passes through for *specular* rays, typically involved in the calculation of perfect mirror reflective or refractive surfaces.

Subsurface Ray Color The color that passes through for rays of type *subsurface*, typically involved in lighting calculations of BSSRDFs.

---

[1] BDPT for short, see *[38]*
[2] SPPM for short, see *[17]*

1.36.2 Outputs

Output Color The color that passed passed through for the specific ray type.

References

1.37 asSpaceTransform

A node meant to transform an input point, normal, or vector, from a coordinate system into another.

1.37.1 Parameters

Input Parameters

Point to Transform An point like variable to transform.

Normal to Transform A normal like variable to transform.

Vector to Transform A vector like variable to transform.

Space Parameters

From Space The origin coordinate system. It can be one of

- *common*[1]

[1] The *common* coordinate system is the OSL equivalent of RSL's *current* coordinate system and is the coordinate system in which all values start and the one in which lighting calculations are performed. This is a renderer specific coordinate system. In appleseed this is equivalent to the *world* coordinate system.

- *object*[2]
- *shader*[3]
- *world*[4]
- *camera*[5]
- *screen*[6]
- *raster*[7]
- *NDC*[8]

To Space The destination coordinate system. It can be one of

- *common*
- *object*
- *shader*
- *world*
- *camera*
- *screen*
- *raster*
- *NDC*

Normalize Output Vectors When the input to transform is either of *vector* type or of *normal* type, this parameter will make sure the resulting transformed *vector* or *normal* are of unit length.

See also:

The Open Shading Language documentation.

1.37.2 Outputs

Transformed Point The transformed input point.

Transformed Normal The transformed input normal, optionally normalized.

Transformed Vector The transformed input vector, optionally normalized.

Transform Matrix The transformation matrix that transforms from *From Space* to *To Space*.

---

[2] The *object* coordinate system refers to the local coordinate system of the geometry currently shaded.
[3] The *shader* coordinate system, refers to the coordinate system active by the time of the shader instantiation.
[4] The *world* coordinate system is the world coordinate system of the scene.
[5] The *camera* coordinate system refers to the coordinate system with its origin at the center of the camera lens, the *x* axis pointing right, the *y* axis pointing up, and the *z* axis pointing into the screen.
[6] The *screen* coordinate system refers to the coordinate system of the camera's image plane after the perspective transformation, if any, with the origin looking along the *z* axis of the *camera* coordinate system.
[7] The *raster* coordinate system refers to the 2D coordinate system with origin at the upper left corner of the image, and *xy* resolution at the bottom right corner of the image.
[8] The *NDC* coordinate system, or *Normalized Device Coordinates*, refers to the coordinate system with origin at the upper left corner of the image, and [1,1] at the xy coordinates of the bottom right of the image.

1.38 asSwitchTexture

A node that switches the output texture from a list of 8 inputs, based on object or instance IDs, names, or string patterns, facilitating the creation of automatic variation in large scenes.

1.38.1 Parameters

Color Parameters

Input Color 0 The first color in the list of colors or textures to evaluate.

(. . .)

Note: The subsequent inputs follow exactly the same structure and parameterization.

Input Color 7 The last color in the list of colors or textures to evaluate.

Cycle Mode When cycling through the list of colors or textures, if the index goes outside the list bounds, then one has the choice to cycle back to the beginning of the list, or if to clamp to the list size.

Manifold Parameters

Manifold Type The type of manifold to use in order to determine which lookup to do in the list of colors or textures. It can be one of the following

- Object Name
- Object Instance Name
- Assembly Name
- Assembly Instance Name
- Face ID
- String Prefix
- String Suffix

- Find String

String

Expression The expression[1] to search in the expression domain.

Domain The domain used to search an expression for.

Seed The seed[2] to use for the manifold.

1.38.2 Outputs

Output Color The resulting color.

1.39 asSwizzle

A node that rearranges the elements of an input color+alpha, or vector.

1.39.1 Parameters

Color

Input Color The color to rearrange.

Input Alpha The alpha channel for the final RGBA output.

[1] Regular expressions, or regex. If you're unfamiliar with it, it allows the creation of complex patterns for string and substring matching. You can validate your expressions here at regex101.

[2] A number used to initialize a pseudo random number generator, to allow some degree of determinism in a system. See random seed for more information.

Red Channel What RGBA channel to use as the new *Red* channel.

Green Channel What RGBA channel to use as the new *Green* channel.

Blue Channel What RGBA channel to use as the new *Blue* channel.

Alpha Channel What RGBA channel to use as the new *Alpha* channel.

Invert Red Inverts the new *Red* channel. It assumes the color was in [0,1] range.

Invert Green Inverts the new *Green* channel. It assumes the color was in [0,1] range.

Invert Blue Inverts the new *Blue* channel. It assumes the color was in [0,1] range.

Invert Alpha Inverts the new *Alpha* channel. It assumes the color was in [0,1] range.

Vector

Vector Type A vector type variable, it can be one of types

- *normal*
- *vector*
- *point*

X Component What XYZ component to use as the new *X* component.

Y Component What XYZ component to use as the new *Y* component.

Z Component What XYZ component to use as the new *Z* component.

Invert X Inverts the new *X* component.

Invert Y Inverts the new *Y* component.

Invert Z Inverts the new *Z* component.

1.39.2 Outputs

Output Color The rearranged color.

Output Alpha The rearranged alpha.

Output Vector The rearranged vector.

1.40 asTextureInfo

A node providing the user with a number of potentially useful information for the input texture. Notice that the input texture can be any texture, a 2D or 3D texture. The list of information queried might be extended to include appleseed specific metadata.

1.40.1 Parameters

Texture Parameters

Texture File The input texture file.

1.40.2 Outputs

Average Color The average color of the first 3 channels of the texture file.

Average Alpha The average value of the channel with the *A* name in the texture file, if it exists.

Resolution A vector containing the X and Y resolution in its first two components, or the X, Y and Z resolution in the case of input 3D texture file.

Channels The total number of channels in the input texture file.

Subimages The number of subimages in the texture file. In the EXR case, this refers to the number of additional channels.

Texture Type Returns the type of the texture file, with the following strings according to the input type

- *Plain Texture*
- *Shadow*
- *Environment*
- *Volume Texture*

Texture Format The texture format of the input file, differentiating between the format of the fundamental texture types. The return strings can take the following values

- *Plain Texture*
- *Shadow*
- *CubeFace Shadow*
- *Volume Shadow*
- *CubeFace Environment*
- *LatLong Environment*
- *Volume Texture*

Data Window Minimum A vector containing the minimum X and Y values of the pixel data window of the input texture, in the case of a 2D texture file. And the minimum X, Y and Z values of the pixel data window in the case of a 3D texture file.

Data Window Maximum A vector containing the maximum X and Y values of the pixel data window of the input texture, in the case of a 2D texture file. And the maximum X, Y and Z values of the pixel data window in the case of a 3D texture file.

Display Window Minimum A vector containing the minimum X and Y values of the full window of the image, in the case of a 2D texture file, and the minimum X, Y and Z values in the case of a 3D texture file.

Display Window Maximum A vector containing the maximum X and Y values of the full window of the image, in the case of a 2D texture file, and the maximum X, Y and Z values in the case of a 3D texture file.

World To Camera Matrix When the input texture is a rendered image, this parameter provides the world to camera transformation matrix used.

