



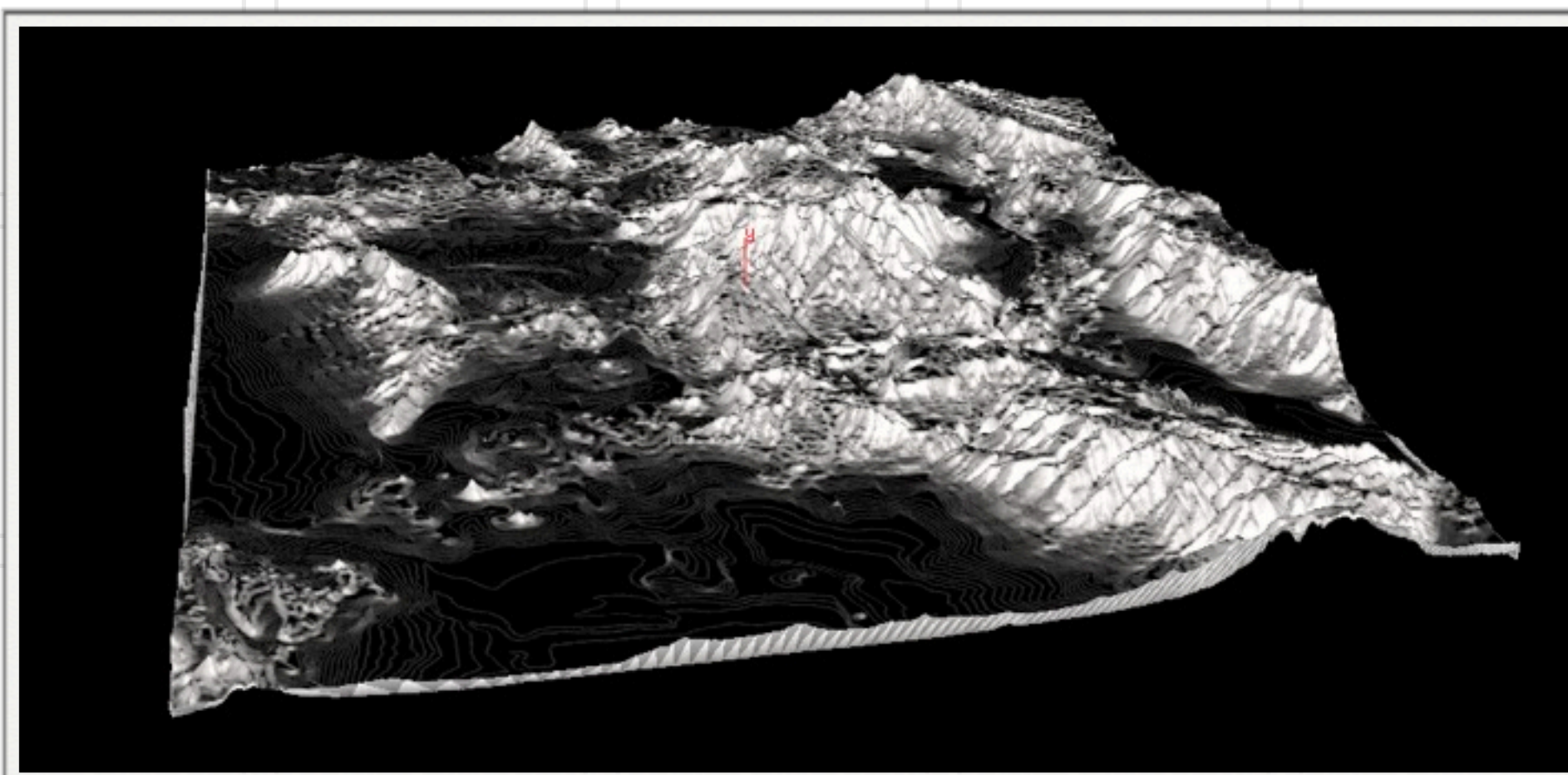
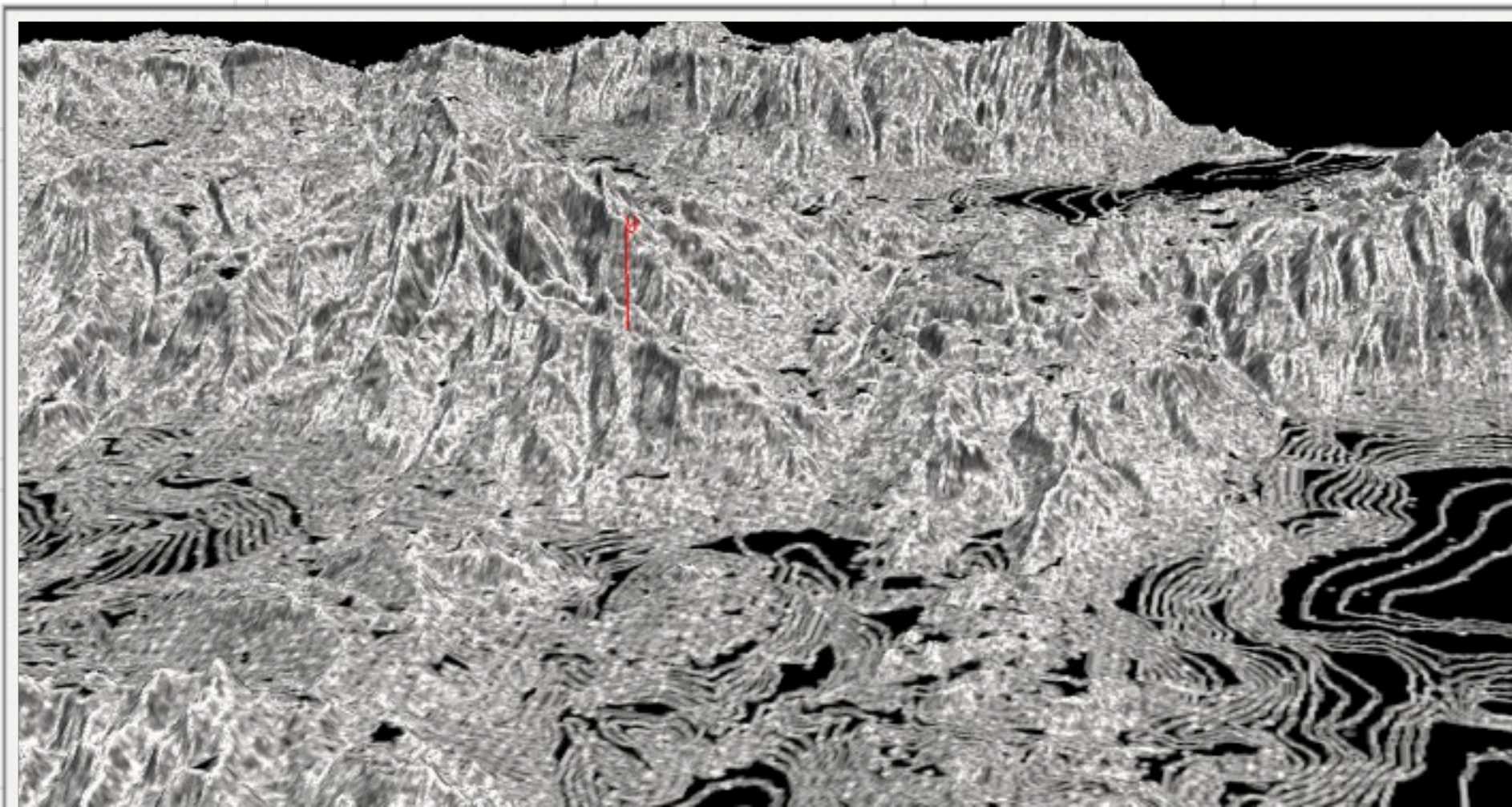
Next Steps: Houdini Procedural Modeling

M07: Building Vegetation Attributes into Terrain

Ari Danesh
ari@sidefx.com

**SIDE EFFECTS
SOFTWARE**

Agenda



- ▶ Bring in a height map to create terrain
- ▶ Calculate curvature of terrain
- ▶ Calculate slope of terrain
- ▶ Calculate peaks of terrain
- ▶ Apply grass attribute to flat areas
- ▶ Snow to hills
- ▶ Rocks to peaks

