10.3.4.6 Render - Cycles Render Engine - Nodes - More Nodes

More Nodes

Value

Input a scalar value.

Value
Value output.

RGB

Input an RGB color.

Color
RGB color output.

Geometry

Geometric information about the current shading point. All vector coordinates are in *World Space*. For volume shaders, only the position and incoming vector are available.

Position
Position of the shading point.
Normal

Shading normal at the surface (includes smooth normals and bump mapping).

Tangent
Tangent at the surface.

True Normal
Geometry or flat normal of the surface.

Incoming
Vector pointing towards the point the shading point is being viewed from.

Parametric
Parametric coordinates of the shading point on the surface.

Backfacing
1.0 if the face is being viewed from the backside, 0.0 for the frontside.

Wireframe

Node for a wireframe shader (Triangles only for now).

Pixel Size
Use screen pixel size instead of world units.

Size
Controls the thickness of the wireframe.

Fac output
1.0 if shading is executed on an edge, 0.0 otherwise.

Wavelength

A wavelength to rgb converter.

Wavelength
The color wavelength from 380 to 780 nanometers.

Color
RGB color output.

Blackbody

A blackbody temperature to RGB converter.

Temperature
The temperature in Kelvin.

Color
RGB color output.

Texture Coordinates

Commonly used texture coordinates, typically used as inputs for the *Vector* input for texture nodes.

Generated
Automatically-generated texture coordinates from the vertex positions of the mesh without deformation, keeping them sticking to the surface under animation. Range from 0.0 to 1. 0 over the bounding box of the undeformed mesh.

Normal
Object space normal, for texturing objects with the texture staying fixed on the object as it transformed.

UV

UV texture coordinates from the active render UV layer.

Object
Position coordinate in object space.

Camera
Position coordinate in camera space.

Window
Location of shading point on the screen, ranging from 0.0 to 1. 0 from the left to right side and bottom to top of the render.

Reflection
Vector in the direction of a sharp reflection, typically used for environment maps.

Bump

Generate a perturbed normal from a height texture, for bump mapping. The height value will be sampled at the shading point and two nearby points on the surface to determine the local direction of the normal.

Invert
Invert the bump mapping, to displace into the surface instead of out.

Strength Input
Strength of the bump mapping effect, interpolating between no bump mapping and full bump mapping.

Distance Input
Multiplier for the height value to control the overall distance for bump mapping.

Height Input
Scalar value giving the height offset from the surface at the shading point; this is where you plug in textures.

Vector Transform

Allows converting a Vector, Point or Normal between World <=> Camera <=> Object coordinate space.

Type
Specifies the input/output type: Vector, Point or Normal.

Convert From
Coordinate Space to convert from: World, Object or Camera.

Convert To
Coordinate Space to convert to: World, Object or Camera.

Vector Input
The input vector.

Vector Output
The transformed output vector.

Tangent

Generate a tangent direction for the Anisotropic BSDF.

Direction Type
The tangent direction can be derived from a cylindrical projection around the X, Y or Z axis (Radial), or from a manually created UV Map for full control.

Tangent Output
The tangent direction vector.

Normal Map

Generate a perturbed normal from an RGB normal map image. This is usually chained with an Image Texture node in the color input, to specify the normal map image. For tangent space normal maps, the UV coordinates for the image must match, and the image texture should be set to Non-Color mode to give correct results.

Space
: The input RGB color can be in one of 3 spaces: Tangent, Object and World space. Tangent space normal maps are the most common, as they support object transformation and mesh deformations. Object space normal maps keep sticking to the surface under object transformations, while World normal maps do not.

UV Map
: Name of the UV map to derive normal mapping tangents from. When chained with an Image Texture node, this UV map should be the same as the UV map used to map the texture.

Strength
: Strength of the normal mapping effect.

Color Input
: RGB color that encodes the normal in the specified space.

Normal Output
: Normal that can be used as an input to BSDF nodes.

Object Info

Information about the object instance. This can be useful to give some variation to a single material assigned to multiple instances, either manually controlled through the object index, based on the object location, or randomized for each instance. For example a Noise texture can give random colors or a Color ramp can give a range of colors to be randomly picked from.

Note that this node only works for material shading nodes; it does nothing for lamp and world shading nodes.

Location
: Location of the object in world space.

Object Index
: Object pass index, same as in the Object Index pass.transformed.

Material Index
: Material pass index, same as in the Material Index pass.

Random
: Random number between 0 and 1 unique to a single object instance.

Particle Info

For objects instanced from a particle system, this node give access to the data of the particle that spawned the instance. This node currently only supports parent particles, info from child particles is not available.

Index
: Index number of the particle (from 0 to number of particles).

Age
: Age of the particle in frames.

Lifetime
: Total lifespan of the particle in frames.

Location
: Location of the particle.

Size

Size of the particle.

