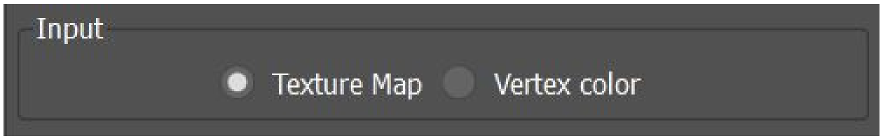
Bitmap Controller

This controller extract the RGB , HSV and Alpha values from a given bitmap (image) or procedural map or the vertex color channel of the mesh geometries

Input:



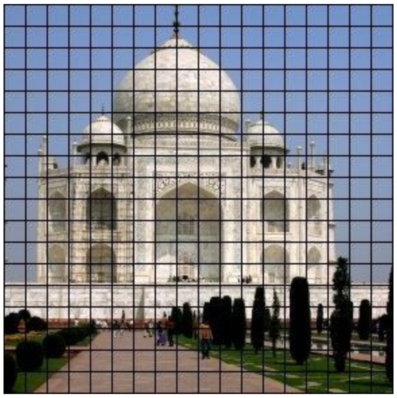
Input

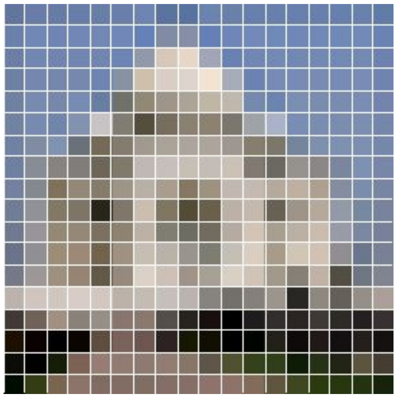
● Texture Map ○ Vertex color

.

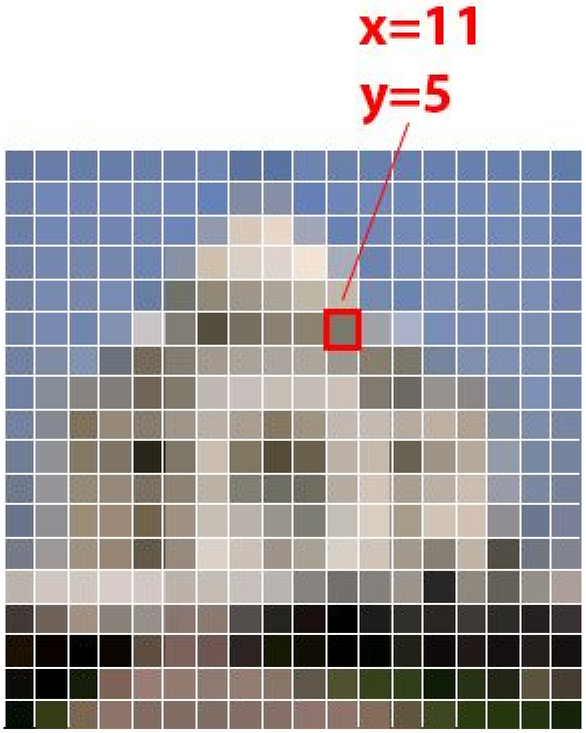
Here you can select between texture map and vertex color. If you choose texture map then you need to select a bitmap or a procedural map from the map and material library. If the vertex color is selected then an object with a vertex color channel must be selected from the scene.