SIDE EFFECTS
SOFTWARE

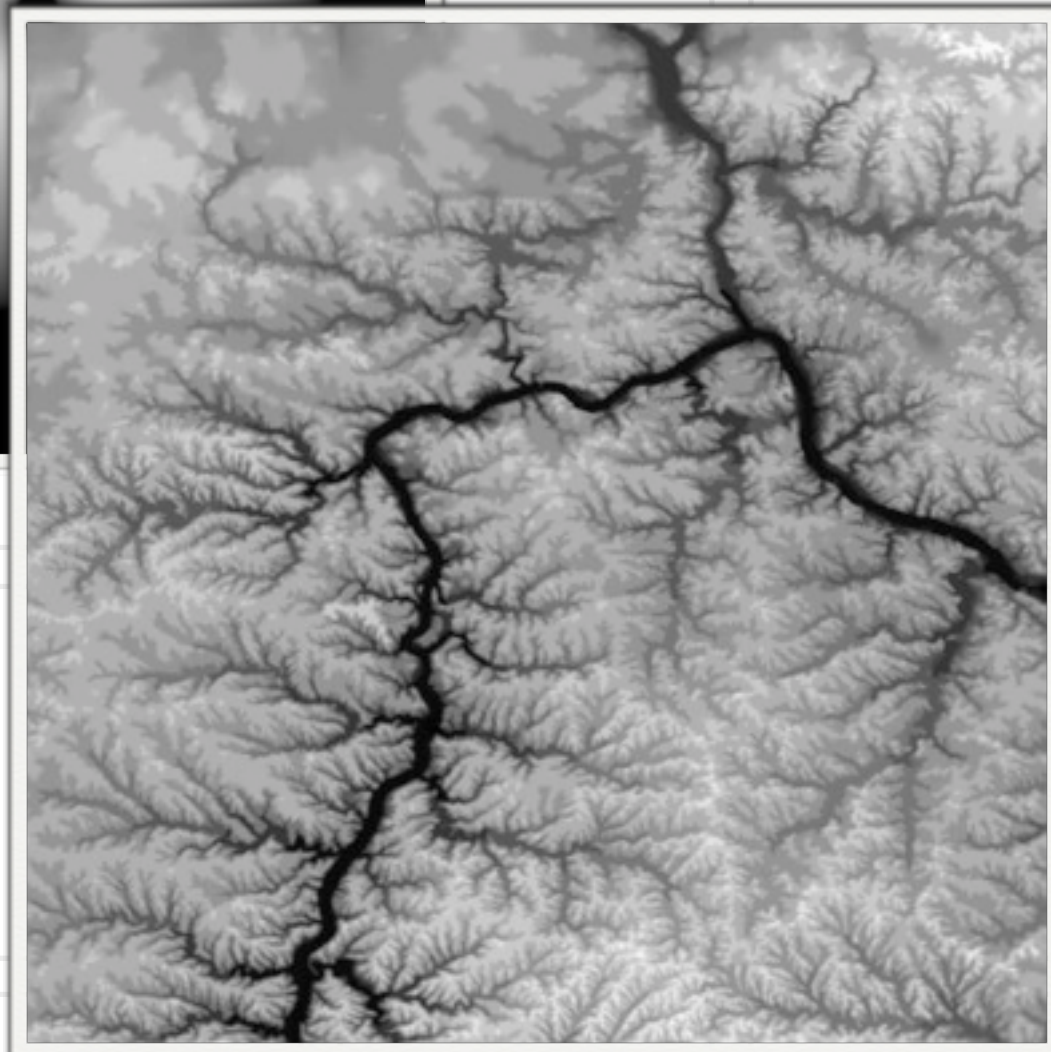
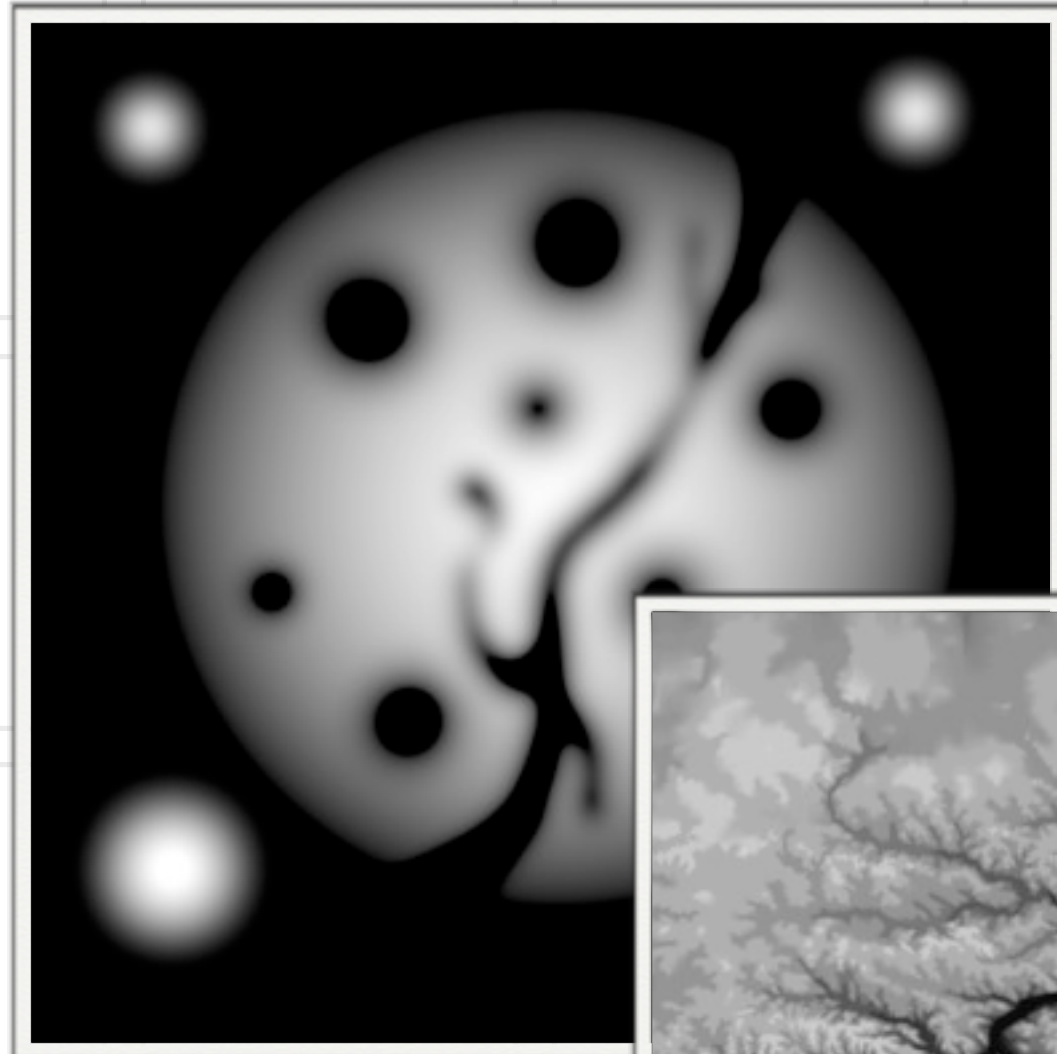


