10.1.16 Editors - Compositor Editor - Header - Add Menu - Texture

Table of content

Detailed table of content....1
Add menu - Texture....5
Brick Texture....5
Checker Texture....6
Gabor Texture....7
Gradient Texture....8
Magic Texture....9
Noise Texture....10
Voronoi Texture....12
Wave Texture....15
White Noise Texture....16

Detailed table of content

Detailed table of content

Detailed table of content....1
Add menu - Texture....5
Brick Texture....5
Inputs....5
Color 1, Color 2 and Mortar....5
Scale....5
Mortar Size....5
Mortar Smooth....5
Bias....5
Brick Width....5
Row Height....5
Properties....5
Offset....5
Frequency....6
Squash....6
Frequency....6
Outputs....6
Color....6
Factor....6
Checker Texture....6
Inputs....6
Vector....6
Color1, Color 2....6
Scale....6
Outputs....6
Color....6
Factor....6
Gabor Texture....7
Inputs....7

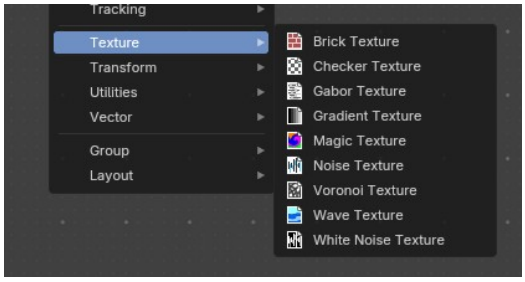
Vector 7
Scale 7
Frequency 7
Anisotropic 7
Orientation 7
Properties 7
Type 7
Outputs 7
Value 7
Phase 7
Intensity 7
Gradient Texture 8
Inputs 8
Vector 8
Properties 8
Gradient Type 8
Linear 8
Quadratic 8
Easing 8
Diagonal 8
Spherical 8
Quadratic Sphere 8
Radial 8
Outputs 8
Color 8
Factor 9
Magic Texture 9
Inputs 9
Vector 9
Scale 9
Distortion 9
Properties 9
Depth 9
Outputs 9
Color 9
Factor 9
Noise Texture 10
Inputs 10
Vector 10
Normalize 10
W 10
Scale 10
Detail 10
Roughness 10
Lacunarity 10
Offset 10
Gain 11
Distortion 11
Properties 11
Dimensions 11
1D 11
2D 11

3D....11
4D....11
Type....11
Multifractal....11
Ridged Multifractal....11
Hybrid Multifractal....11
fBM (fractal Brownian Motion)....11
Hetero Terrain (Heterogeneous Terrain)....11
Outputs....12
Factor....12
Color....12
Voronoi Texture....12
Inputs....12
Normalize....12
Vector....12
W....13
Scale....13
Detail....13
Roughness....13
Lacunarity....13
Smoothness....13
Randomness....13
Properties....13
Dimensions....13
1D....13
2D....13
3D....13
4D....13
Feature Output....13
F1....13
Smooth F1....14
Distance To Edge....14
N-Sphere Radius....14
Distance Metric....14
Euclidean....14
Manhattan....14
Chebychev....14
Minkowski....14
Outputs....14
Distance....14
Color....14
Position....14
W....14
Radius....14
Wave Texture....15
Inputs....15
Vector....15
Scale....15
Distortion....15
Detail....15
Detail Scale....15
Detail Roughness....15

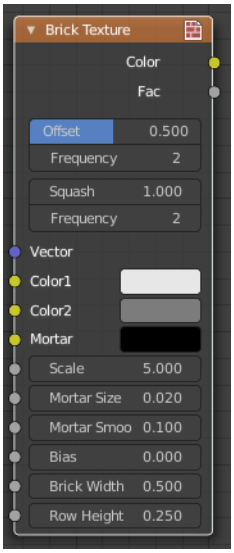
Phase Offset....15
Properties....15
Wave Type....15
Bands direction....16
Wave Profile....16
Saw....16
Sine....16
Triangle....16
Outputs....16
Color....16
Factor....16
White Noise Texture....16
Inputs....16
Vector....16
W....16
Properties....17
Dimensions....17
1D....17
2D....17
3D....17
4D....17
Outputs....17
Value....17

Add menu - Texture

Here are the texture nodes. They allow you to add different texture types to the scene.
These should be identical to the texture nodes you can access in the Shader Editor.