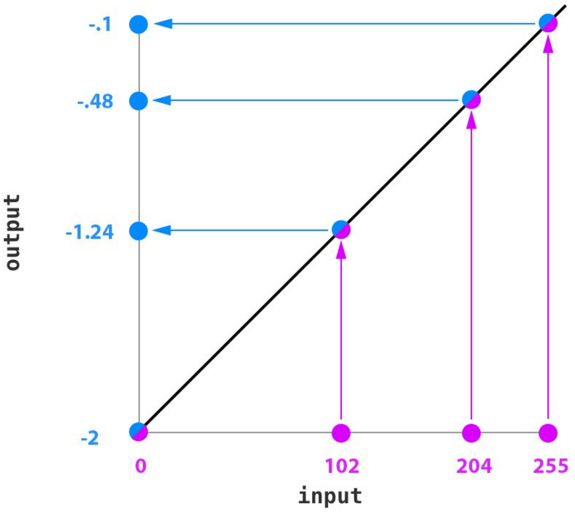
Texture map:





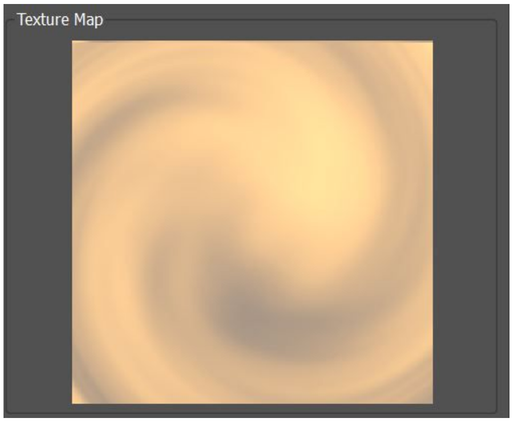
A Bitmap controller extract information from bitmaps and procedural maps. Maps are considered as 2D domain in which each point in the domain can be addressed by a 2D vector. The image on the left demonstrate bitmap that has been subdivided using an 18×18 domain. The Bitmap controller then create a table of the colors by finding the average color in each sample of the image (the image on the right). From this table we can now recall the value stored in each cell by having the vector position of that particular cell.(The image in below) Note the bitmap controller reads the samples from top-left corner to the bottom right corner.





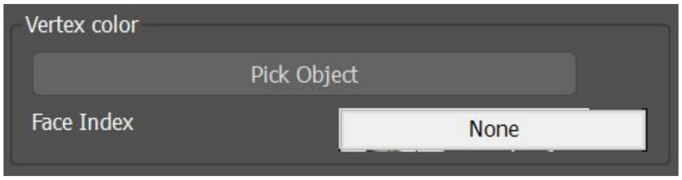
Normally in each cell 7 values are stored that are corresponding to the Red,Green,Blue,Hue,Saturation,Value and Alpha of the sample. Each of these values can vary between o to 255. Once the value is retrieved it will be remapped to the output range. for example if you choose -.1 and -2 for the Min and Max range , the output will be adjusted to the target range using a linear mapping technique. (right figure)

Map button:

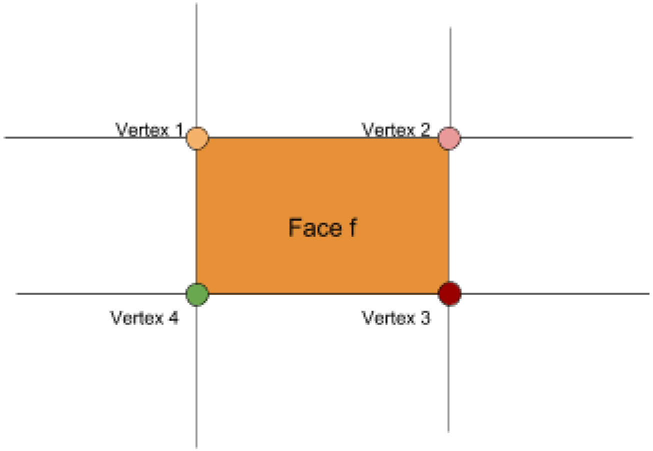


When the texture map is selected you need to assign a map to the controller. Left click on the button opens up the map/material browser and you can choose between the maps available in the library. Right click on the button opens up the material editor so that you can edit the selected map. When the material editor is open you can right click on the button to refresh the map. Also if you modify the bitmap image assigned to the controller you can refresh the image shown in controller settings by right clicking on the button.