Height Maps

Ways to create Terrain from Images

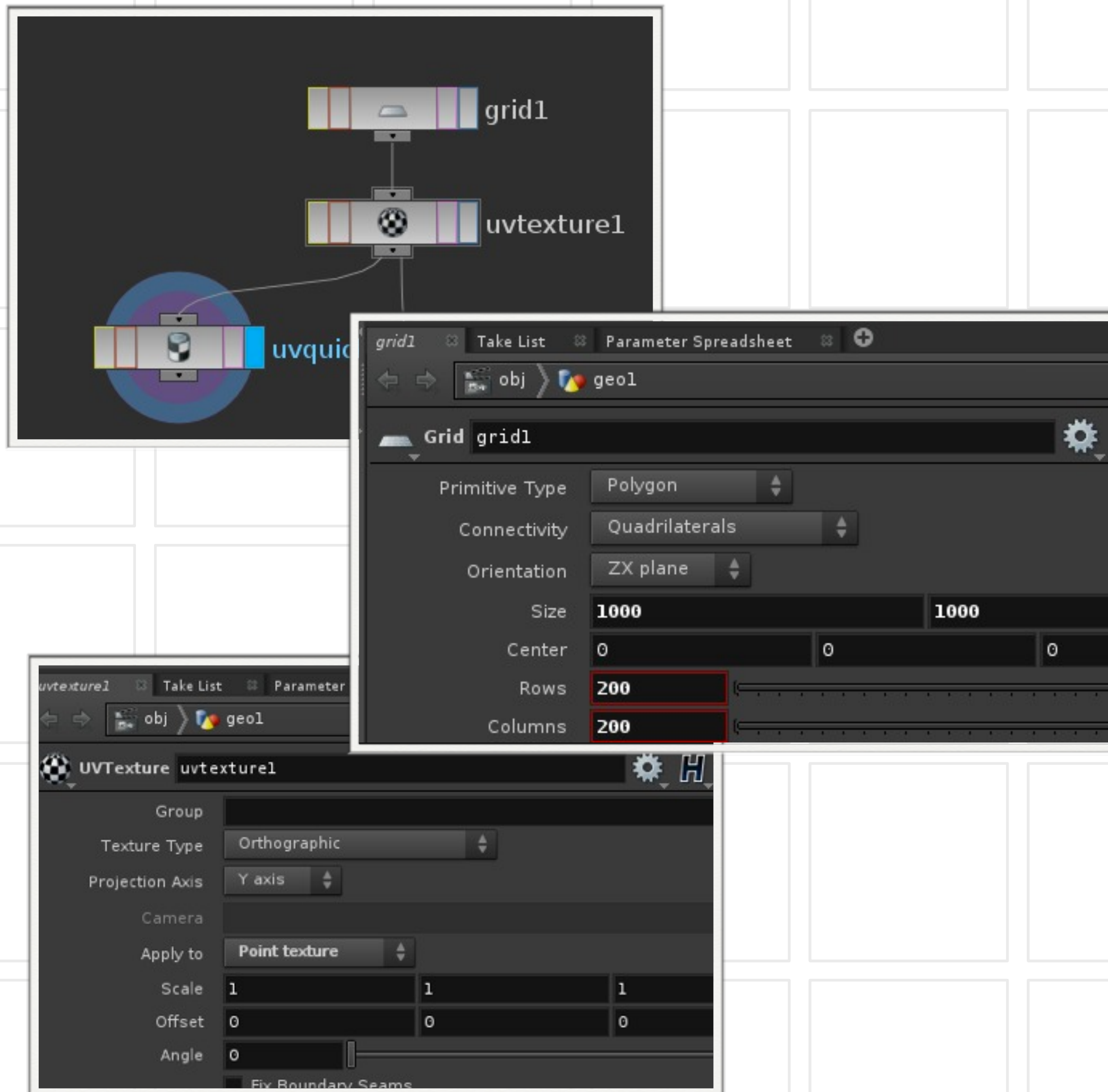
**SIDE EFFECTS
SOFTWARE**

Different Types of Height Maps



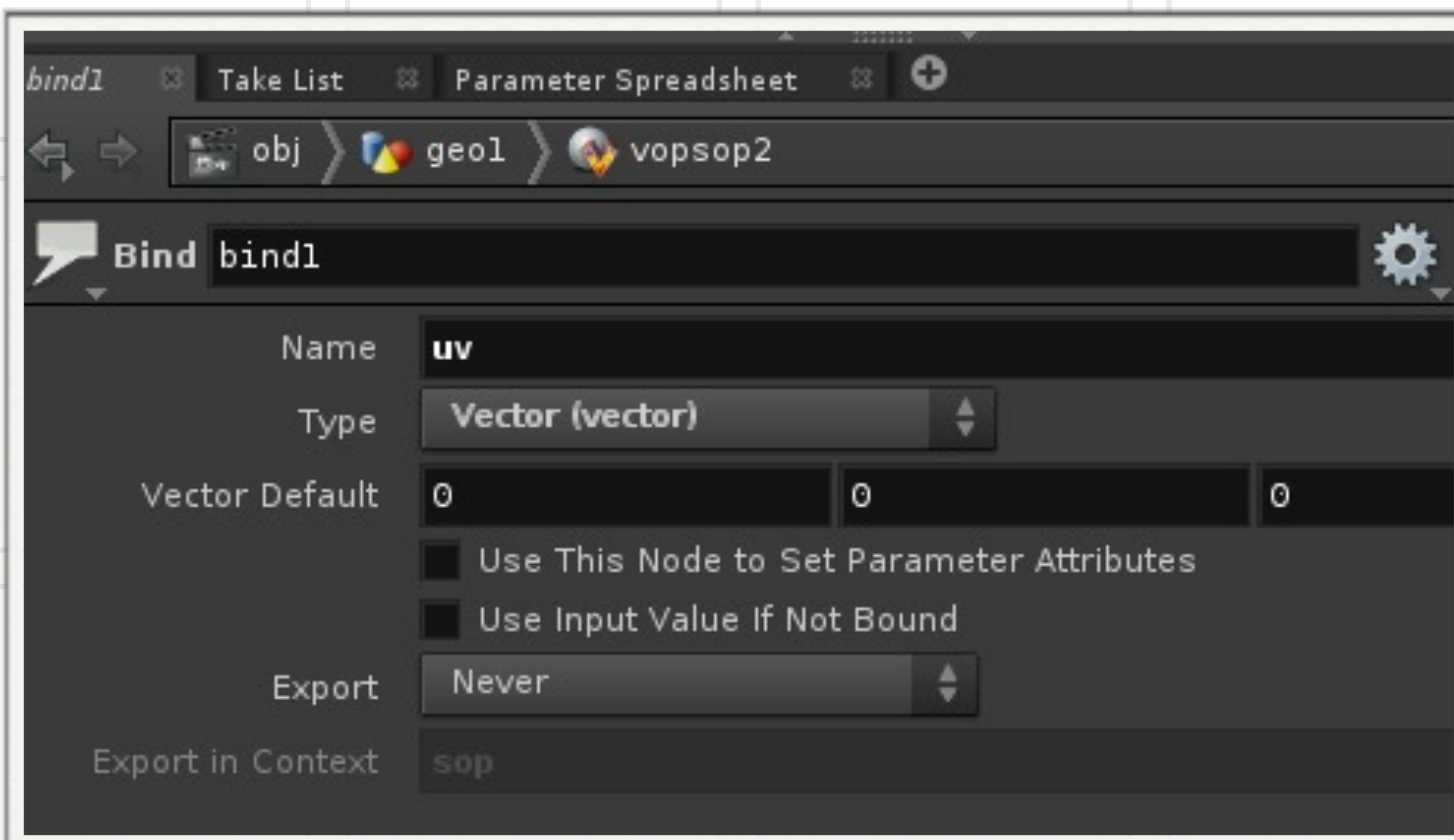
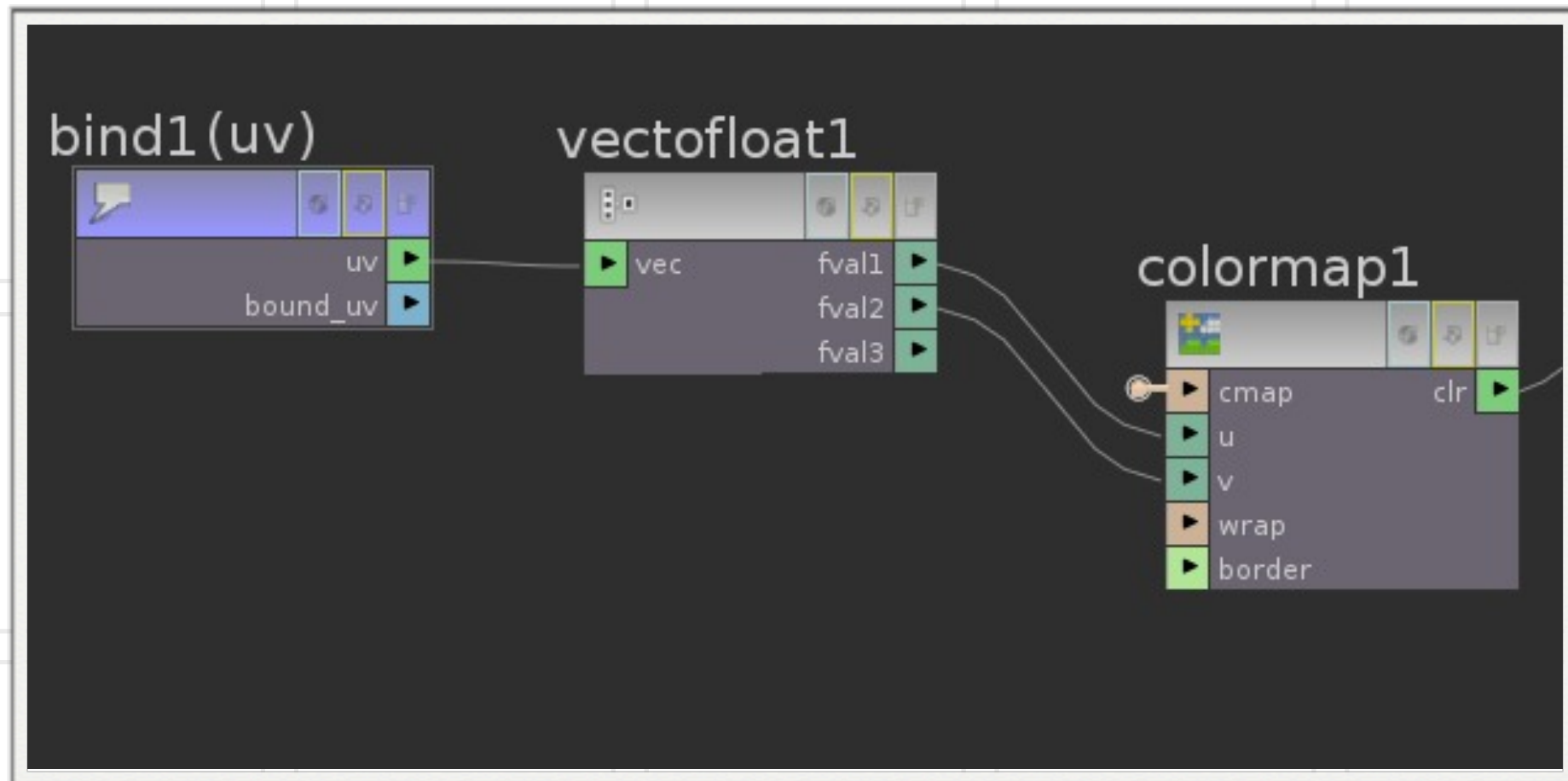
- ▶ DTED - Digital Terrain Elevation Data
- ▶ Photoshop - Paint your own
- ▶ DEM

Loading in a Height Map



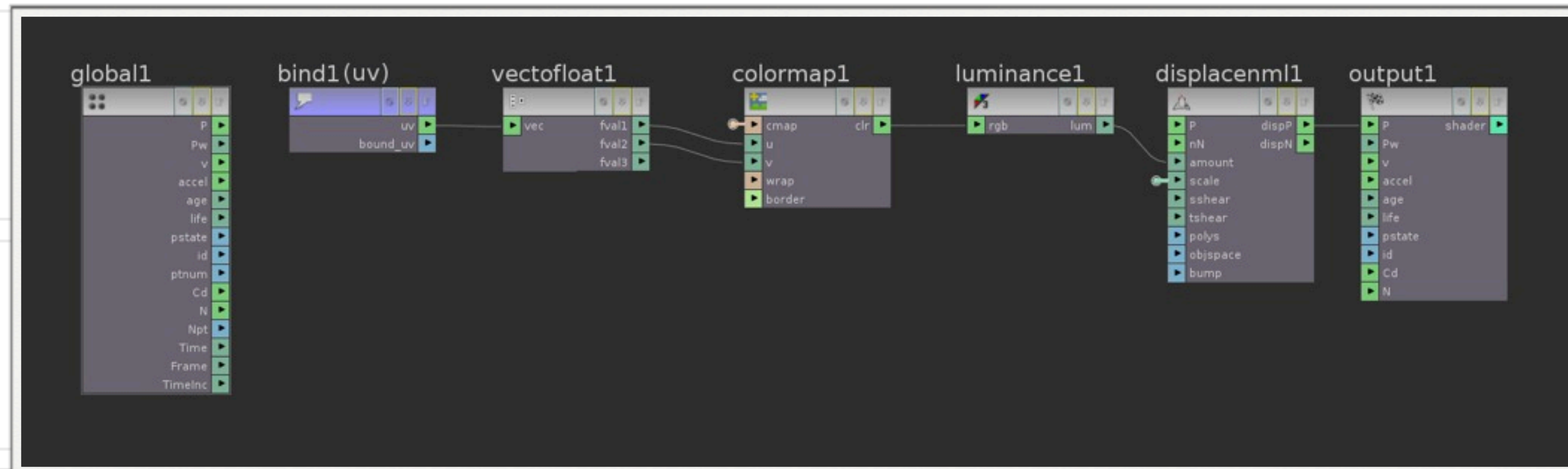
- ▶ Drop down a Grid
 - ▶ Make is 1000x1000 in width and height
 - ▶ Make it 200x200 in rows and columns
- ▶ Append a UVTexture
 - ▶ Texture Type - Orthographic
 - ▶ Projection Axis - Y
 - ▶ Apply to - Point Texture

Loading in the Height Map



- ▶ Append a VOPSOP and Dive inside
- ▶ We need uvs
 - ▶ Drop down a Bind SOP
 - ▶ Name - uv
 - ▶ Type - Vector
 - ▶ Drop Down a Color Map to bring in our image
 - ▶ Promote “cmap” - Color Map
 - ▶ Notice it needs u and v separately
 - ▶ Drop down a Vector to Float VOP
- ▶ Wire it as show on right

Loading in the Height Map (cont.)

