

## 10.1.10 Editors - Shader Editor - Header - Add Menu - Vector

Detailed table of content.....	1
Add menu - Vector.....	4
Bump.....	4
Displacement.....	5
Mapping.....	6
Normal.....	7
Normal Map.....	8
Vector Curves.....	8
Vector Displacement.....	10
Vector Rotate.....	11
Vector Transform.....	12

### Detailed table of content

#### Detailed table of content

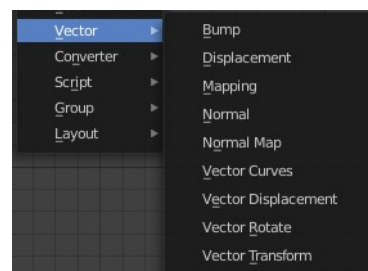
Detailed table of content.....	1
Add menu - Vector.....	4
Bump.....	4
Inputs.....	4
Strength.....	4
Distance.....	4
Height.....	4
Normal.....	4
Properties.....	4
Invert.....	4
Outputs.....	4
Normal.....	4
Displacement.....	5
Inputs.....	5
Height.....	5
Midlevel.....	5
Scale.....	5
Normal.....	5
Properties.....	5
Space.....	5
Outputs.....	5
Displacement.....	5
Mapping.....	6
Inputs.....	6
Vector.....	6
Location.....	6
Rotation.....	6
Scale.....	6
Properties.....	6
Vector type.....	6
Point.....	6
Texture.....	6

Vector.....	7
Normal.....	7
Outputs.....	7
Vector.....	7
Normal.....	7
Inputs.....	7
Normal.....	7
Properties.....	7
Normal Direction.....	7
Outputs.....	7
Normal.....	7
Dot.....	7
Normal Map.....	8
Inputs.....	8
Strength.....	8
Color.....	8
Properties.....	8
Space.....	8
UV Map.....	8
Outputs.....	8
Normal.....	8
Vector Curves.....	8
Inputs.....	8
Factor.....	9
Vector.....	9
Channel buttons.....	9
Curve edit field.....	9
Selecting Points.....	9
Adding Points.....	9
Navigation elements.....	9
Zoom in and out.....	9
Tools.....	9
Reset View.....	9
Vector Handle.....	9
Auto Handle.....	10
Auto Clamped Handle.....	10
Extend horizontal.....	10
Extend extrapolated.....	10
Reset Curve.....	10
Use Clipping.....	10
Delete Points.....	10
Outputs.....	10
Vector.....	10
Vector Displacement.....	10
Inputs.....	10
Vector.....	10
Midlevel.....	11
Scale.....	11
Properties.....	11
Space.....	11
Outputs.....	11
Displacement.....	11
Vector Rotate.....	11

Inputs.....	11
Vector.....	11
Center.....	11
Axis.....	11
Angle.....	11
Properties.....	11
Type.....	11
Outputs.....	12
Vector.....	12
Vector Transform.....	12
Inputs.....	12
Vector Input.....	12
Properties.....	12
Type.....	12
Convert From.....	12
Convert To.....	12
Outputs.....	12
Vector Output.....	12

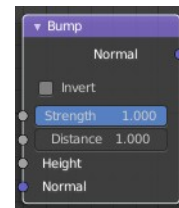
## Add menu - Vector

Here you find nodes that deals with vector data.



### Bump

The Bump node generates 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.



### Inputs

#### **Strength**

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

#### **Distance**

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

#### **Height**

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

#### **Normal**

Standard normal input.

### Properties

#### **Invert**

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

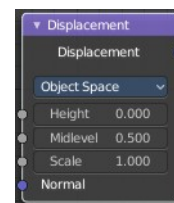
### Outputs

#### **Normal**

Standard normal output.

## Displacement

The Displacement node is used to displace the surface along all the surface normal, to add more detail to the geometry. Both procedural textures and baked displacement maps may be used.



For best results the mesh must be subdivided finely to bring out the detail in the displacement texture.

It is also possible to use the displacement as bump mapping only by changing the material settings, so that no high resolution mesh is needed.

## Inputs

### *Height*

Distance to displace the surface along the normal. This is where a texture node can be connected.

### *Midlevel*

Neutral displacement value that causes no displacement. With the default 0.5, any lower values will cause the surfaces to be pushed inwards, and any higher values will push them outwards.

### *Scale*

Increase or decrease the amount of displacement.

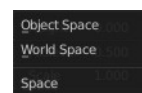
### *Normal*

Standard normal input.

## Properties

### *Space*

Object Space means the displacement scales along with the object. When using World Space the object scale is ignored.



## Outputs

### *Displacement*

Displacement offset to be connected into the Material Output.