Velocity
Velocity of the particle.

Angular Velocity
Angular velocity of the particle.

Hair Info

This node gives access to strand information.

Is strand
Returns 1 when the shader is acting on a strand, otherwise 0.

Intersect
The point along the strand where the ray hits the strand (1 at the tip and 0 at the root).

Thickness
The thickness of the strand at the point where the ray hits the strand.

Tangent Normal
Tangent normal of the strand.

Attribute

Retrieve attribute attached to the object or mesh. Currently UV maps and vertex color layers can be retrieved this way by their names, with layers and attributes planned to be added. Also internal attributes like *P* (position), *N* (normal), *Ng* (geometric normal) may be accessed this way, although there are more convenient nodes for this.

Name
Name of the attribute.

Color output
RGB color interpolated from the attribute.

Vector output
XYZ vector interpolated from the attribute.

Fac output
Scalar value interpolated from the attribute.

Mapping

Transform a coordinate; typically used for modifying texture coordinates.

Location
Vector translation.

Rotation
Rotation of the vector along XYZ axes.

Scale
Scale of the vector.

Vector input
Vector to be transformed.

Vector output
Transformed vector.

Layer Weight

Output weights typically used for layering shaders with the *Mix Shader* node.

Blend input
: Blend between the first and second shader.

Fresnel output
: Dielectric fresnel weight, useful for example to layer diffuse and glossy shaders to create a plastic material. This is like the *Fresnel* node, except that the input of this node is in the often more-convenient 0.0 to 1.0 range.

Facing output
: Weight that blends from the first to the second shader as the surface goes from facing the viewer to viewing it at a grazing angle.

Fresnel

Dielectric fresnel, computing how much light is reflected off a layer, where the rest will be refracted through the layer. The resulting weight can be used for layering shaders with the *Mix Shader* node. It is dependent on the angle between the surface normal and the viewing direction.

The most common use is to mix between two BSDFs using it as a blending factor in a mix shader node. For a simple glass material you would mix between a glossy refraction and glossy reflection. At grazing angles more light will be reflected than refracted as happens in reality.

For a two-layered material with a diffuse base and a glossy coating, you can use the same setup, mixing between a diffuse and glossy BSDF. By using the fresnel as the blending factor you're specifying that any light which is refracted through the glossy coating layer would hit the diffuse base and be reflected off that.

IOR input
: Index of refraction of the material being entered.

Fresnel output
: Fresnel weight, indicating the probability with which light will reflect off the layer rather than passing through.

Light Path

Node to find out for which kind of incoming ray the shader is being executed; particularly useful for non-physically based tricks. More information about the meaning of each type is in the *Light Paths* documentation.

Is Camera Ray output
: 1.0 if shading is executed for a camera ray, 0.0 otherwise.

Is Shadow Ray output
: 1.0 if shading is executed for a shadow ray, 0.0 otherwise.

Is Diffuse Ray output
: 1.0 if shading is executed for a diffuse ray, 0.0 otherwise.

Is Glossy Ray output
: 1.0 if shading is executed for a glossy ray, 0.0 otherwise.

Is Singular Ray output
: 1.0 if shading is executed for a singular ray, 0.0 otherwise.

Is Reflection Ray output
: 1.0 if shading is executed for a reflection ray, 0.0 otherwise.

Is Transmission Ray output

1.0 if shading is executed for a transmission ray, 0.0 otherwise.

Ray Length output
Distance travelled by the light ray from the last bounce or camera.

Ray Depth output
Returns the current light bounce.

Transparent Depth output
Returns the number of transparent surfaces passed through.

Light Falloff

Manipulate how light intensity decreases over distance. In reality light will always fall off quadratically; however it can be useful to manipulate as a non-physically based lighting trick. Note that using Linear or Constant falloff may cause more light to be introduced with every global illumination bounce, making the resulting image extremely bright if many bounces are used.

Strength input
Light strength before applying falloff modification.

Smooth input
Smooth intensity of light near light sources. This can avoid harsh highlights, and reduce global illumination noise. 0.0 corresponds to no smoothing; higher values smooth more. The maximum light strength will be strength/smooth.

Quadratic output
Quadratic light falloff; this will leave strength unmodified if smooth is 0.0 and corresponds to reality.

Linear output
Linear light falloff, giving a slower decrease in intensity over distance.

Constant output
Constant light falloff, where the distance to the light has no influence on its intensity.

Nodes shared with the Compositor

Some nodes are common with Composite nodes, their documentation can be found at their relevant pages rather than repeated here.

- *Brightness Contrast*
- *Separate RGB*
- *Combine RGB*
- *Separate HSV*
- *Combine HSV*
- *Gamma*
- *Hue Saturation Value*
- *Invert*
- *Math*
- *Mix RGB*
- *RGB Curves*
- *RGB to BW*
- *Vector Curve*