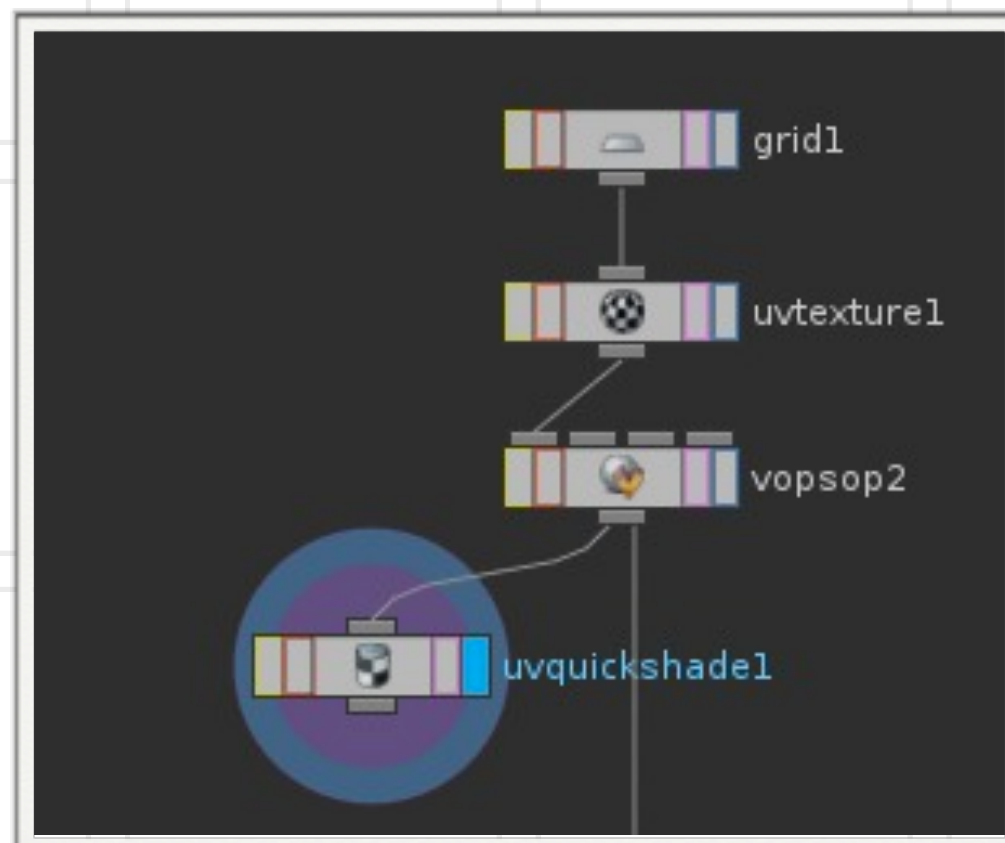


- ▶ Right now we have a color map coming in but what we really want is calculate the luminance. The brighter the image is the more mountainous and darker the more flat.
- ▶ Append a Luminance VOP to the Color Map. Note the output of the luminance is no longer a Vector but a Float
- ▶ Append a Displace Along Normal VOP. Wire the output of the Luminance to the “Amount” parameter of the Displace along Normal.
- ▶ Promote the Scale parameter
- ▶ Wire dispP to Global Output Position

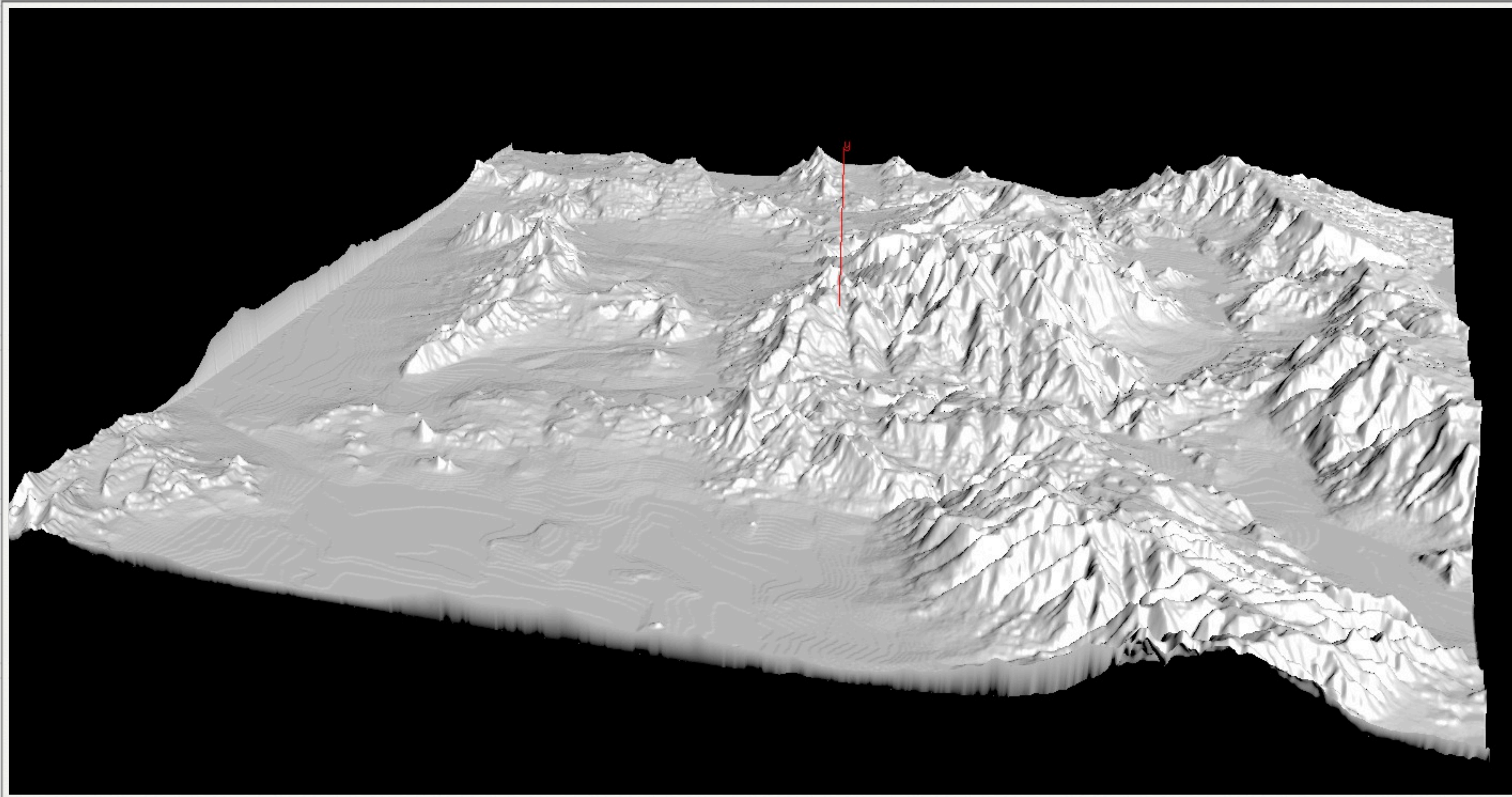
Test Out The Height Map

New Section of
User Interface

- ▶ Pop back up to the Geometry level
- ▶ Select the VOPSOP - You Interface should look like the top left image
- ▶ Load up a terrain image (provided a couple on disk)
- ▶ Test Out Scale and see the results