Brick Texture



The Brick Texture node is used to add a procedural brick texture.

Inputs

Color 1, Color 2 and Mortar

Color of the bricks and mortar.

Scale

Overall texture scale.

Mortar Size

The size of the filling between the bricks known as “mortar”; 0 means no mortar.

Mortar Smooth

Blurs/softens the edge between the mortar and the bricks. This can be useful with a texture and displacement textures.

Bias

The color variation between Color 1/2. Values of -1 and 1 only use one of the two colors; values in between mix the colors.

Brick Width

The width of the bricks.

Row Height

The height of the brick rows.

Properties

Offset

Determines the brick offset of the various rows.

Frequency

Determines the offset frequency. A value of 2 gives an even/uneven pattern of rows.

Squash

Amount of brick squashing.

Frequency

Brick squashing frequency.

Outputs

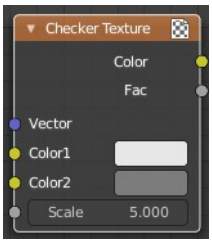
Color

Texture color output.

Factor

Mortar mask (1 = mortar).

Checker Texture



The Checker Texture node adds a procedural checkerboard texture.

Inputs

Vector

Texture coordinate to sample texture at; defaults to Generated texture coordinates if the socket is left unconnected.

Color1, Color 2

Color of the checkers.

Scale

Overall texture scale. The scale is a factor of the bounding box of the face divided by the scale. For example, a scale of 15 will result in 15 alternate patterns over the overall UV bounding box. Different patterns could be achieved using other nodes to give different input patterns to this socket. For example, using the Math Node.

Outputs

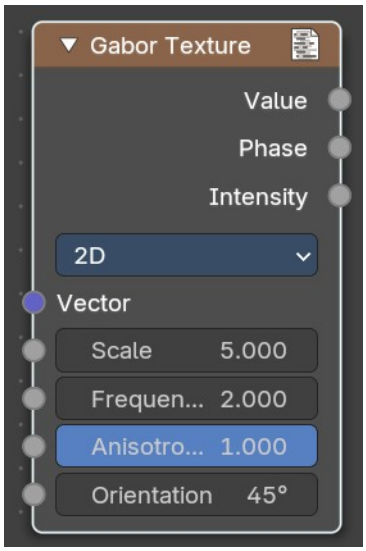
Color

Texture color output.

Factor

Checker 1 mask (1 = Checker 1).

Gabor Texture



The Gabor Texture is a special directional noise texture from the OSL Shading language.

Inputs

Vector

Texture coordinate to sample texture at defaults to Generated texture coordinates if the socket is left unconnected.

Scale

The scale of the Gabor noise.

Frequency

The rate at which the Gabor noise changes across space. It differs from the input scale in that it only scales perpendicular to the gabor noise direction.

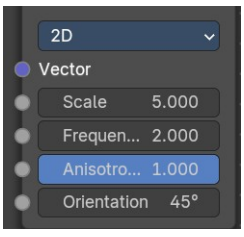
Anisotropic

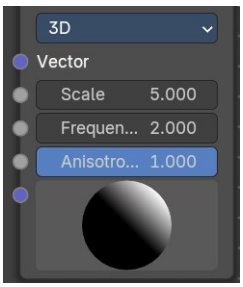
The directionality of the Gabor noise. 1 means the noise is completely directional. With a value of 0 the noise goes in all directions.

Orientation

The direction of the anisotropic Gabor noise. With the 2ds method you have a orientation value. With the 3d method you will have a trackball to adjust the orientation.

Properties





Type

2D ignores the Z input. 3D goes in all 3 directions.

Outputs

Value

The Gabor noise value with both, random phase and random intensity.

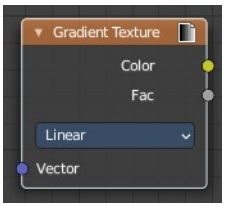
Phase

The phase of the Gabor noise, without random intensity.

Intensity

The intensity of the Gabor noise, withbout random phase.

Gradient Texture



The Gradient Texture node generates interpolated color and intensity values based on the input vector.

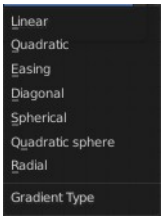
Inputs

Vector

Texture coordinate to sample texture at; defaults to Generated texture coordinates if the socket is left unconnected.

Properties

Gradient Type



Controls the type of gradient generated.