World To Screen Matrix When the input texture is a rendered image, this parameter provides the matrix that transforms points from *world* space into *screen* space - a 2D coordinate system where the *x* and *y* values are in [-1,1] range.

See also:

The Open Shading Language documentation at github.

1.41 asTriplanar

A triplanar projection shader, allowing individual texture(s) to be projected along the object's X, Y and Z axis, and blended. It can be used to blend colors or tangent space normal maps.

1.41.1 Parameters

Projection

Blend Mode Allowing the user to choose between blending colors, or blending tangent space normal maps[1]. It can take the values

[1] See blending in detail for details on blending tangent space normals using several methods. This node uses the *Reoriented Normal Mapping* (or RNM) blend method to blend tangent space normals when *Blend Mode* is set to *Tangent Normal*.

- Color
- Tangent Normal

Blend Softness Controls the transition softness between the projections, with higher values having a faded and very wide transition, and lower values having sharper well defined transition areas.

Surface Point The point being shaded.

Coordinate Space The coordinate system to use, it can be one of

- Object Space
- World Space

X Axis

Solid Color Solid color for the X axis projection.

Texture Filename The texture to use for the X axis projection.

Tip: Nothing prevents the user from using the same texture for all axis projections.

Horizontal Frequency The horizontal frequency of the texture for the X axis projection.

Vertical Frequency The vertical frequency of the texture for the X axis projection.

Horizontal Offset The horizontal offset of the texture for the X axis projection.

Vertical Offset The vertical offset of the texture for the X axis projection.

Rotation The rotation angle for the X axis projection, in degrees, between -360 and 360 degrees.

S Wrap Mode The texture wrapping mode along the *s* texture coordinate for the X axis projection. It can be one of

- Default
- Black
- Periodic
- Clamp
- Mirror

T Wrap Mode The texture wrapping mode along the *t* texture coordinate for the X axis projection. It can be one of

- Default
- Black
- Periodic
- Clamp
- Mirror

S Flip A flag that toggles the mirroring of the projected texture along the *s* texture coordinate.

T Flip A flag that toggles the mirroring of the projected texture along the *t* texture coordinate.

Color Management

Input Transfer Function Applies an Electro-Optical Transfer Function, or EOTF, to the input texture, linearizing it. It can take the following values

- None/Raw
- sRGB
- Rec.709
- Gamma 2.2
- Gamma 2.4
- Gamma 2.6 (DCI)
- Rec.1886[2]
- Rec.2020

RGB Primaries It allows the user to set the RGB primaries that define the color space of the input texture, and can take the following values

- Raw[3]
- sRGB/Rec.709[4]
- AdobeRGB[5]
- Rec.2020 *[SCW14]*
- DCI-P3 *[Ini11]*
- ACES *[oMPAS12]*
- ACEScg *[DFD+15]*

Y and Z axis

(. . .)

Attention: The Y and Z axis projection parameters follow exactly the same structure as the X axis projection parameters and are omitted here for brevity.

Randomization

Randomization Allows the user to add variation to the triplanar node by randomly rotating the projection axis frame, with a value of 0.0 meaning no variation at all, and a value of 1.0 allowing rotations between -360.0 and 360.0 degrees.

Manifold An integer hash, usually provided by the *asIdManifold* node. Lacking such a connection, it defaults to adding variation based on the object's assembly instance name.

[2] See ITU-R BT.1886 recommendation for details on the electro-optical transfer function.

[3] Because it makes no sense whatsoever to use colorimetry on non-color information or data, such as normal maps, or Z depth, motion vectors, and so on.

[4] sRGB shares the same CIE xy chromaticity coordinates with ITU-R BT.709/Rec.709 , hence this node refers to the RGB primaries shared by these two color spaces as *sRGB/Rec.709*.

[5] See encoding characteristics of AdobeRGB specification.

Bump Mapping

Bump Mapping The unit length bumped shading normal.

Color Management

Enable CMS Toggles the color management options *on* or *off*.

Rendering RGB Primaries It allows the user to set the RGB primaries of the rendering or working space, and it should match the choice of rendering/working space of the renderer. It can take the following values

- sRGB/Rec.709
- Rec.2020
- DCI-P3
- ACES
- ACEScg

1.41.2 Outputs

Output Color The color resulting from the *Features Mode* choice.

Output Alpha The alpha resulting from the *Features Mode* choice, usually luminance of the color only.

Output Normal The resulting blended tangent space normal maps as unit length normals in world space when the *Blend Mode* is set to *Tangent Normal*.

References

1.42 asVaryColor

A color variation utility node, meant to automate the coloring and shading of large numbers of objects in a scene based on user-set criteria such as object or instance names or IDs.

1.42.1 Parameters

Color Parameters

Input Color The original input color.

Color Mode How to apply the manifold color to the input color. The manifold generated value can be added to the input color, it can modulate the input color, or it can bypass it completely, via the parameters

- Add
- Scale
- Override

Manifold Parameters

Manifold Type The manifold type to use for the variation. It can take the following values

- Object Name
- Object Instance Name
- Assembly Name
- Assembly Instance Name
- Face ID
- String Prefix
- String Suffix
- Find String

String Parameters

Expression When *Manifold Type* is set to *String Prefix*, *String Suffix* or *Find String*, the expression can be a regex[1] expression defining the pattern to search in the expression domain.

Domain The domain to search the expression for. The domains can take the values

- Object Name
- Object Instance Name
- Assembly Name
- Assembly Instance Name

Seed An extra seed to provide some measure of determinism in the resulting colors.

[1] Regular expressions, or regex. If you're unfamiliar with it, it allows the creation of complex patterns for string and substring matching. You can validate your expressions here at regex101.

Variation Parameters

Variation Mode This parameter controls what exactly is going to be *randomized* or vary, according to the user-set options outlined earlier. One can vary the individual (and/or full) components of the input color in HSV[2] space, in RGB or in CIELAB[3] space. Accordingly this parameter takes the following values

- HSV
- RGB
- CIE L*a*b* 1976

HSV Variation Parameters

Vary Hue The extent or scaling factor of the *hue* variation.

Vary Saturation The extent or scaling factor of the *saturation* variation.

Vary Value The extent or scaling factor of the *value* variation.

RGB Variation Parameters

Vary Red The extent or scaling factor of the variation of the *red* channel.

Vary Green The extent or scaling factor of the variation of the *green* channel.

Vary Blue The extent or scaling factor of the variation of the *blue* channel.

CIELAB Variation Parameters

Vary L* The extent or scaling factor of the variation of the *lightness* or *L** channel.

Vary a* The extent or scaling factor of the variation of the *a** channel.

Vary b* The extent or scaling factor of the variation of the *b** channel.

1.42.2 Outputs

Output Color The resulting *randomized* color.

Output Hash An integer hash ID.

Output ID A color ID.

Output Greyscale A greyscale ID.

---

[2] A different color representation based on hue, saturation and value. See HSV color space for more details.

[3] Also known as *Lab* color space, but it's in fact referring to CIE 1976 L*a*b* color space, or CIELAB. See CIELAB color space for more details.

1.43 Custom appleseed shaders

These shaders are meant to either replace some of Maya's functionality that wasn't possible to implement directly in OSL, or to provide functionality that is inexistent and we feel might be useful to the end user.