Example Results



**SIDE EFFECTS
SOFTWARE**



Calculating Curvature

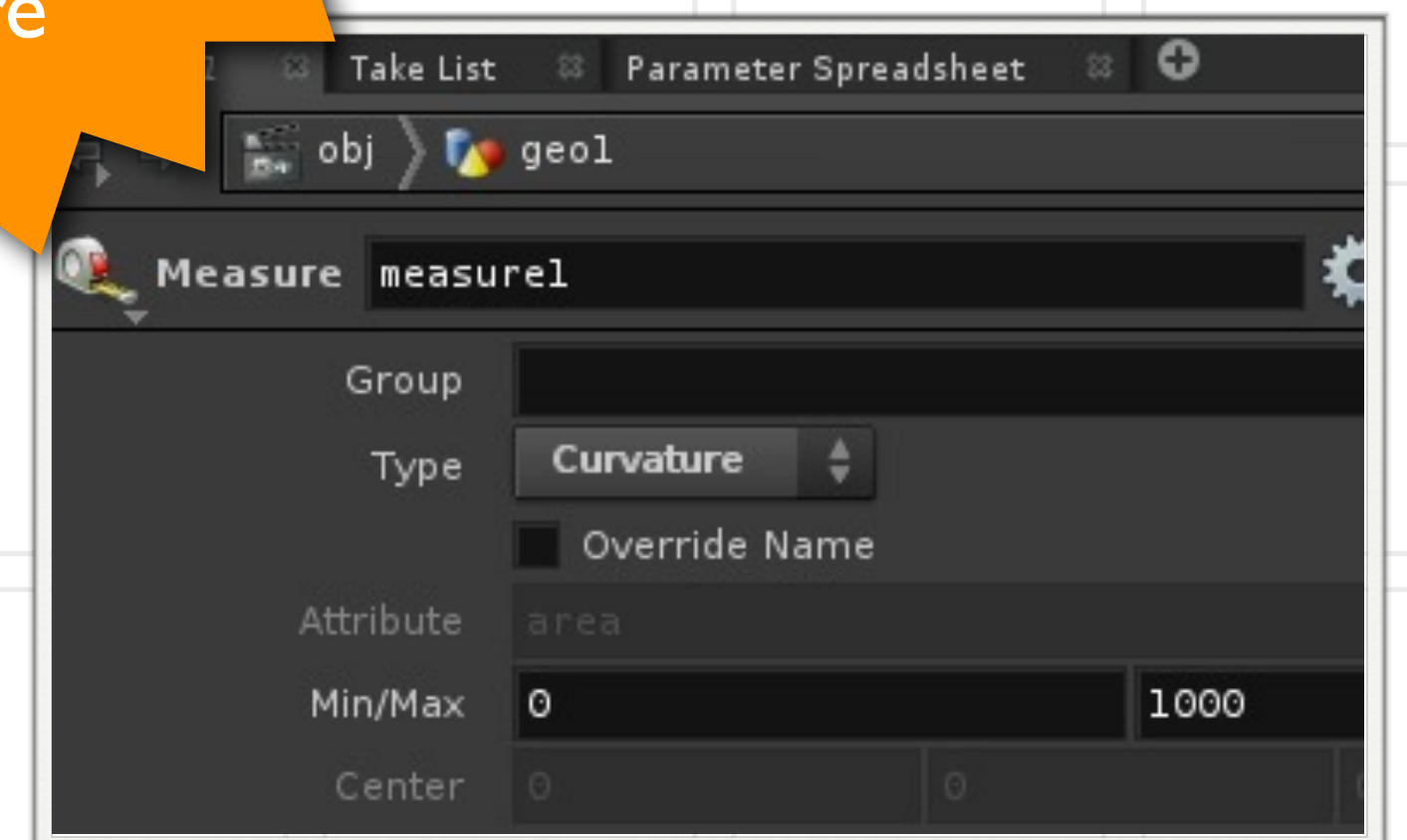
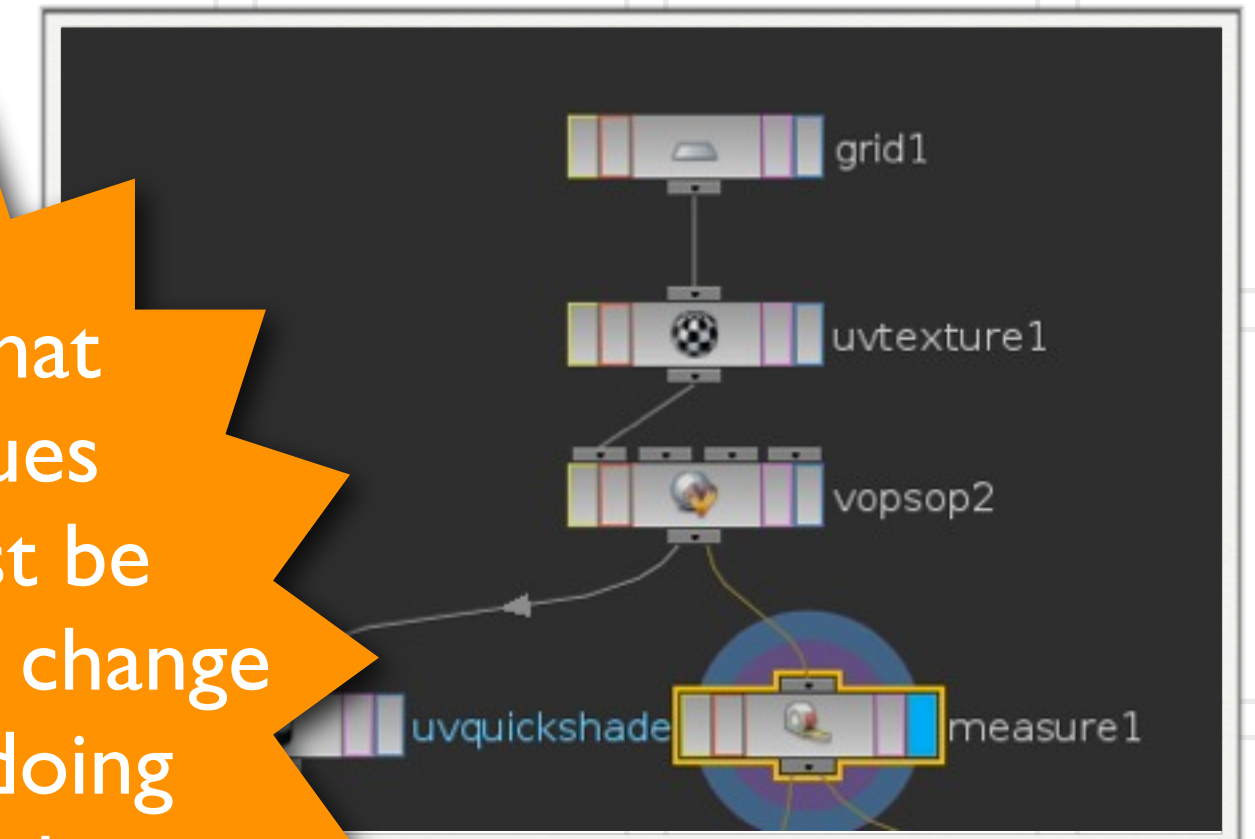
Using the Measure SOP and VOPSOP

**SIDE EFFECTS
SOFTWARE**

The Measure SOP

- ▶ Append a Measure SOP to the VOPSOP
- ▶ Open up the Details View
 - ▶ Look at the values for curvature
 - ▶ Now change the Grid resolution
 - ▶ Notice the Curvature Values Change
 - ▶ Change the Grid from Polygon to NURBs
 - ▶ Notice the Curvature values change again

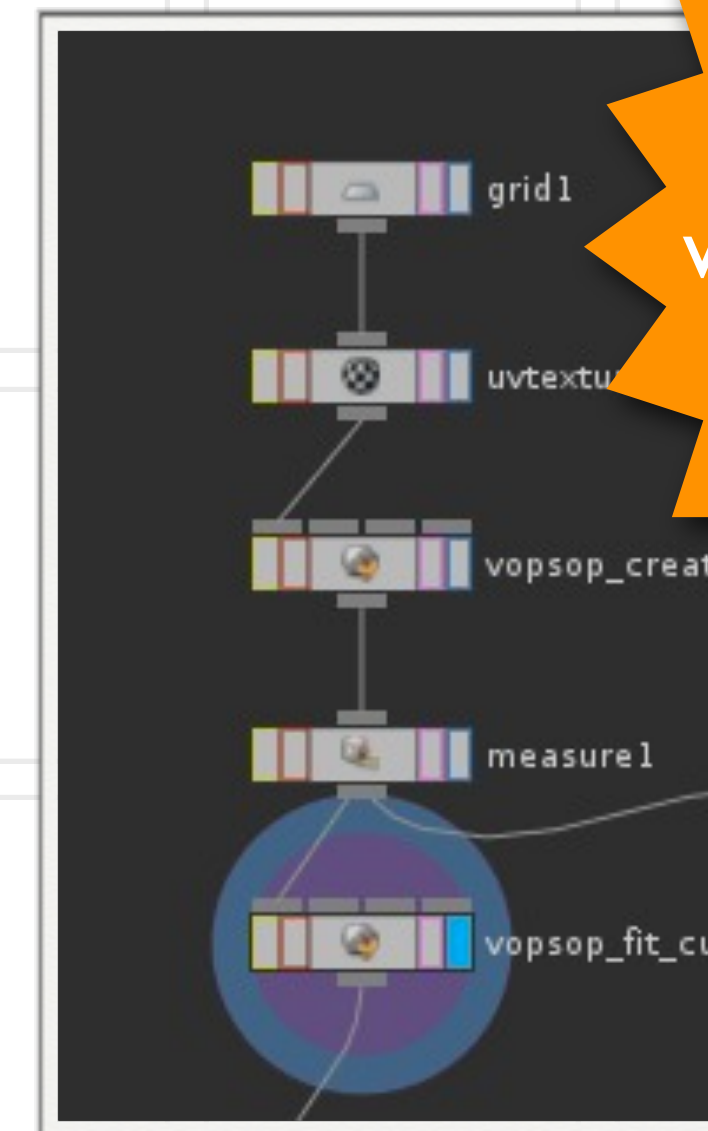
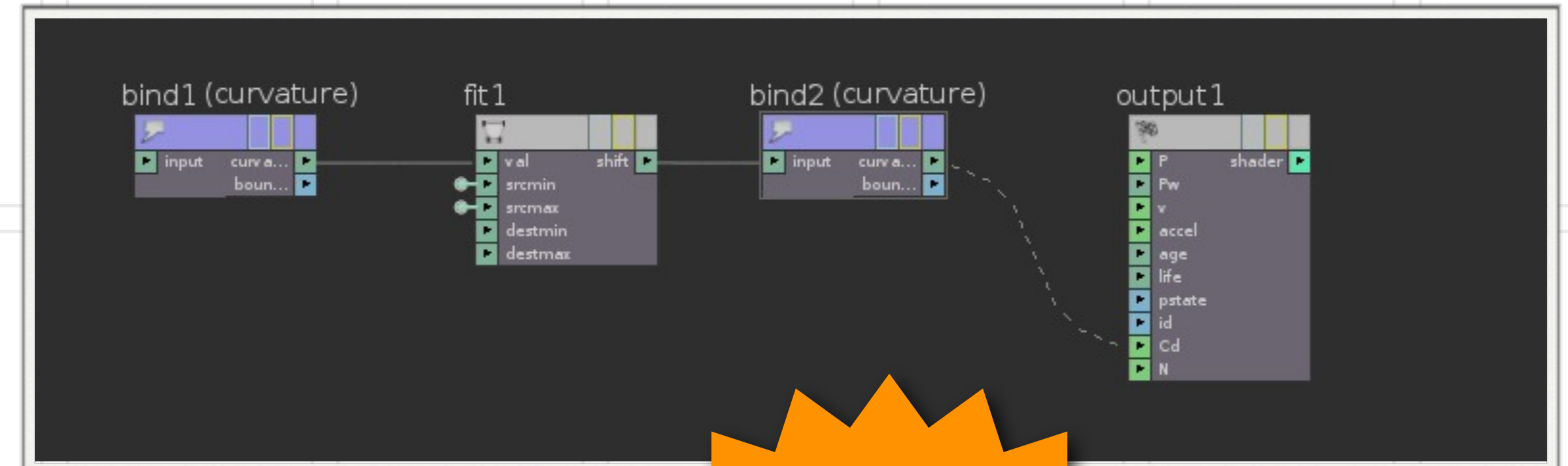
While it OK that Curvature values change you must be aware that they do change in case you are doing calculations based on Curvature