Vertex color:

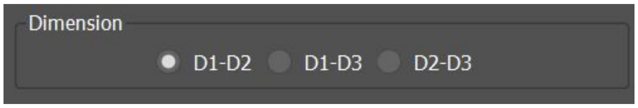


When vertex color is specified as the source of the color, then user must select an Editable poly object (an object with Edit-poly modifier) with vertex paint channel. You can use vertex paint modifier to add vertex color channel to a mesh object. Since vertex paint modifier stores the color values in the vertices of the mesh and not in the faces,The controller first retrieves the vertices of the given face then the average color of the vertices will be calculated.



Face 'f' has 4 vertices , the color of the face is the average color of all the vertices. Once the color is retrieved the result will be calculated similar to the texture map (see texture map).

Dimension:



If the texture map is used the settings in dimension group defines the 2D domain in which the bitmap will be subdivided. Let us assume the controller is used under a container node with 20,30 and 10 members in D1,D2 and D3 respectively. If the D1-D2 is selected the bitmap will be subdivided in 20x30 and if D1-D3 is selected the bitmap will be subdivided in 20x10 divisions.
If the vertex color is used and no controller has been assigned to the face index channel then the first dimension will be used to retrieve the face in Edit-poly object. For example if D1-D2 or

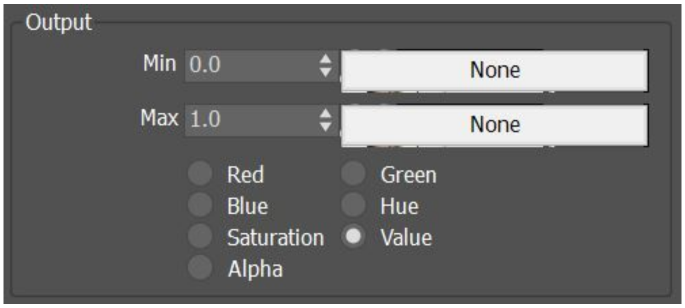
D1-D3 is selected the face index will be defined by the index1 and if D2-D3 is selected the face index will be defined by index2.

> ***Note :*** If it's required to associate the third dimension with face index then an ***Array variable controller*** can be assigned to the face index channel and from there the index3 can be selected.

Output:

Bitmap controller supports Scalar and Vector types.

Scalar Output:



Output

Min 0.0 None

Max 1.0 None

○ Red ○ Green
○ Blue ○ Hue
○ Saturation ● Value
○ Alpha

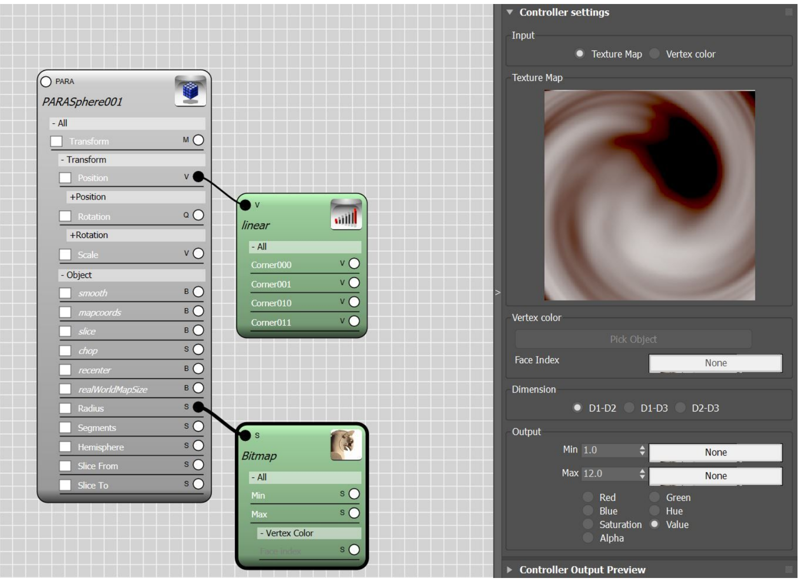
You can choose a scalar component of the color. The values of each component are integer values between 0 and 255. The output is then remapped to the range defined by Min and Max settings. (see texture map)

> ***Note:*** The alpha channel will return the min value if the selected bitmap has no alpha channel.