1.43.1 Texture Nodes

- *asNoise2D*
- *asNoise3D*
- *asTexture*
- *asTexture3D*
- *asVoronoi2D*
- *asVoronoi3D*

1.43.2 Color Utilities

- *asBlendColor*
- *asColorTransform*
- *asCompositeColor*
- *asLuminance*
- *asSwitchTexture*
- *asVaryColor*
- *asFresnel*

1.43.3 General Utilities

- *asAnisotropyVectorField*
- *asAttributes*
- *asBump*
- *asCreateMask*
- *asDoubleShade*
- *asFalloffAngle*
- *asGlobals*
- *asIdManifold*
- *asManifold2D*
- *asRaySwitch*
- *asSpaceTransform*
- *asSwizzle*
- *asTextureInfo*

- *asTriplanar*

1.43.4 Materials

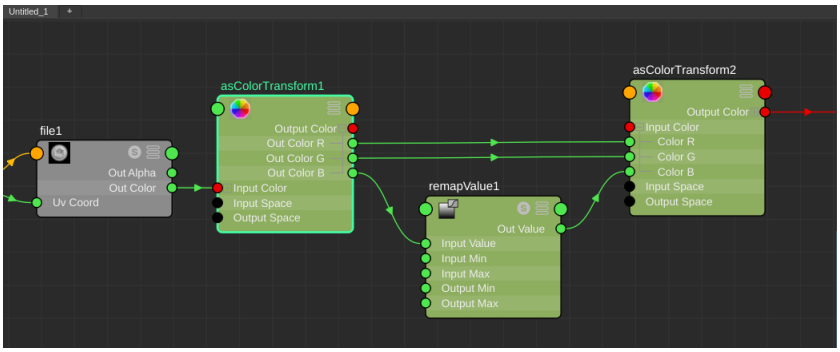
- *asBlackbody*
- *asDisneyMaterial*
- *asGlass*
- *asBlendShader*
- *asMatte*
- *asMetal*
- *asPlastic*
- *asSbsPBRMaterial*
- *asStandardSurface*
- *asSubsurface*
- *asSwitchSurface*
- *asToon*

1.44 Transforming Colors

In this example, we'll be adjusting the colors of an input texture. Ideally this would be done offline, but sometimes that's either not convenient or there is a specific need to do it online as a part of a particular effect, for instance, driven by an expression or other parameters.

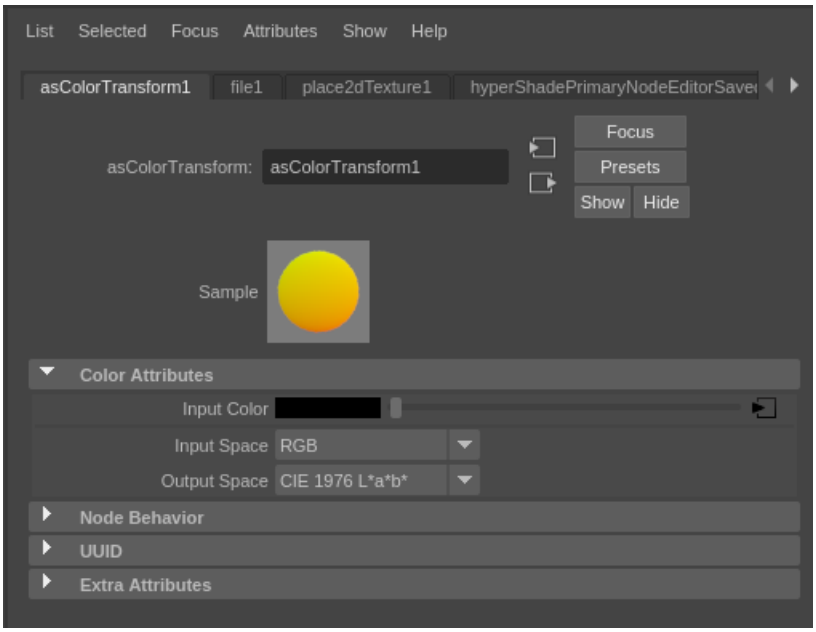
Given the example image, we want to adjust the *redness* or *greenness* of the image, without changing its lightness, or giving it an overall tint shift. We connect the output color to an *asColorTransform* node's input. The input color

is *RGB* and we will be outputting to *CIELAB*[1] color space, decomposing the output components. The first one is *lightness*, the second one is a* and the second one is b*.



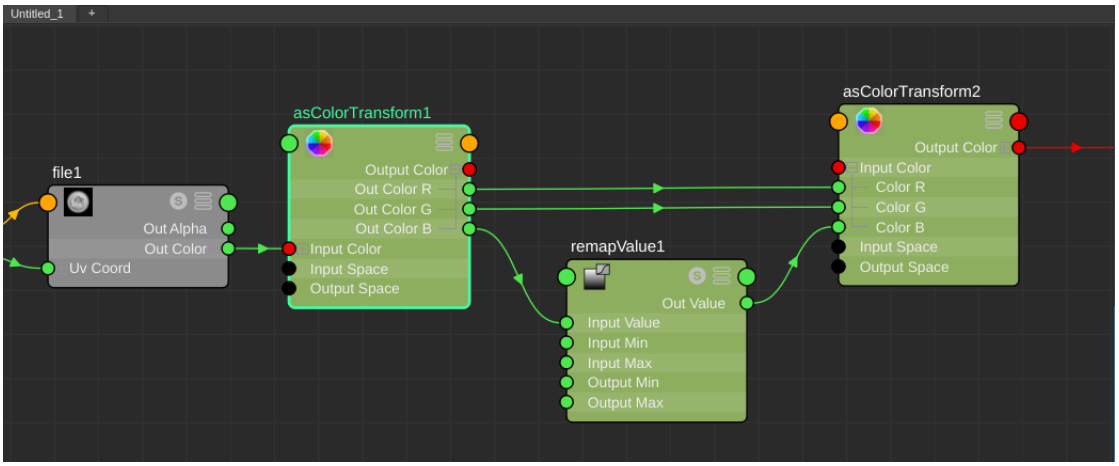
The shading network shows the texture note RGB output connected to the *asColorTransform* , a *remapValue* node changing one of the components, and the round trip back to RGB via a second *asColorTransform* node.

[1] CIELAB color space, https://en.wikipedia.org/wiki/Lab_color_space
Colorimetry - Part 4: CIE 1976 L*a*b* Colour Space pdf

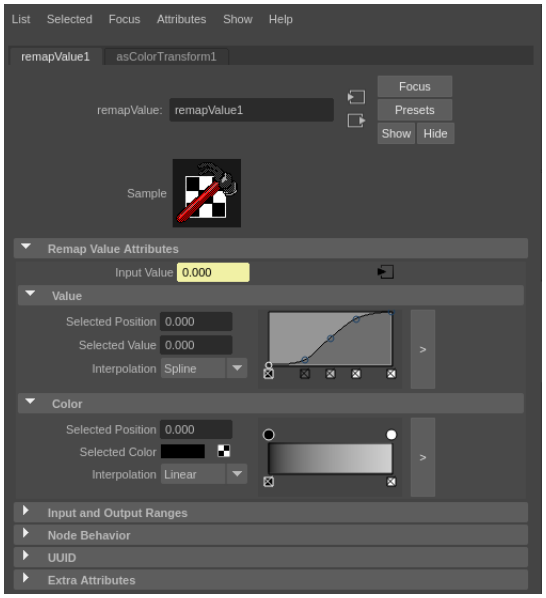


Set the input to *RGB* and output to *CIELAB* color space. The a* component is responsible for the magenta/green opposition. A value of 0.5 is grey, lower values shift towards magenta, higher values shift towards green.