Linear

Directly returns the input X coordinate.

Quadratic

Interpolates the input X coordinate quadratically.

Easing

Uses a combination of quadratic and linear interpolation to return a smooth gradient from the input X coordinate.

Diagonal

Averages the input X and Y coordinates.

Spherical

Creates an inverse gradient using the length of the input vector; the maximum value is at (0, 0, 0).

Quadratic Sphere

The same as Spherical, except interpolated quadratically.

Radial

Returns a value based on the angle of the input around the Z axis.

Outputs

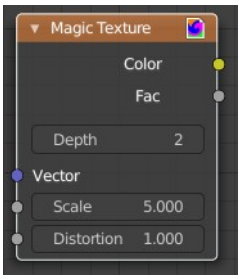
Color

Texture color output.

Factor

Texture intensity output.

Magic Texture



The Magic Texture node is used to add a procedural psychedelic color texture.

Inputs

Vector

Texture coordinate to sample texture at; defaults to Generated texture coordinates if the socket is left unconnected.

Scale

Scale of the texture.

Distortion

Amount of distortion.

Properties

Depth

Number of iterations.

Outputs

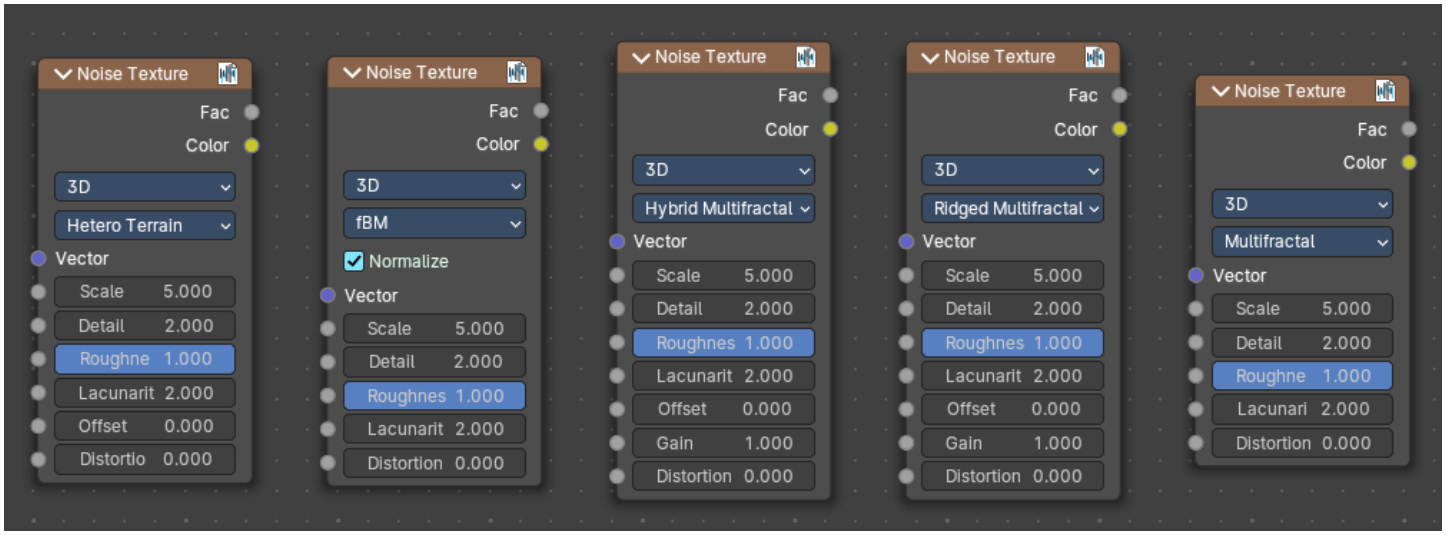
Color

Texture color output.

Factor

Texture intensity output.

Noise Texture



The Noise Texture node evaluates a fractal Perlin noise at the input texture coordinates. This nodes allows great control over how noise octaves are combined.

Inputs

The inputs are dynamic, they become available if needed depending on the node properties.

Vector

Texture coordinate to evaluate the noise at; defaults to Generated texture coordinates if the socket is left unconnected.

Normalize

Normalize the output to the 0 - 1 range.

W

Texture coordinate to evaluate the noise at. Appears with 4 dimensions.

Scale

Scale of the base noise octave.