SIDE EFFECTS
SOFTWARE

Analyzing the Curvature

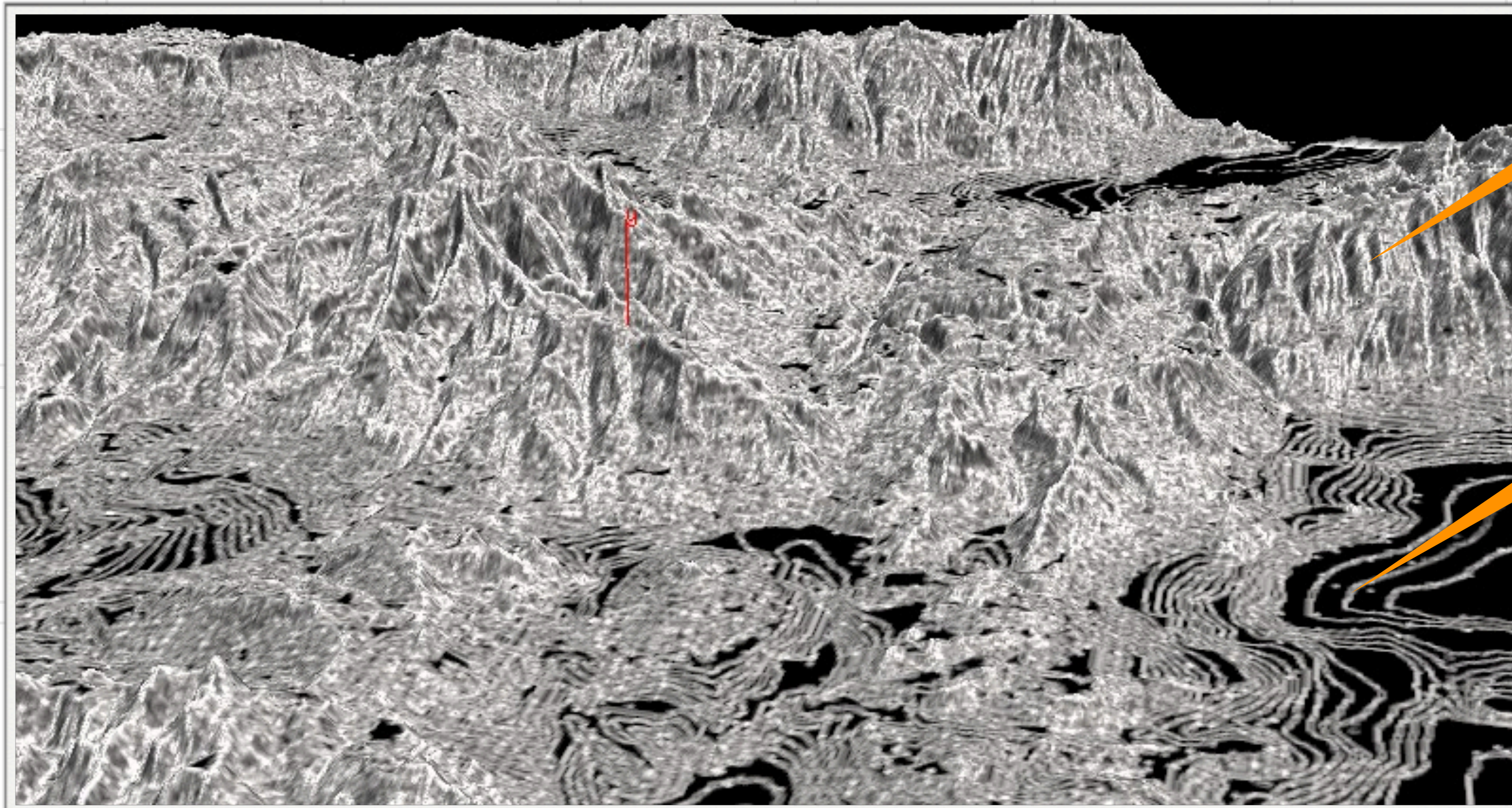
- ▶ Append a VOPSOP to the Curvature SOP
- ▶ Dive inside
 - ▶ Drop down a Bind VOP
 - ▶ name - curvature
 - ▶ Append a Fit Range VOP
 - ▶ Promote Source Min and Source Max
 - ▶ Destination Min - 0
 - ▶ Destination Max - 0
 - ▶ Append a Bind Export to the Fit Range
 - ▶ name - curvature
- ▶ Wire the Bind Export to the Cd of the Global Outputs



We are doing this because we want to control the range of curvature from 0 to 1

Analyzing the Curvature (cont.)

- ▶ Open up the Details View and notice the min and max values of Curvature
- ▶ Enter the min value into the Min and Max Value parameters.