Example 1:

In below example the bitmap controller is used to modify the radius of the sphere in a 2D array. In this example "value" has been selected as output therefore the white area of the image returns the maximum value and the dark area returns the minimum .

> ***Note:*** The image looks upside down in the scene , this is because the controller reads the map from top-left corner to bottom right corner while the Y axis of the scene is from the bottom to the top.



▼ **Controller settings**

Input

● Texture Map ○ Vertex color

Texture Map

Vertex color

Pick Object

Face Index: None

Dimension

● D1-D2 ○ D1-D3 ○ D2-D3

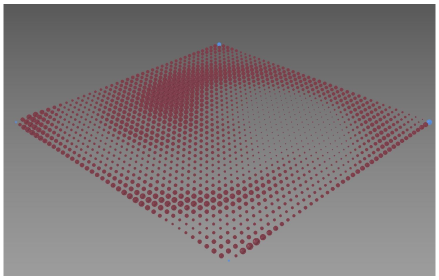
Output

Min 1.0 None

Max 12.0 None

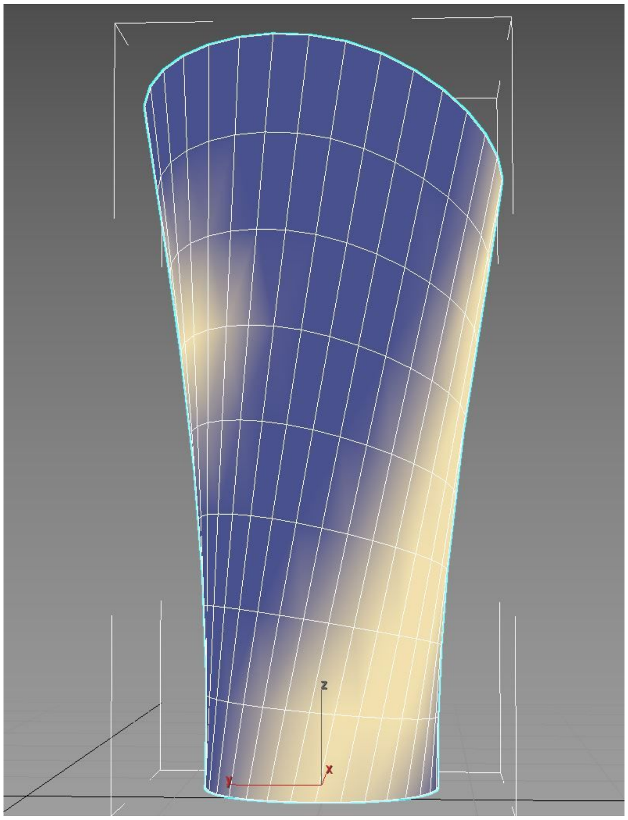
○ Red ○ Green
○ Blue ○ Hue
○ Saturation ● Value
○ Alpha