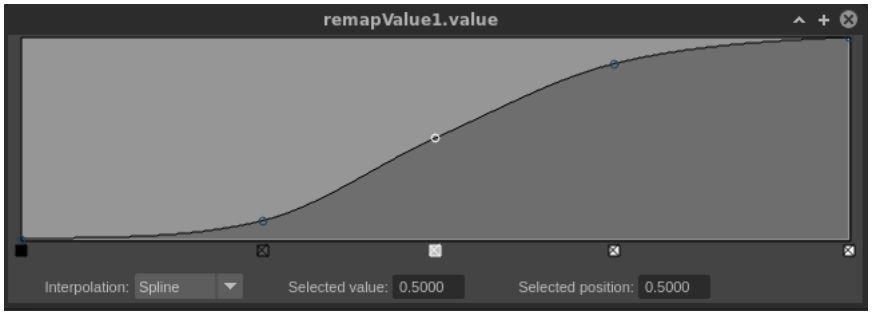
The same applies to the b* component, but this time it regard to the blue/yellow opposition instead.



Since we just want to adjust the intensity of the magentas and greens in the image, the *Lightness* and *b* are unchanged. We connect these to another *asColorTransform* node's input components, and set its input mode to *CIELAB* and its output to *RGB*, then connect the *a* component to a standard Maya *remapValue* node's input value



Leaving the position 0.5 with a value of 0.5 (greys unchanged, without a color shift), we apply a S like curve. This S like curve will increase the magentas and greens in a way similar to the S like tonal curves you are familiar with in image editing applications.



The S like curve used. Conversely, a flattening curve would bring the magent and greens towards the 0.5 values (the greys), flattening the colors.

If we wanted to do the same to the blues and yellows, we would be using the b* components of the *CIELAB* color space instead. One could also just want to increase the *blueness* of the image, or flatten the greens, or affect the lightness of the image only, leaving its colors unchanged.

1.44.1 Screenshots

1.45 Tutorials

1.45.1 Working with Custom Nodes

- *asColorTransform Tutorial*

1.46 About

appleseed-maya is an open-source, MIT-licensed multi-platform[1] renderer integration plugin for Autodesk® Maya® 2017 and later.

1.47 Bibliography

[1] Besides Linux, macOS and Windows, appleseed was known to build and run on the Raspberry Pi3, and IBM POWER8.

Bibliography

[BB17] Malik Boughida and Tamy Boubekeur. Bayesian collaborative denoising for monte carlo rendering. *Comput. Graph. Forum*, 36(4):137–153, jul 2017. URL: https://doi.org/10.1111/cgf.13231, doi:10.1111/cgf.13231.

[HDEon14] Eric Heitz and Eugene D'Eon. Importance Sampling Microfacet-Based BSDFs using the Distribution of Visible Normals. *Computer Graphics Forum*, 33(4):103–112, July 2014. URL: https://hal.inria.fr/hal-00996995, doi:10.1111/cgf.12417.

[WW11] A. Wilkie and A. Weidlich. A physically plausible model for light emission from glowing solid objects. *Computer Graphics Forum*, 30(4):1269–1276, 2011. URL: http://dx.doi.org/10.1111/j.1467-8659.2011.01986.x, doi:10.1111/j.1467-8659.2011.01986.x.

[22] Wenzel Jakob, Eugene d'Eon, Otto Jakob, and Steve Marschner. A comprehensive framework for rendering layered materials. *ACM Trans. Graph.*, 33(4):118:1–118:14, jul 2014. URL: http://doi.acm.org/10.1145/2601097.2601139, doi:10.1145/2601097.2601139.

[MHH+12] Stephen McAuley, Stephen Hill, Naty Hoffman, Yoshiharu Gotanda, Brian Smits, Brent Burley, and Adam Martinez. Practical physically-based shading in film and game production. In *ACM SIGGRAPH 2012 Courses*, SIGGRAPH '12, 10:1–10:7. New York, NY, USA, 2012. ACM. URL: http://doi.acm.org/10.1145/2343483.2343493, doi:10.1145/2343483.2343493.

[CT82] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Trans. Graph.*, 1(1):7–24, January 1982. URL: http://doi.acm.org/10.1145/357290.357293, doi:10.1145/357290.357293.

[HDEon14] Eric Heitz and Eugene D'Eon. Importance Sampling Microfacet-Based BSDFs using the Distribution of Visible Normals. *Computer Graphics Forum*, 33(4):103–112, July 2014. URL: https://hal.inria.fr/hal-00996995, doi:10.1111/cgf.12417.

[RBMS17] Mickael Ribardière, Benjamin Bringier, Daniel Meneveaux, and Lionel Simonot. STD: Student's t-Distribution of Slopes for Microfacet Based BSDFs. *Computer Graphics Forum*, 2017. doi:10.1111/cgf.13137.

[WMLT07] Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR'07, 195–206. Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association. URL: http://dx.doi.org/10.2312/EGWR/EGSR07/195-206, doi:10.2312/EGWR/EGSR07/195-206.

[CT82] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Trans. Graph.*, 1(1):7–24, January 1982. URL: http://doi.acm.org/10.1145/357290.357293, doi:10.1145/357290.357293.

[Gul14] Ole Gulbrandsen. Artist friendly metallic fresnel. *Journal of Computer Graphics Techniques (JCGT)*, 3(4):64–72, December 2014. URL: http://jcgt.org/published/0003/04/03/.

[HHdEonD16] Eric Heitz, Johannes Hanika, Eugene d'Eon, and Carsten Dachsbacher. Multiple-scattering microfacet bsdfs with the smith model. *ACM Trans. Graph.*, 35(4):58:1–58:14, jul 2016. URL: http://doi.acm.org/10.1145/2897824.2925943, doi:10.1145/2897824.2925943.

[WMLT07] Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR'07, 195–206. Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association. URL: http://dx.doi.org/10.2312/EGWR/EGSR07/195-206, doi:10.2312/EGWR/EGSR07/195-206.

[BS63] P. Beckmann and A. Spizzichino. *The scattering of electromagnetic waves from rough surfaces*. International series of monographs on electromagnetic waves. Pergamon Press; [distributed in the Western Hemisphere by Macmillan, New York], 1963. URL: https://books.google.com.my/books?id=QBEIAQAAIAAJ.

[CT82] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Trans. Graph.*, 1(1):7–24, January 1982. URL: http://doi.acm.org/10.1145/357290.357293, doi:10.1145/357290.357293.

[HHdEonD16] Eric Heitz, Johannes Hanika, Eugene d'Eon, and Carsten Dachsbacher. Multiple-scattering microfacet bsdfs with the smith model. *ACM Trans. Graph.*, 35(4):58:1–58:14, jul 2016. URL: http://doi.acm.org/10.1145/2897824.2925943, doi:10.1145/2897824.2925943.

[MHH+12] Stephen McAuley, Stephen Hill, Naty Hoffman, Yoshiharu Gotanda, Brian Smits, Brent Burley, and Adam Martinez. Practical physically-based shading in film and game production. In *ACM SIGGRAPH 2012 Courses*, SIGGRAPH '12, 10:1–10:7. New York, NY, USA, 2012. ACM. URL: http://doi.acm.org/10.1145/2343483.2343493, doi:10.1145/2343483.2343493.