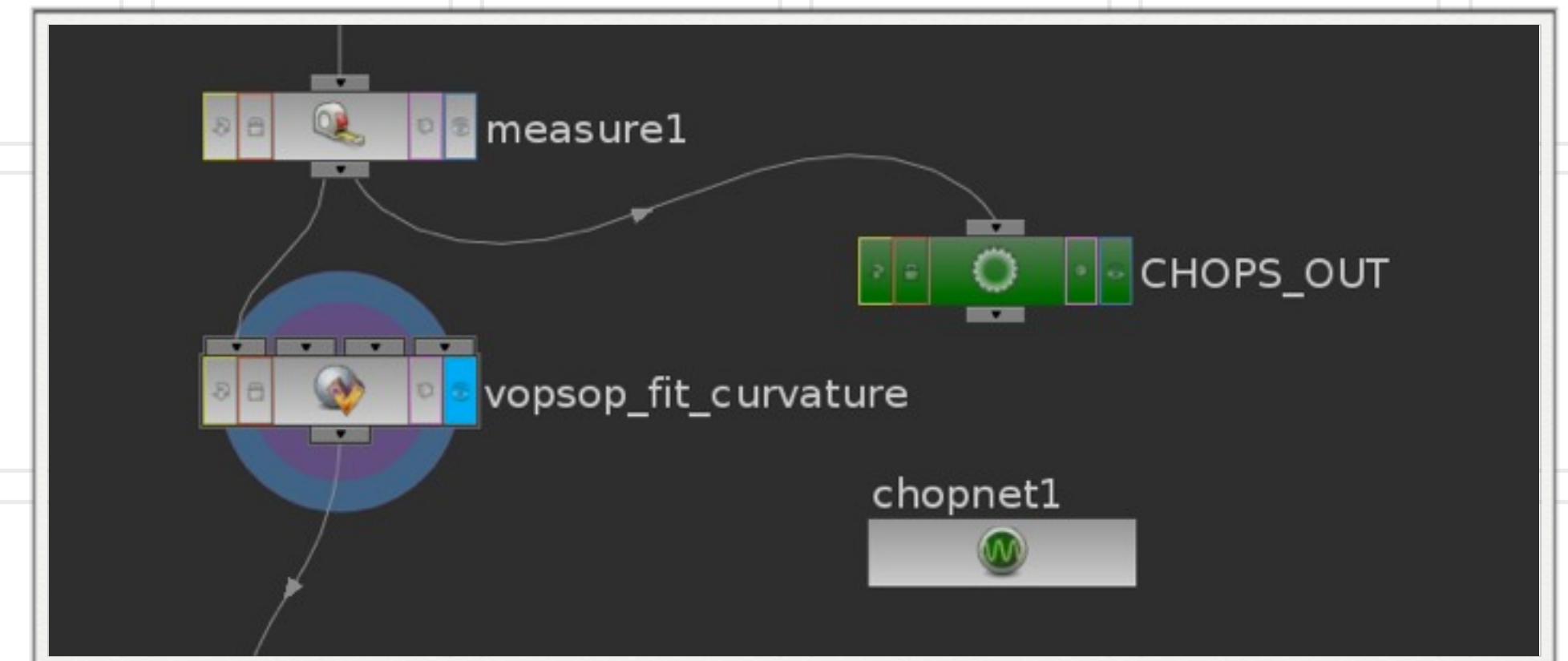
High Curvature
have a Cd that is
white

Flat areas have a
Cd that is black

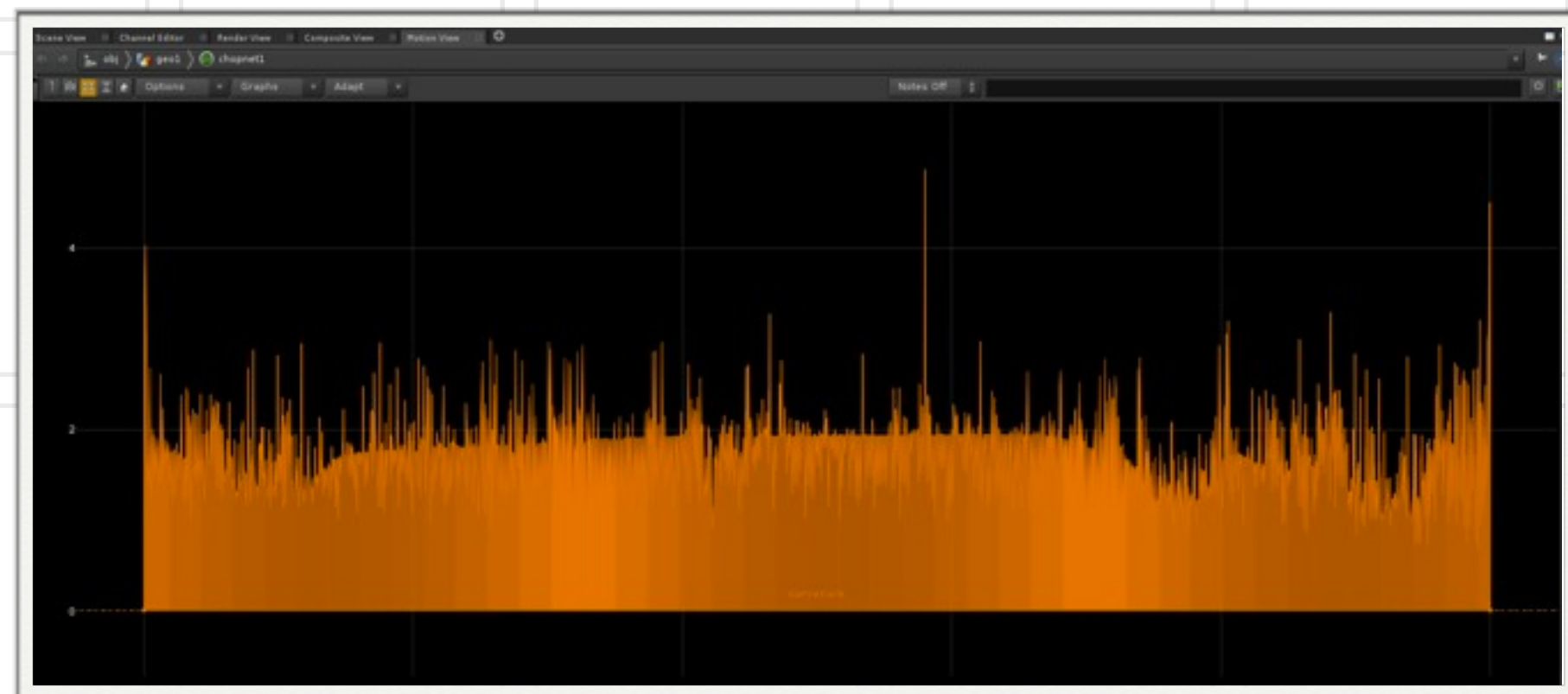
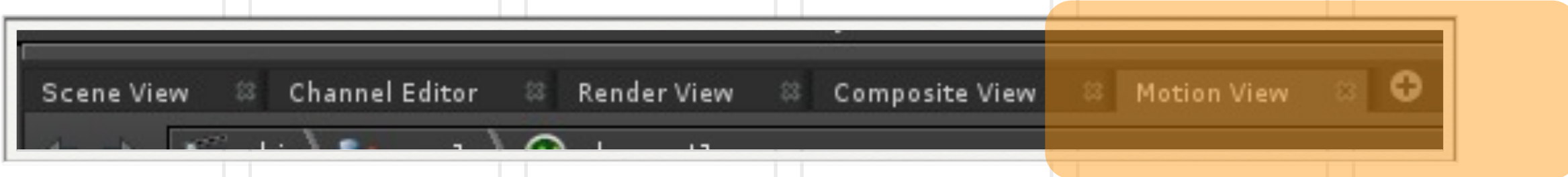
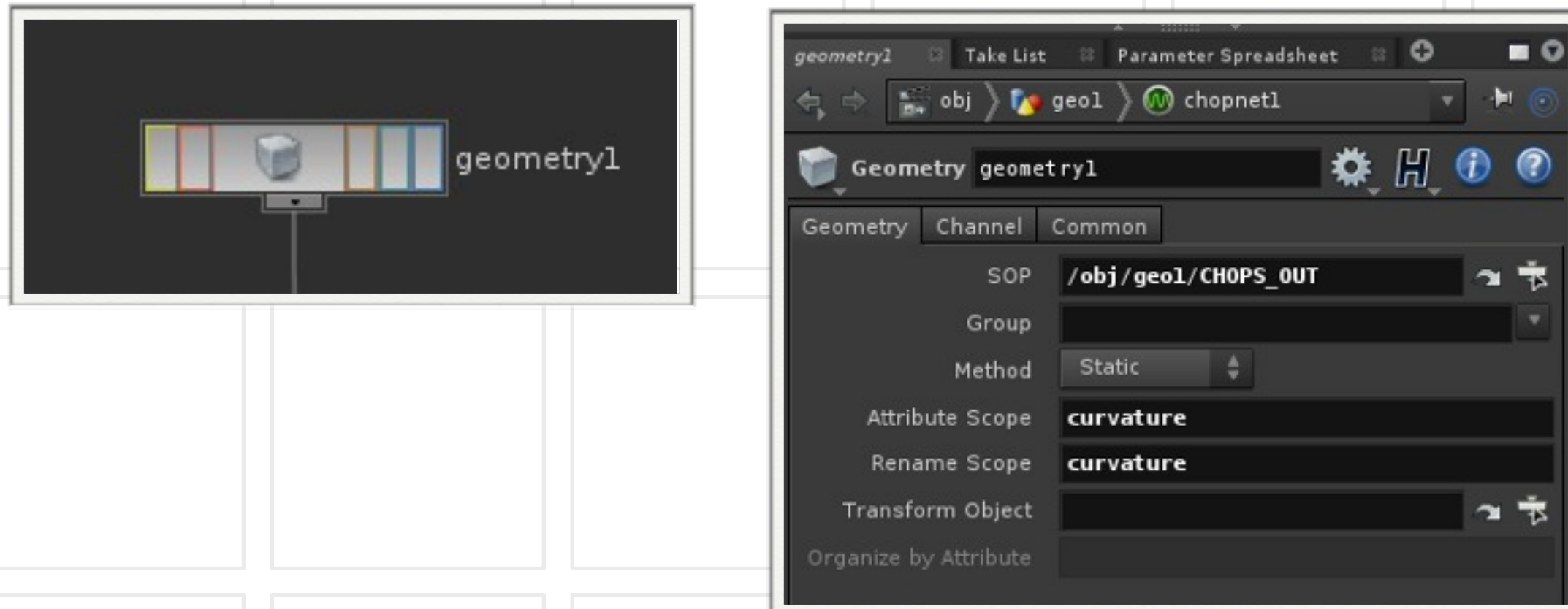
**SIDE EFFECTS
SOFTWARE**

Automating Min and Max Values

- ▶ It would be a real pain to have to go into the details view every time you change a height map to calculate the new min and max values for curvature
- ▶ We can use CHOPs to automate this process
- ▶ After the Measure SOP append a NULL as its own separate branch
 - ▶ name - CHOPS_OUT
- ▶ Drop down a CHOPNET

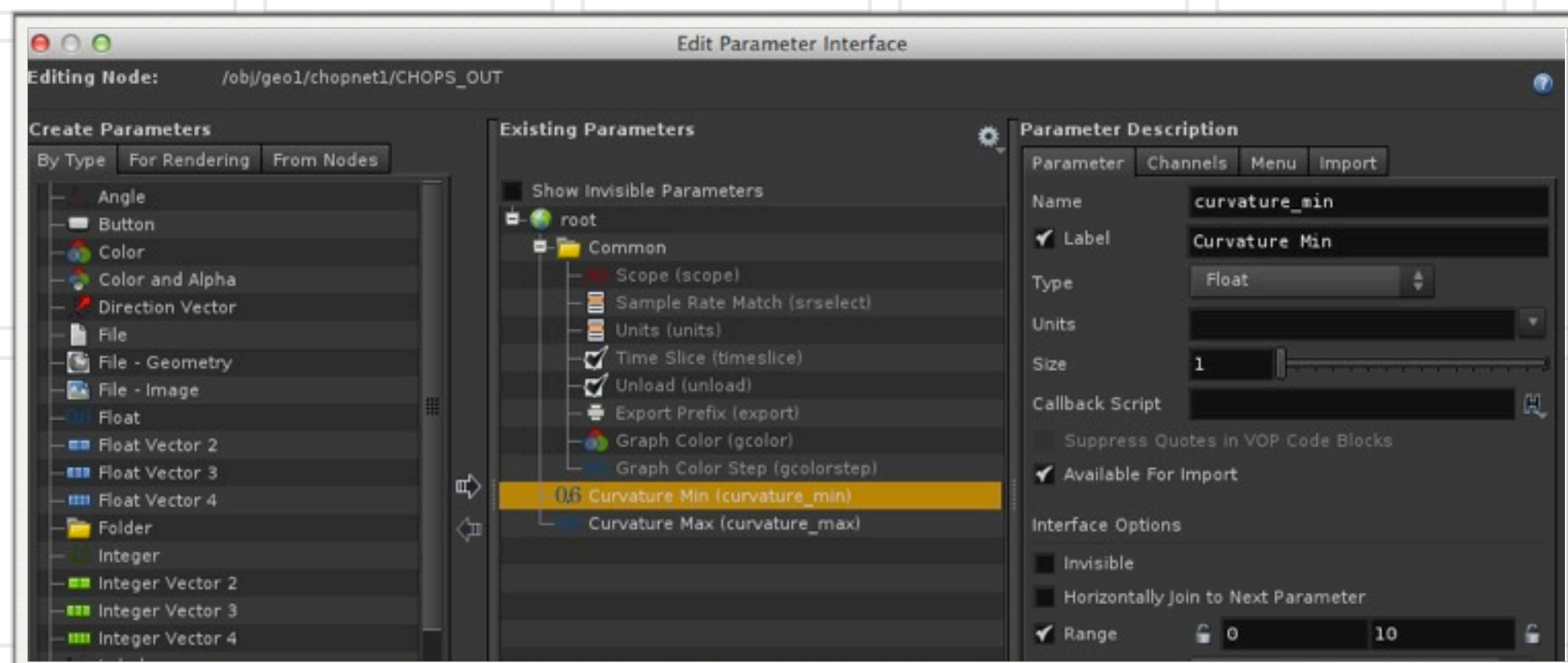
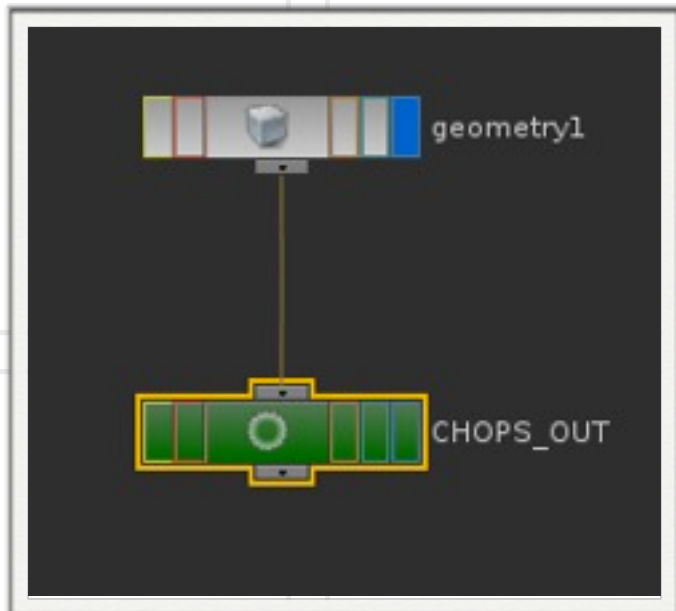


Automating Min and Max Values (cont.)