## Mapping

The Mapping node transforms the input vector by applying translation, rotation, and scaling.

### Inputs

The inputs of the node are dynamic. In particular, the Location input is only available in the Texture and Point vector types.

### Vector

The vector to be transformed.

### Location

The amount of translation along each axis.

### Rotation

The amount of rotation along each axis. XYZ order.

### Scale

The amount of scaling along each axis.

## Properties

### Vector type

The node applies the transformation differently depending on the semantic type of the input vector.



### Point

For this vector type, the node performs a straightforward transformation.

Transforming a texture coordinates is analogous to transforming a UV map. For instance, translating the texture coordinates along the positive X axis would result in the evaluated texture to move in the negative X axis, much like if one translated a UV map. Similarly, scaling the texture coordinates up would result in the evaluated texture to scale down. So transforming the texture coordinates would appear to have the opposite effect on the evaluated texture.

The order of transformation is: Scale → Rotate → Translate, which means:

Translation moves the input along the local rotation axis.

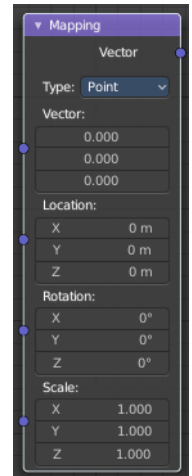
Rotation rotates the input around the origin of the space.

Scaling scales the input along the global axis.

### Texture

For this vector type, the node performs an inverse transformation.

Inverse transforming a texture coordinates would, as opposed to the Point type, transform the evaluated texture itself. For instance, translating the texture coordinates along the positive X axis would result in the evaluated texture to move in the positive X axis, as one would expected. Similarly, scaling the texture coordinates up



would result in the evaluated texture to scale up, as one would expect.

The order of transformation is: Translate → Rotate → Scale, which means:

Translation moves the input along the global axis.

Rotation rotates the input around the translation vector.

Scaling scales the input along the local rotation axis.

## Vector

For this vector type, a Point transformation is performed, but with zero translation.

## Normal

For this vector type, the node performs the inverse transpose of the transformation and normalize the result. Such transformation ensures correct normals after non-uniform scaling. So this type should be used when transforming normals.

## Outputs

### Vector

The input vector after transformation.

---

## Normal

The Normal node generates a normal vector and a dot product.

## Inputs

### Normal

Normal vector input.

## Properties

### Normal Direction

To manually set a fixed normal direction vector. LMB click and drag on the sphere to set the direction of the normal. Holding Ctrl while dragging snaps to 45 degree rotation increments.

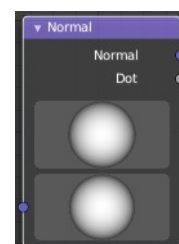
## Outputs

### Normal

Normal vector output.

### Dot

Dot product output. The dot product is a scalar value.



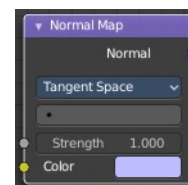
If two normals are pointing in the same direction the dot product is 1.

If they are perpendicular the dot product is zero (0).

If they are antiparallel (facing directly away from each other) the dot product is -1.

## Normal Map

The Normal Map node generates 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.



### Inputs

#### *Strength*

Strength of the normal mapping effect.

#### *Color*

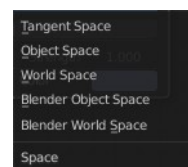
RGB color that encodes the normal in the specified space.

### Properties

#### *Space*

The input RGB color can be in one of three 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.

### Outputs

#### *Normal*

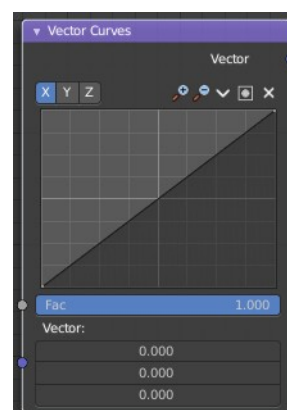
Normal that can be used as an input to BSDF nodes.

## Vector Curves

The Vector Curves node maps an input vector components to a curve.

### Inputs

In the shader context the node also has an additional Factor property.





## Factor

Controls the amount of influence the node exerts on the output vector.

## Vector

Standard vector input.

Properties

Channel

## Channel buttons

X, Y, Z. Clicking on one of the channels displays the curve for each.



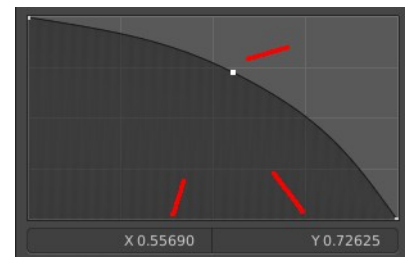
## Curve edit field

Here you can create and tweak a Bézier curve that varies the input levels (X axis) to produce an output level (Y axis).