[RBMS17] Mickael Ribardière, Benjamin Bringier, Daniel Meneveaux, and Lionel Simonot. STD: Student's t-Distribution of Slopes for Microfacet Based BSDFs. *Computer Graphics Forum*, 2017. doi:10.1111/cgf.13137.

[WMLT07] Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR'07, 195–206. Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association. URL: http://dx.doi.org/10.2312/EGWR/EGSR07/195-206, doi:10.2312/EGWR/EGSR07/195-206.

[SB02] Charles M. Schmidt and Brian Budge. Simple nested dielectrics in ray traced images. *Journal of Graphics Tools*, 7(2):1–8, 2002. URL: https://doi.org/10.1080/10867651.2002.10487555, arXiv:https://doi.org/10.1080/10867651.2002.10487555, doi:10.1080/10867651.2002.10487555.

[BS63] P. Beckmann and A. Spizzichino. *The scattering of electromagnetic waves from rough surfaces*. International series of monographs on electromagnetic waves. Pergamon Press; [distributed in the Western Hemisphere by Macmillan, New York], 1963. URL: https://books.google.com.my/books?id=QBEIAQAAIAAJ.

[Chr15] Per H. Christensen. An approximate reflectance profile for efficient subsurface scattering. In *ACM SIGGRAPH 2015 Talks*, SIGGRAPH '15, 25:1–25:1. New York, NY, USA, 2015. ACM. URL: http://doi.acm.org/10.1145/2775280.2792555, doi:10.1145/2775280.2792555.

[CT82] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Trans. Graph.*, 1(1):7–24, January 1982. URL: http://doi.acm.org/10.1145/357290.357293, doi:10.1145/357290.357293.

[dEonLE07] Eugene d'Eon, David Luebke, and Eric Enderton. Efficient rendering of human skin. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR'07, 147–157. Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association. URL: http://dx.doi.org/10.2312/EGWR/EGSR07/147-157, doi:10.2312/EGWR/EGSR07/147-157.

[Gul14] Ole Gulbrandsen. Artist friendly metallic fresnel. *Journal of Computer Graphics Techniques (JCGT)*, 3(4):64–72, December 2014. URL: http://jcgt.org/published/0003/04/03/.

[MHD16] Johannes Meng, Johannes Hanika, and Carsten Dachsbacher. Improving the dwivedi sampling scheme. *Comput. Graph. Forum*, 35(4):37–44, jul 2016. URL: https://doi.org/10.1111/cgf.12947, doi:10.1111/cgf.12947.

[30] Michael Oren and Shree K. Nayar. Generalization of lambert's reflectance model. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '94, 239–246. New York, NY, USA, 1994. ACM. URL: http://doi.acm.org/10.1145/192161.192213, doi:10.1145/192161.192213.

[RBMS17] Mickael Ribardière, Benjamin Bringier, Daniel Meneveaux, and Lionel Simonot. STD: Student's t-Distribution of Slopes for Microfacet Based BSDFs. *Computer Graphics Forum*, 2017. doi:10.1111/cgf.13137.

[WMLT07] Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR'07, 195–206. Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association. URL: http://dx.doi.org/10.2312/EGWR/EGSR07/195-206, doi:10.2312/EGWR/EGSR07/195-206.

[Chr15] Per H. Christensen. An approximate reflectance profile for efficient subsurface scattering. In *ACM SIGGRAPH 2015 Talks*, SIGGRAPH '15, 25:1–25:1. New York, NY, USA, 2015. ACM. URL: http://doi.acm.org/10.1145/2775280.2792555, doi:10.1145/2775280.2792555.

[DEonI11] Eugene D'Eon and Geoffrey Irving. A quantized-diffusion model for rendering translucent materials. In *ACM SIGGRAPH 2011 Papers*, SIGGRAPH '11, 56:1–56:14. New York, NY, USA, 2011. ACM. URL: http://doi.acm.org/10.1145/1964921.1964951, doi:10.1145/1964921.1964951.

[dEonLE07] Eugene d'Eon, David Luebke, and Eric Enderton. Efficient rendering of human skin. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR'07, 147–157. Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association. URL: http://dx.doi.org/10.2312/EGWR/EGSR07/147-157, doi:10.2312/EGWR/EGSR07/147-157.

[DJ05] Craig Donner and Henrik Wann Jensen. Light diffusion in multi-layered translucent materials. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, 1032–1039. New York, NY, USA, 2005. ACM. URL: http://doi.acm.org/10.1145/1186822.1073308, doi:10.1145/1186822.1073308.

[FHK14] Jeppe Revall Frisvad, Toshiya Hachisuka, and Thomas Kim Kjeldsen. Directional dipole model for subsurface scattering. *ACM Trans. Graph.*, 34(1):5:1–5:12, December 2014. URL: http://doi.acm.org/10.1145/2682629, doi:10.1145/2682629.

[HHdEonD16] Eric Heitz, Johannes Hanika, Eugene d'Eon, and Carsten Dachsbacher. Multiple-scattering microfacet bsdfs with the smith model. *ACM Trans. Graph.*, 35(4):58:1–58:14, jul 2016. URL: http://doi.acm.org/10.1145/2897824.2925943, doi:10.1145/2897824.2925943.

[JMLH01] Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan. A practical model for subsurface light transport. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, 511–518. New York, NY, USA, 2001. ACM. URL: http://doi.acm.org/10.1145/383259.383319, doi:10.1145/383259.383319.

[MHD16] Johannes Meng, Johannes Hanika, and Carsten Dachsbacher. Improving the dwivedi sampling scheme. *Comput. Graph. Forum*, 35(4):37–44, jul 2016. URL: https://doi.org/10.1111/cgf.12947, doi:10.1111/cgf.12947.

[WMLT07] Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR'07, 195–206. Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association. URL: http://dx.doi.org/10.2312/EGWR/EGSR07/195-206, doi:10.2312/EGWR/EGSR07/195-206.

[15] Amy Gooch, Bruce Gooch, Peter Shirley, and Elaine Cohen. A non-photorealistic lighting model for automatic technical illustration. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, 447–452. New York, NY, USA, 1998. ACM. URL: http://doi.acm.org/10.1145/280814.280950, doi:10.1145/280814.280950.

[GLLD12] Bruno Galerne, Ares Lagae, Sylvain Lefebvre, and George Drettakis. Gabor noise by example. *ACM Trans. Graph.*, 31(4):73:1–73:9, July 2012. URL: http://doi.acm.org/10.1145/2185520.2185569, doi:10.1145/2185520.2185569.

[LLDDutre09] Ares Lagae, Sylvain Lefebvre, George Drettakis, and Philip Dutré. Procedural noise using sparse gabor convolution. *ACM Trans. Graph.*, 28(3):54:1–54:10, jul 2009. URL: http://doi.acm.org/10.1145/1531326.1531360, doi:10.1145/1531326.1531360.

[Per85] Ken Perlin. An image synthesizer. *SIGGRAPH Comput. Graph.*, 19(3):287–296, jul 1985. URL: http://doi.acm.org/10.1145/325165.325247, doi:10.1145/325165.325247.

[Per02] Ken Perlin. Improving noise. *ACM Trans. Graph.*, 21(3):681–682, jul 2002. URL: http://doi.acm.org/10.1145/566654.566636, doi:10.1145/566654.566636.