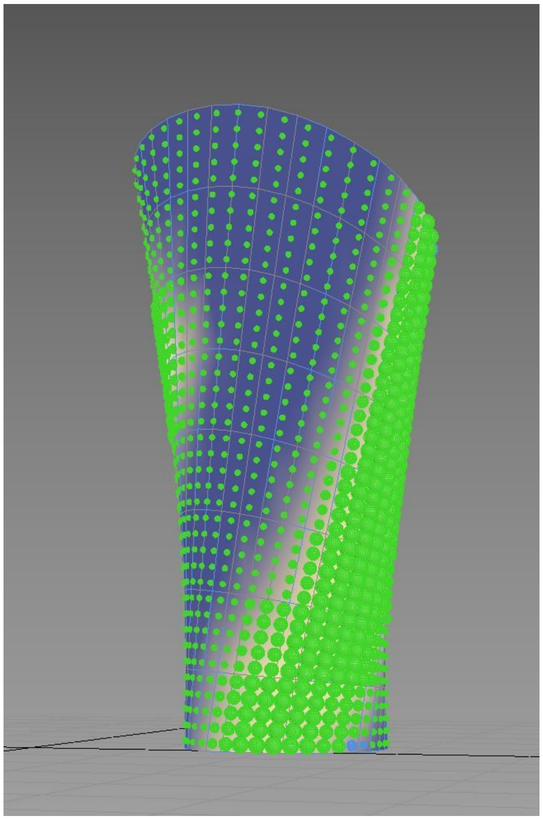
▶ **Controller Output Preview**

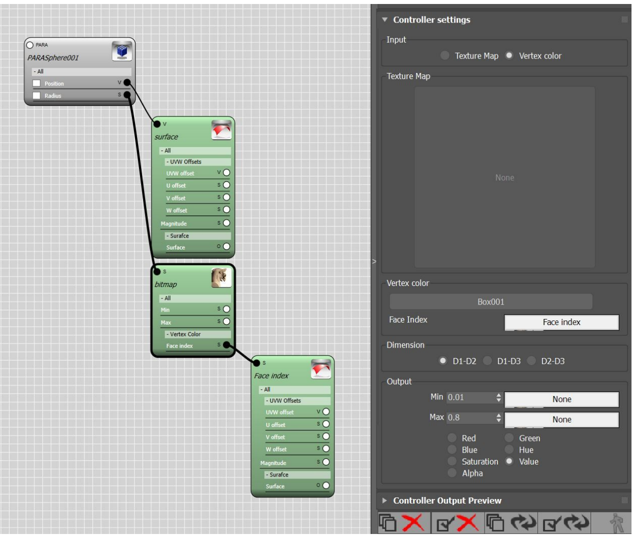


Example 2:

In below example the vertex color is selected as source of the color. The face index is provided by the surface controller to the face index channel.







Controller settings

Input: ○ Texture Map ● Vertex color

Texture Map: None

Vertex color: Box001

Face Index: Face index

Dimension: ● D1-D2 ○ D1-D3 ○ D2-D3

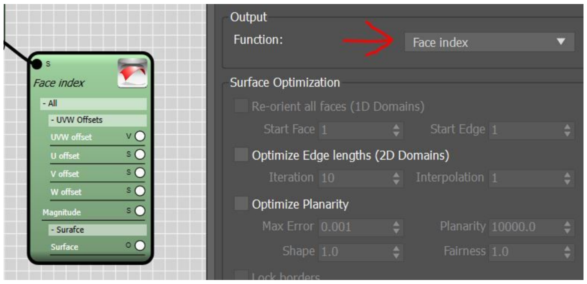
Output

Min 0.01 None

Max 0.8 None

○ Red ○ Green
○ Blue ○ Hue
○ Saturation ● Value
○ Alpha

Controller Output Preview



Output

Function: Face index

Surface Optimization

□ Re-orient all faces (1D Domains)

Start Face 1 Start Edge 1

□ Optimize Edge lengths (2D Domains)

Iteration 10 Interpolation 1

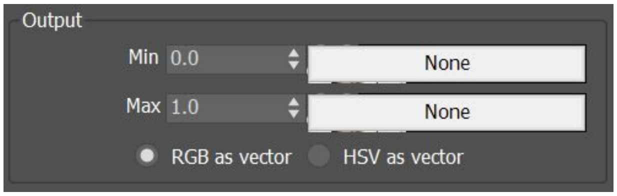
□ Optimize Planarity

Max Error 0.001 Planarity 10000.0

Shape 1.0 Fairness 1.0

□ Lock borders

Vector Output:



Output

Min 0.0 None

Max 1.0 None

● RGB as vector ○ HSV as vector

RGB as vector:

The controller returns the RGB color in vector format where the X ,Y and Z are corresponding to red,green and blue. The values are mapped from [0,255] to the given domain [min,max].

HSV as vector:

Similar to above except that X,Y and Z are corresponding to Hue,Saturation and Value.