Detail

Number of noise octaves. The fractional part of the input is multiplied by the magnitude of the highest octave. Higher number of octaves corresponds to a higher render time.

Roughness

Adds a roughness noise.

Lacunarity

The scale of a perlin noise octave relative to the perlin noise octave from the previous octave.

Offset

An added offset to each octave, determines the level where the highest octave will appear.

Gain
An extra multiplier to tune the magnitude of octaves.

Distortion
Amount of distortion.

Properties

Dimensions

The dimensions of the space to evaluate the noise in.

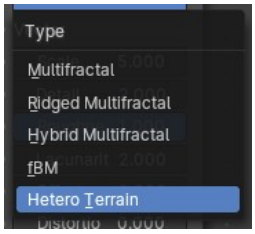
1D
Evaluate the noise in 1D space at the input W.

2D
Evaluate the noise in 2D space at the input Vector. The Z component is ignored.

3D
Evaluate the noise in 3D space at the input Vector.

4D
Evaluate the noise in 4D space at the input Vector and the input W as the fourth dimension.

Type



Type of the perlin noise texture.

Multifractal
The result is more uneven (varies with location), more similar to a real terrain. Uses a multiplicative cascade.

Ridged Multifractal
Creates sharp peaks. Calculates the absolute value of the noise, creating “canyons”, and then flips the surface upside down.

Hybrid Multifractal
Creates peaks and valleys with different roughness values, like real mountains rise out of flat plains. Combines the additive cascade with a multiplicative cascade.

fBM (fractal Brownian Motion)
Produces an unnatural homogeneous and isotropic result. Uses an additive cascade, the values are simply added together.

Hetero Terrain (Heterogeneous Terrain)
Similar to Hybrid Multifractal creates a heterogeneous terrain, but with the likeness of river channels.

Outputs

Factor

Value of fractal noise.

Color

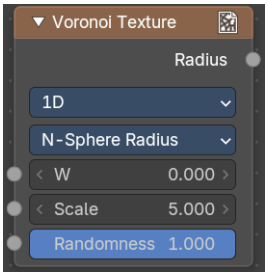
Color with different fractal noise in each component.

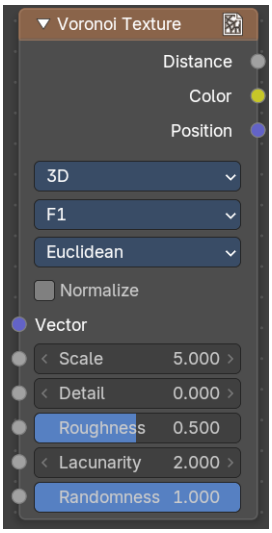
Voronoi Texture

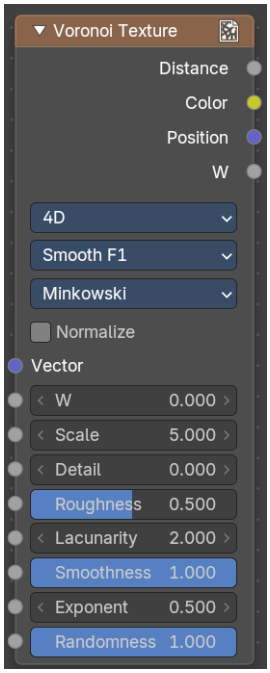
The Voronoi Texture node evaluates a Worley Noise at the input texture coordinates.

Inputs

The inputs are dynamic, they become available if needed depending on the node properties and how you set up dimensions, feature output and distance metric. It would be too much to list all possible combinations. Let’s go through the inputs one by one.







Normalize

Normalize output distance to the 0.0 to 1.0 range.

Vector

Texture coordinate to evaluate the noise at; defaults to Generated texture coordinates if the socket is left unconnected.

W

Texture coordinate to evaluate the noise at.

Scale

Scale of the noise.

Detail

The number of voronoi layers to work with.

Roughness

The influence of the previous voronoi layer to the current layer.

Lacunarity

The scale of a voronoi layer relative to the previous layer.

Smoothness

The overall smoothness.

Randomness

The randomness of the noise.

Properties

Dimensions

The dimensions of the space to evaluate the noise in.

1D

Evaluate the noise in 1D space at the input W.

2D

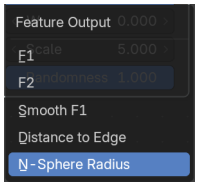
Evaluate the noise in 2D space at the input Vector. The Z component is ignored.