[Wor96] Steven Worley. A cellular texture basis function. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, 291–294. New York, NY, USA, 1996. ACM. URL: http://doi.acm.org/10.1145/237170.237267, doi:10.1145/237170.237267.

[GLLD12] Bruno Galerne, Ares Lagae, Sylvain Lefebvre, and George Drettakis. Gabor noise by example. *ACM Trans. Graph.*, 31(4):73:1–73:9, July 2012. URL: http://doi.acm.org/10.1145/2185520.2185569, doi:10.1145/2185520.2185569.

[LLDDutre09] Ares Lagae, Sylvain Lefebvre, George Drettakis, and Philip Dutré. Procedural noise using sparse gabor convolution. *ACM Trans. Graph.*, 28(3):54:1–54:10, jul 2009. URL: http://doi.acm.org/10.1145/1531326.1531360, doi:10.1145/1531326.1531360.

[Per85] Ken Perlin. An image synthesizer. *SIGGRAPH Comput. Graph.*, 19(3):287–296, jul 1985. URL: http://doi.acm.org/10.1145/325165.325247, doi:10.1145/325165.325247.

[Per02] Ken Perlin. Improving noise. *ACM Trans. Graph.*, 21(3):681–682, jul 2002. URL: http://doi.acm.org/10.1145/566654.566636, doi:10.1145/566654.566636.

[Wor96] Steven Worley. A cellular texture basis function. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, 291–294. New York, NY, USA, 1996. ACM. URL: http://doi.acm.org/10.1145/237170.237267, doi:10.1145/237170.237267.

[DFD+15] Haarm-Pieter Duiker, Alexander Forsythe, Scott Dyer, Ray Feeney, Will McCown, Jim Houston, Andy Maltz, and Doug Walker. Acescg: a common color encoding for visual effects applications. In *Proceedings of the 2015 Symposium on Digital Production*, DigiPro '15, 53–53. New York, NY, USA, 2015. ACM. URL: http://doi.acm.org/10.1145/2791261.2791273, doi:10.1145/2791261.2791273.

[Ini11] Digital Cinema Initiatives. Rp 431-2:2011 - smpte recommended practice - d-cinema quality #x2014; reference projector and environment. *SMPTE RP 431-2:2011*, pages 1–14, April 2011. doi:10.5594/SMPTE.RP431-2.2011.

[oMPAS12] The Academy of Motion Picture Arts and Sciences. St 2065-1:2012 - smpte standard - academy color encoding specification (aces). *ST 2065-1:2012*, pages 1–23, April 2012. doi:10.5594/SMPTE.ST2065-1.2012.

[SCW14] M. Sugawara, S. Y. Choi, and D. Wood. Ultra-high-definition television (rec. itu-r bt.2020): a generational leap in the evolution of television [standards in a nutshell]. *IEEE Signal Processing Magazine*, 31(3):170–174, May 2014. doi:10.1109/MSP.2014.2302331.

[EMP+02] David S. Ebert, F. Kenton Musgrave, Darwyn Peachey, Ken Perlin, and Steven Worley. *Texturing and Modeling: A Procedural Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2002. ISBN 1558608486.

[Wor96] Steven Worley. A cellular texture basis function. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, 291–294. New York, NY, USA, 1996. ACM. URL: http://doi.acm.org/10.1145/237170.237267, doi:10.1145/237170.237267.

[EMP+02] David S. Ebert, F. Kenton Musgrave, Darwyn Peachey, Ken Perlin, and Steven Worley. *Texturing and Modeling: A Procedural Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2002. ISBN 1558608486.

[Wor96] Steven Worley. A cellular texture basis function. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, 291–294. New York, NY, USA, 1996. ACM. URL: http://doi.acm.org/10.1145/237170.237267, doi:10.1145/237170.237267.

[SB02] Charles M. Schmidt and Brian Budge. Simple nested dielectrics in ray traced images. *Journal of Graphics Tools*, 7(2):1–8, 2002. URL: https://doi.org/10.1080/10867651.2002.10487555, arXiv:https://doi.org/10.1080/10867651.2002.10487555, doi:10.1080/10867651.2002.10487555.

[Wal81] Bruce A. Wallace. Merging and transformation of raster images for cartoon animation. *SIGGRAPH Comput. Graph.*, 15(3):253–262, August 1981. URL: http://doi.acm.org/10.1145/965161.806813, doi:10.1145/965161.806813.

[Bli78] James F. Blinn. Simulation of wrinkled surfaces. *SIGGRAPH Comput. Graph.*, 12(3):286–292, August 1978. URL: http://doi.acm.org/10.1145/965139.507101, doi:10.1145/965139.507101.

[CMSR98] P. Cignoni, C. Montani, R. Scopigno, and C. Rocchini. A general method for preserving attribute values on simplified meshes. In *Proceedings of the Conference on Visualization '98*, VIS '98, 59–66. Los Alamitos, CA, USA, 1998. IEEE Computer Society Press. URL: http://dl.acm.org/citation.cfm?id=288216.288224.

[COM98] Jonathan Cohen, Marc Olano, and Dinesh Manocha. Appearance-preserving simplification. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, 115–122. New York, NY, USA, 1998. ACM. URL: http://doi.acm.org/10.1145/280814.280832, doi:10.1145/280814.280832.

[25] Thomas Mansencal, Michael Mauderer, Michael Parsons, Luke Canavan, Sean Cooper, Nick Shaw, Kevin Wheatley, Katherine Crowson, and Ofek Lev. Colour 0.3.11. February 2018. URL: https://doi.org/10.5281/zenodo.1175177, doi:10.5281/zenodo.1175177.

[27] K. McLAREN. Xiii—the development of the cie 1976 (l* a* b*) uniform colour space and colour-difference formula. *Journal of the Society of Dyers and Colourists*, 92(9):338–341, 1976. URL: http://dx.doi.org/10.1111/j.1478-4408.1976.tb03301.x, doi:10.1111/j.1478-4408.1976.tb03301.x.

[34] Charles Poynton. *Digital Video and HD: Algorithms and Interfaces*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2 edition, 2012. ISBN 9780123919267, 9780123919328.

[PD84] Thomas Porter and Tom Duff. Compositing digital images. In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '84, 253–259. New York, NY, USA, 1984. ACM. URL: http://doi.acm.org/10.1145/800031.808606, doi:10.1145/800031.808606.

[Wal81] Bruce A. Wallace. Merging and transformation of raster images for cartoon animation. *SIGGRAPH Comput. Graph.*, 15(3):253–262, August 1981. URL: http://doi.acm.org/10.1145/965161.806813, doi:10.1145/965161.806813.

[Gul14] Ole Gulbrandsen. Artist friendly metallic fresnel. *Journal of Computer Graphics Techniques (JCGT)*, 3(4):64–72, December 2014. URL: http://jcgt.org/published/0003/04/03/.

[DFD+15] Haarm-Pieter Duiker, Alexander Forsythe, Scott Dyer, Ray Feeney, Will McCown, Jim Houston, Andy Maltz, and Doug Walker. Acescg: a common color encoding for visual effects applications. In *Proceedings of the 2015 Symposium on Digital Production*, DigiPro '15, 53–53. New York, NY, USA, 2015. ACM. URL: http://doi.acm.org/10.1145/2791261.2791273, doi:10.1145/2791261.2791273.