### Selecting Points

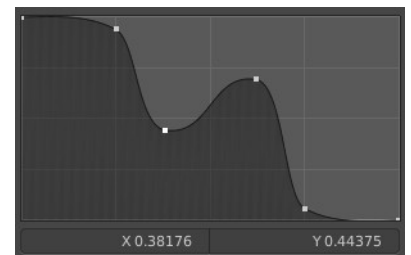
You can select curve points. This reveals two edit boxes for the x and y coordinate of this point.

Selected points can be moved around. Left click at them, hold the mouse button down and move them to a new location.



### Adding Points

You can add new curve points by simply left clicking at the curve. Move the mouse to position them where you need it.



## Navigation elements

The navigation elements at the top are described from left to right.

### Zoom in and out

The two buttons with the magnifying glass at it zooms in and out in the curve window.



## Tools

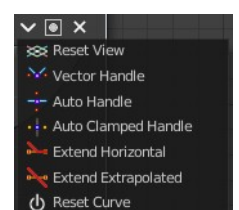
Tools is a menu where you can find some cuve related tools.

### Reset View

Resets the curve windows zoom.

### Vector Handle

Set handle type to Vector.



### **Auto Handle**

Set handle type to Auto.

### **Auto Clamped Handle**

Set handle type to Auto Clamped.

### **Extend horizontal**

Extends the curve before the first curve point and behind the last curve point horizontally.

### **Extend extrapolated**

Extends the curve before the first curve point and behind the last curve point extrapolated.

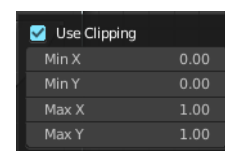
### **Reset Curve**

Resets the curve to the initial shape.

---

## **Use Clipping**

Clipping options. Here you can set up clipping for the stroke.



## **Delete Points**

Deletes selected curve points.

---

## **Outputs**

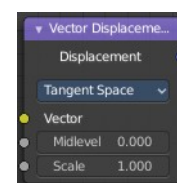
### **Vector**

Standard vector output.

---

## **Vector Displacement**

The Vector Displacement node is used to displace the surface along arbitrary directions, unlike the regular Displacement node which only displaces along the surface normal.



It is typically used to apply vector displacement maps created by other sculpting software.

Vector displacement maps can fully represent the high resolution detail to be applied on a smooth base mesh, unlike regular displacement maps.

For best results the mesh must be subdivided finely to bring out the detail in the displacement texture.

## **Inputs**

### **Vector**

Vector specifying the displacement along three axes. This is where a texture node can be connected.

Typically a baked vector displacement image texture is used. For Object Space, RGB colors in the image are interpreted as an XYZ offset in object space. For Tangent Space, R is an offset along the tangent, G along the normal and B along the bitangent.

## Midlevel

Neutral displacement value that causes no displacement. With the default 0.0, any lower values will cause the surfaces to be pushed inwards, and any higher values will push them outwards.

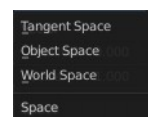
## Scale

Increase or decrease the amount of displacement.

## Properties

### Space

Object Space maps work for static meshes, and will render slightly faster with less memory usage. Tangent Space maps can be used for meshes that will be deformed, like animated characters, so the displacement follows the deformation.



## Outputs

### Displacement

Displacement offset to be connected into the Material Output.

## Vector Rotate

This node provides the ability to rotate a vector around a center point using either Axis Angle, Single Axis or Euler methods.

## Inputs

### Vector

The input vector.

### Center

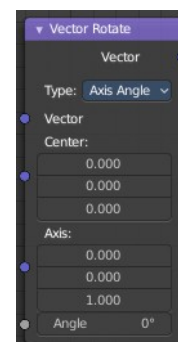
The center for the rotation.

### Axis

The axis angles.

### Angle

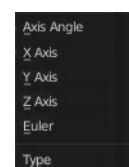
The rotation angle.



## Properties

### Type

The rotation type.



## Outputs

### *Vector*

The output vector

---

## Vector Transform

The Vector Transform node allows converting a vector, point, or normal between world and camera and object coordinate space.

### Inputs

#### *Vector Input*

Standard vector input.

### Properties

#### *Type*

Specifies the input/output type.

Vector, Point, Normal.

#### *Convert From*

Coordinate Space to convert from World, Object or Camera.

#### *Convert To*

Coordinate Space to convert to World, Object or Camera.

### Outputs

#### *Vector Output*

The transformed output vector.