3D

Evaluate the noise in 3D space at the input Vector.

4D

Evaluate the noise in 4D space at the input Vector and the input W as the fourth dimension.



Feature Output

The Voronoi feature that the node will compute and return.

F1

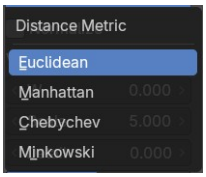
Compute and return the distance to the closest feature point as well as its position and color.

Smooth F1
Compute and return a smooth version of F1.

Distance To Edge
Compute and return the distance to the edges of the Voronoi cells.

N-Sphere Radius
Compute and return the radius of the n-sphere inscribed in the Voronoi cells. In other words, it is half the distance between the closest feature point and the feature point closest to it.

Distance Metric



The distance metric used to compute the texture.

Euclidean
Use the Euclidean distance metric.

Manhattan
Use the Manhattan distance metric.

Chebychev
Use the Chebychev distance metric.

Minkowski
Use the Minkowski distance metric. The Minkowski distance is a generalization of the aforementioned metrics with an Exponent as a parameter. Minkowski with an exponent of one is equivalent to the Manhattan distance metric. Minkowski with an exponent of two is equivalent to the Euclidean distance metric. Minkowski with an infinite exponent is equivalent to the Chebychev distance metric.

Outputs

Distance
The Distance.

Color
Cell color. The color is arbitrary.

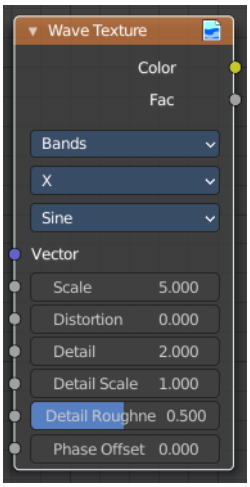
Position
Position of feature point.

W
Position of feature point.

Radius
N-Sphere radius.

Note. In some configurations of the node, especially for low values of Randomness, rendering artifacts may occur. This happens due to the same reasons described in the Notes section in the White Noise Texture page and can be fixed in a similar manner as described there.

Wave Texture



The Wave Texture node adds procedural bands or rings with noise distortion.

Inputs

Vector

Texture coordinate to sample texture at; defaults to Generated texture coordinates if the socket is left unconnected.

Scale

Overall texture scale.

Distortion

Amount of distortion of the wave (similar to the Marble texture in Blender Internal).

Detail

Amount of distortion noise detail.

Detail Scale

Scale of distortion noise.

Detail Roughness

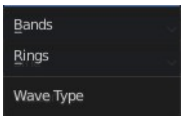
Adds a roughness noise.

Phase Offset

Set an offset for the phase.

Properties

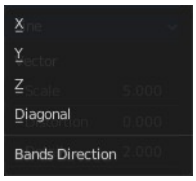
Wave Type



Bands or Rings shaped waves.

Bands direction

In which direction the bands should point.



Wave Profile

Controls the shape and look of the wave type.

Saw

Uses a saw tooth profile.

Sine

Uses the standard sine profile.

Triangle

Uses a triangle shape.

Outputs

Color

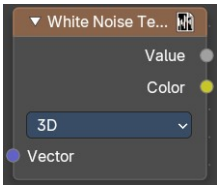
Texture color output.

Factor

Texture intensity output.

White Noise Texture

This node adds noise.



Inputs

The inputs are dynamic, they become available if needed depending on the node properties.

Vector

Vector used as seed in 2D, 3D, and 4D dimensions.

W

Value used as seed in 1D and 4D dimensions.

Properties

Dimensions

The dimensions of the space to evaluate the noise in.

1D

The W input is used as seed.

2D

The X and Y components of the Vector input are used as seed.

3D

The Vector input is used as seed.

4D

Both the Vector input and the W input are used as seed.

Outputs

Value

Output random value.

Note! The slightest difference in seed values would result in completely different outputs. Consequently, bad precision may have significant impact on the output. Usually, we can mitigate this issue by:

Eliminating the problematic seed value. If the problematic seed value is constant, it should be eliminated by choosing a lower dimension or multiplying it by zero.

Adding an arbitrary value to the seed. The issue might only happen at certain boundaries, like unit boundaries, so simply adding an arbitrary value might solve the issue.

Taking the absolute value of the seed. In computing, zero may be positive or negative, so taking the absolute values unifies the zero into a single value.