[Ini11] Digital Cinema Initiatives. Rp 431-2:2011 - smpte recommended practice - d-cinema quality #x2014; reference projector and environment. *SMPTE RP 431-2:2011*, pages 1–14, April 2011. doi:10.5594/SMPTE.RP431-2.2011.

[oMPAS12] The Academy of Motion Picture Arts and Sciences. St 2065-1:2012 - smpte standard - academy color encoding specification (aces). *ST 2065-1:2012*, pages 1–23, April 2012. doi:10.5594/SMPTE.ST2065-1.2012.

[SCW14] M. Sugawara, S. Y. Choi, and D. Wood. Ultra-high-definition television (rec. itu-r bt.2020): a generational leap in the evolution of television [standards in a nutshell]. *IEEE Signal Processing Magazine*, 31(3):170–174, May 2014. doi:10.1109/MSP.2014.2302331.

[17] Toshiya Hachisuka and Henrik Wann Jensen. Stochastic progressive photon mapping. *ACM Trans. Graph.*, 28(5):141:1–141:8, December 2009. URL: http://doi.acm.org/10.1145/1618452.1618487, doi:10.1145/1618452.1618487.

[18] Paul S. Heckbert. Adaptive radiosity textures for bidirectional ray tracing. *SIGGRAPH Comput. Graph.*, 24(4):145–154, September 1990. URL: http://doi.acm.org/10.1145/97880.97895, doi:10.1145/97880.97895.

[38] Eric Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, Stanford, CA, USA, 1997. AAI9837162.

[DFD+15] Haarm-Pieter Duiker, Alexander Forsythe, Scott Dyer, Ray Feeney, Will McCown, Jim Houston, Andy Maltz, and Doug Walker. Acescg: a common color encoding for visual effects applications. In *Proceedings of the 2015 Symposium on Digital Production*, DigiPro '15, 53–53. New York, NY, USA, 2015. ACM. URL: http://doi.acm.org/10.1145/2791261.2791273, doi:10.1145/2791261.2791273.

[Ini11] Digital Cinema Initiatives. Rp 431-2:2011 - smpte recommended practice - d-cinema quality #x2014; reference projector and environment. *SMPTE RP 431-2:2011*, pages 1–14, April 2011. doi:10.5594/SMPTE.RP431-2.2011.

[oMPAS12] The Academy of Motion Picture Arts and Sciences. St 2065-1:2012 - smpte standard - academy color encoding specification (aces). *ST 2065-1:2012*, pages 1–23, April 2012. doi:10.5594/SMPTE.ST2065-1.2012.

[SCW14] M. Sugawara, S. Y. Choi, and D. Wood. Ultra-high-definition television (rec. itu-r bt.2020): a generational leap in the evolution of television [standards in a nutshell]. *IEEE Signal Processing Magazine*, 31(3):170–174, May 2014. doi:10.1109/MSP.2014.2302331.

[BS63] P. Beckmann and A. Spizzichino. *The scattering of electromagnetic waves from rough surfaces*. International series of monographs on electromagnetic waves. Pergamon Press; [distributed in the Western Hemisphere by Macmillan, New York], 1963. URL: https://books.google.com.my/books?id=QBEIAQAAIAAJ.

[Bli78] James F. Blinn. Simulation of wrinkled surfaces. *SIGGRAPH Comput. Graph.*, 12(3):286–292, August 1978. URL: http://doi.acm.org/10.1145/965139.507101, doi:10.1145/965139.507101.

[BB17] Malik Boughida and Tamy Boubekeur. Bayesian collaborative denoising for monte carlo rendering. *Comput. Graph. Forum*, 36(4):137–153, jul 2017. URL: https://doi.org/10.1111/cgf.13231, doi:10.1111/cgf.13231.

[Chr15] Per H. Christensen. An approximate reflectance profile for efficient subsurface scattering. In *ACM SIGGRAPH 2015 Talks*, SIGGRAPH '15, 25:1–25:1. New York, NY, USA, 2015. ACM. URL: http://doi.acm.org/10.1145/2775280.2792555, doi:10.1145/2775280.2792555.

[CMSR98] P. Cignoni, C. Montani, R. Scopigno, and C. Rocchini. A general method for preserving attribute values on simplified meshes. In *Proceedings of the Conference on Visualization '98*, VIS '98, 59–66. Los Alamitos, CA, USA, 1998. IEEE Computer Society Press. URL: http://dl.acm.org/citation.cfm?id=288216.288224.

[COM98] Jonathan Cohen, Marc Olano, and Dinesh Manocha. Appearance-preserving simplification. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, 115–122. New York, NY, USA, 1998. ACM. URL: http://doi.acm.org/10.1145/280814.280832, doi:10.1145/280814.280832.

[CT82] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Trans. Graph.*, 1(1):7–24, January 1982. URL: http://doi.acm.org/10.1145/357290.357293, doi:10.1145/357290.357293.

[DEonI11] Eugene D'Eon and Geoffrey Irving. A quantized-diffusion model for rendering translucent materials. In *ACM SIGGRAPH 2011 Papers*, SIGGRAPH '11, 56:1–56:14. New York, NY, USA, 2011. ACM. URL: http://doi.acm.org/10.1145/1964921.1964951, doi:10.1145/1964921.1964951.

[dEonLE07] Eugene d'Eon, David Luebke, and Eric Enderton. Efficient rendering of human skin. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR'07, 147–157. Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association. URL: http://dx.doi.org/10.2312/EGWR/EGSR07/147-157, doi:10.2312/EGWR/EGSR07/147-157.

[DJ05] Craig Donner and Henrik Wann Jensen. Light diffusion in multi-layered translucent materials. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, 1032–1039. New York, NY, USA, 2005. ACM. URL: http://doi.acm.org/10.1145/1186822.1073308, doi:10.1145/1186822.1073308.

[DFD+15] Haarm-Pieter Duiker, Alexander Forsythe, Scott Dyer, Ray Feeney, Will McCown, Jim Houston, Andy Maltz, and Doug Walker. Acescg: a common color encoding for visual effects applications. In *Proceedings of the 2015 Symposium on Digital Production*, DigiPro '15, 53–53. New York, NY, USA, 2015. ACM. URL: http://doi.acm.org/10.1145/2791261.2791273, doi:10.1145/2791261.2791273.

[EMP+02] David S. Ebert, F. Kenton Musgrave, Darwyn Peachey, Ken Perlin, and Steven Worley. *Texturing and Modeling: A Procedural Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2002. ISBN 1558608486.

[FHK14] Jeppe Revall Frisvad, Toshiya Hachisuka, and Thomas Kim Kjeldsen. Directional dipole model for subsurface scattering. *ACM Trans. Graph.*, 34(1):5:1–5:12, December 2014. URL: http://doi.acm.org/10.1145/2682629, doi:10.1145/2682629.

[GLLD12] Bruno Galerne, Ares Lagae, Sylvain Lefebvre, and George Drettakis. Gabor noise by example. *ACM Trans. Graph.*, 31(4):73:1–73:9, July 2012. URL: http://doi.acm.org/10.1145/2185520.2185569, doi:10.1145/2185520.2185569.

[15] Amy Gooch, Bruce Gooch, Peter Shirley, and Elaine Cohen. A non-photorealistic lighting model for automatic technical illustration. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, 447–452. New York, NY, USA, 1998. ACM. URL: http://doi.acm.org/10.1145/280814.280950, doi:10.1145/280814.280950.

[Gul14] Ole Gulbrandsen. Artist friendly metallic fresnel. *Journal of Computer Graphics Techniques (JCGT)*, 3(4):64–72, December 2014. URL: http://jcgt.org/published/0003/04/03/.

[17] Toshiya Hachisuka and Henrik Wann Jensen. Stochastic progressive photon mapping. *ACM Trans. Graph.*, 28(5):141:1–141:8, December 2009. URL: http://doi.acm.org/10.1145/1618452.1618487, doi:10.1145/1618452.1618487.

[18] Paul S. Heckbert. Adaptive radiosity textures for bidirectional ray tracing. *SIGGRAPH Comput. Graph.*, 24(4):145–154, September 1990. URL: http://doi.acm.org/10.1145/97880.97895, doi:10.1145/97880.97895.

[HDEon14] Eric Heitz and Eugene D'Eon. Importance Sampling Microfacet-Based BSDFs using the Distribution of Visible Normals. *Computer Graphics Forum*, 33(4):103–112, July 2014. URL: https://hal.inria.fr/hal-00996995, doi:10.1111/cgf.12417.

[HHdEonD16] Eric Heitz, Johannes Hanika, Eugene d'Eon, and Carsten Dachsbacher. Multiple-scattering microfacet bsdfs with the smith model. *ACM Trans. Graph.*, 35(4):58:1–58:14, jul 2016. URL: http://doi.acm.org/10.1145/2897824.2925943, doi:10.1145/2897824.2925943.

[Ini11] Digital Cinema Initiatives. Rp 431-2:2011 - smpte recommended practice - d-cinema quality #x2014; reference projector and environment. *SMPTE RP 431-2:2011*, pages 1–14, April 2011. doi:10.5594/SMPTE.RP431-2.2011.

[22] Wenzel Jakob, Eugene d'Eon, Otto Jakob, and Steve Marschner. A comprehensive framework for rendering layered materials. *ACM Trans. Graph.*, 33(4):118:1–118:14, jul 2014. URL: http://doi.acm.org/10.1145/2601097.2601139, doi:10.1145/2601097.2601139.

[JMLH01] Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan. A practical model for subsurface light transport. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, 511–518. New York, NY, USA, 2001. ACM. URL: http://doi.acm.org/10.1145/383259.383319, doi:10.1145/383259.383319.

[LLDDutre09] Ares Lagae, Sylvain Lefebvre, George Drettakis, and Philip Dutré. Procedural noise using sparse gabor convolution. *ACM Trans. Graph.*, 28(3):54:1–54:10, jul 2009. URL: http://doi.acm.org/10.1145/1531326.1531360, doi:10.1145/1531326.1531360.

[25] Thomas Mansencal, Michael Mauderer, Michael Parsons, Luke Canavan, Sean Cooper, Nick Shaw, Kevin Wheatley, Katherine Crowson, and Ofek Lev. Colour 0.3.11. February 2018. URL: https://doi.org/10.5281/zenodo.1175177, doi:10.5281/zenodo.1175177.

[MHH+12] Stephen McAuley, Stephen Hill, Naty Hoffman, Yoshiharu Gotanda, Brian Smits, Brent Burley, and Adam Martinez. Practical physically-based shading in film and game production. In *ACM SIGGRAPH 2012 Courses*, SIGGRAPH '12, 10:1–10:7. New York, NY, USA, 2012. ACM. URL: http://doi.acm.org/10.1145/2343483.2343493, doi:10.1145/2343483.2343493.

[27] K. McLAREN. Xiii—the development of the cie 1976 (l* a* b*) uniform colour space and colour-difference formula. *Journal of the Society of Dyers and Colourists*, 92(9):338–341, 1976. URL: http://dx.doi.org/10.1111/j.1478-4408.1976.tb03301.x, doi:10.1111/j.1478-4408.1976.tb03301.x.

[MHD16] Johannes Meng, Johannes Hanika, and Carsten Dachsbacher. Improving the dwivedi sampling scheme. *Comput. Graph. Forum*, 35(4):37–44, jul 2016. URL: https://doi.org/10.1111/cgf.12947, doi:10.1111/cgf.12947.

[oMPAS12] The Academy of Motion Picture Arts and Sciences. St 2065-1:2012 - smpte standard - academy color encoding specification (aces). *ST 2065-1:2012*, pages 1–23, April 2012. doi:10.5594/SMPTE.ST2065-1.2012.

[30] Michael Oren and Shree K. Nayar. Generalization of lambert's reflectance model. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '94, 239–246. New York, NY, USA, 1994. ACM. URL: http://doi.acm.org/10.1145/192161.192213, doi:10.1145/192161.192213.

[Per85] Ken Perlin. An image synthesizer. *SIGGRAPH Comput. Graph.*, 19(3):287–296, jul 1985. URL: http://doi.acm.org/10.1145/325165.325247, doi:10.1145/325165.325247.

[Per02] Ken Perlin. Improving noise. *ACM Trans. Graph.*, 21(3):681–682, jul 2002. URL: http://doi.acm.org/10.1145/566654.566636, doi:10.1145/566654.566636.

[PD84] Thomas Porter and Tom Duff. Compositing digital images. In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '84, 253–259. New York, NY, USA, 1984. ACM. URL: http://doi.acm.org/10.1145/800031.808606, doi:10.1145/800031.808606.

[34] Charles Poynton. *Digital Video and HD: Algorithms and Interfaces*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2 edition, 2012. ISBN 9780123919267, 9780123919328.

[RBMS17] Mickael Ribardière, Benjamin Bringier, Daniel Meneveaux, and Lionel Simonot. STD: Student's t-Distribution of Slopes for Microfacet Based BSDFs. *Computer Graphics Forum*, 2017. doi:10.1111/cgf.13137.

[SB02] Charles M. Schmidt and Brian Budge. Simple nested dielectrics in ray traced images. *Journal of Graphics Tools*, 7(2):1–8, 2002. URL: https://doi.org/10.1080/10867651.2002.10487555, arXiv:https://doi.org/10.1080/10867651.2002.10487555, doi:10.1080/10867651.2002.10487555.

[SCW14] M. Sugawara, S. Y. Choi, and D. Wood. Ultra-high-definition television (rec. itu-r bt.2020): a generational leap in the evolution of television [standards in a nutshell]. *IEEE Signal Processing Magazine*, 31(3):170–174, May 2014. doi:10.1109/MSP.2014.2302331.

[38] Eric Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, Stanford, CA, USA, 1997. AAI9837162.

[Wal81] Bruce A. Wallace. Merging and transformation of raster images for cartoon animation. *SIGGRAPH Comput. Graph.*, 15(3):253–262, August 1981. URL: http://doi.acm.org/10.1145/965161.806813, doi:10.1145/965161.806813.

[WMLT07] Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR'07, 195–206. Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association. URL: http://dx.doi.org/10.2312/EGWR/EGSR07/195-206, doi:10.2312/EGWR/EGSR07/195-206.

[WW11] A. Wilkie and A. Weidlich. A physically plausible model for light emission from glowing solid objects. *Computer Graphics Forum*, 30(4):1269–1276, 2011. URL: http://dx.doi.org/10.1111/j.1467-8659.2011.01986.x, doi:10.1111/j.1467-8659.2011.01986.x.

[Wor96] Steven Worley. A cellular texture basis function. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, 291–294. New York, NY, USA, 1996. ACM. URL: http://doi.acm.org/10.1145/237170.237267, doi:10.1145/237170.237267.

Index