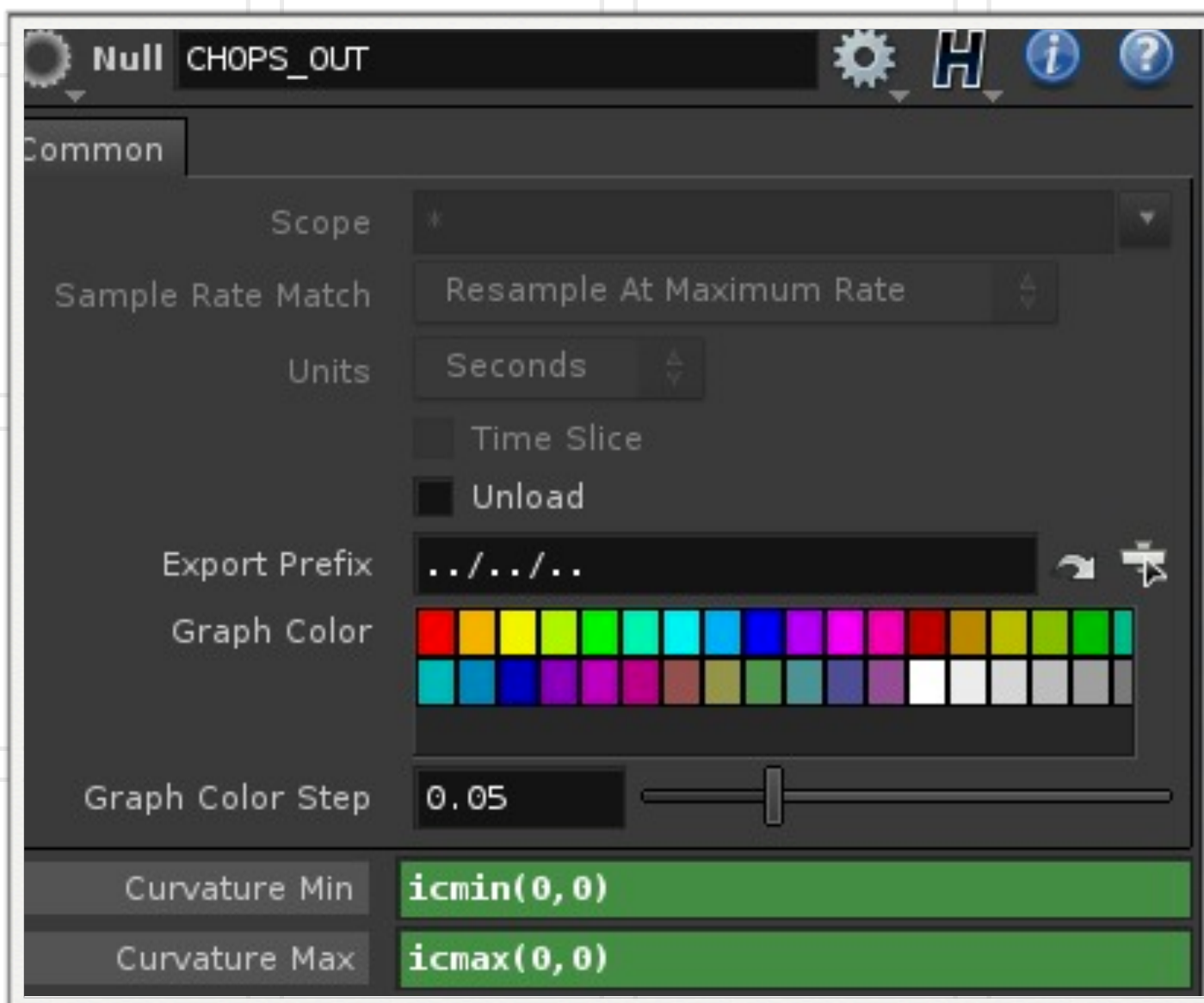
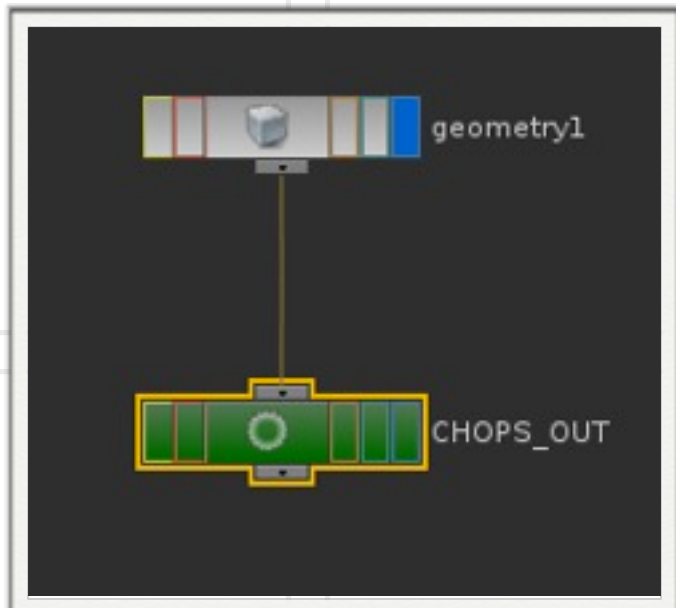
- ▶ Dive inside the CHOPNET
- ▶ Drop down a Geometry CHOP
 - ▶ SOP - /obj/geo1/CHOPS_OUT
 - ▶ Method Static
 - ▶ Attribute Scope - curvature
 - ▶ Rename Scope - curvature
- ▶ Select the Motion View and see the range of curvature

Automating Min and Max Values (cont.)



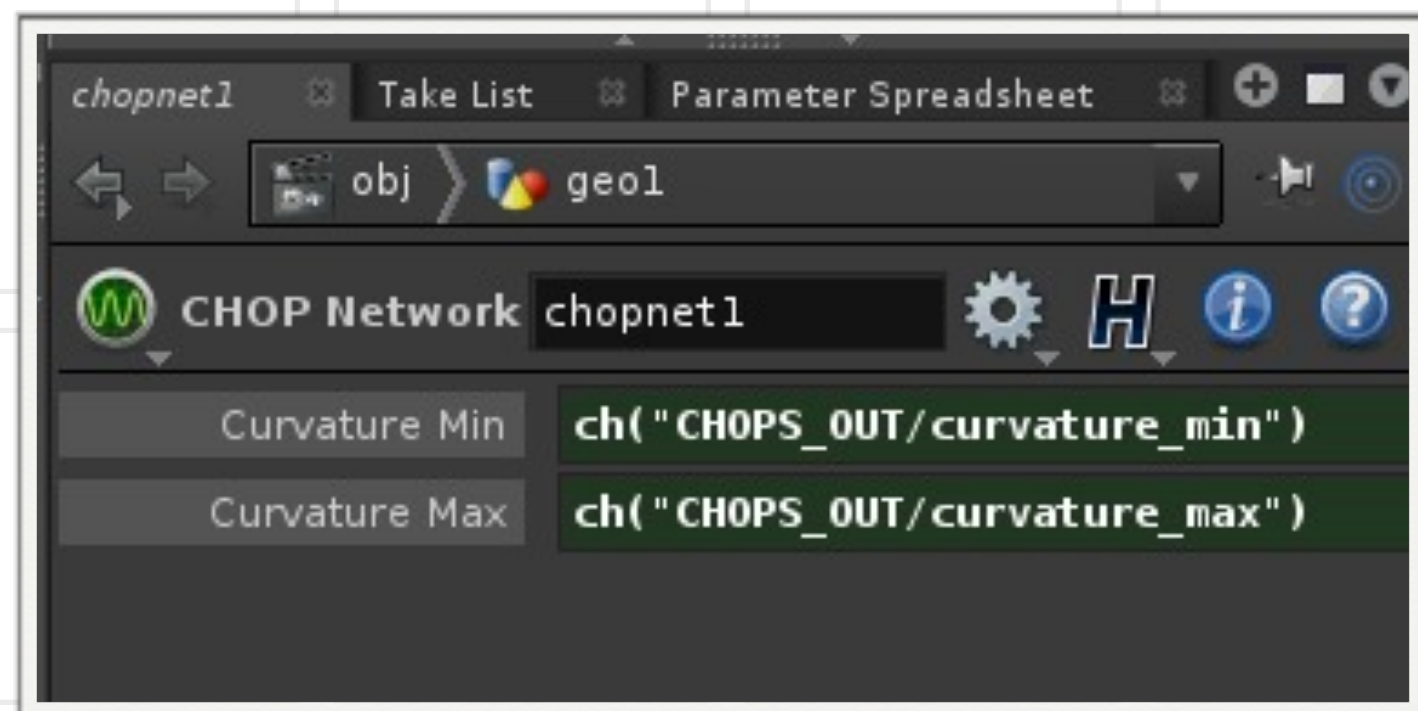
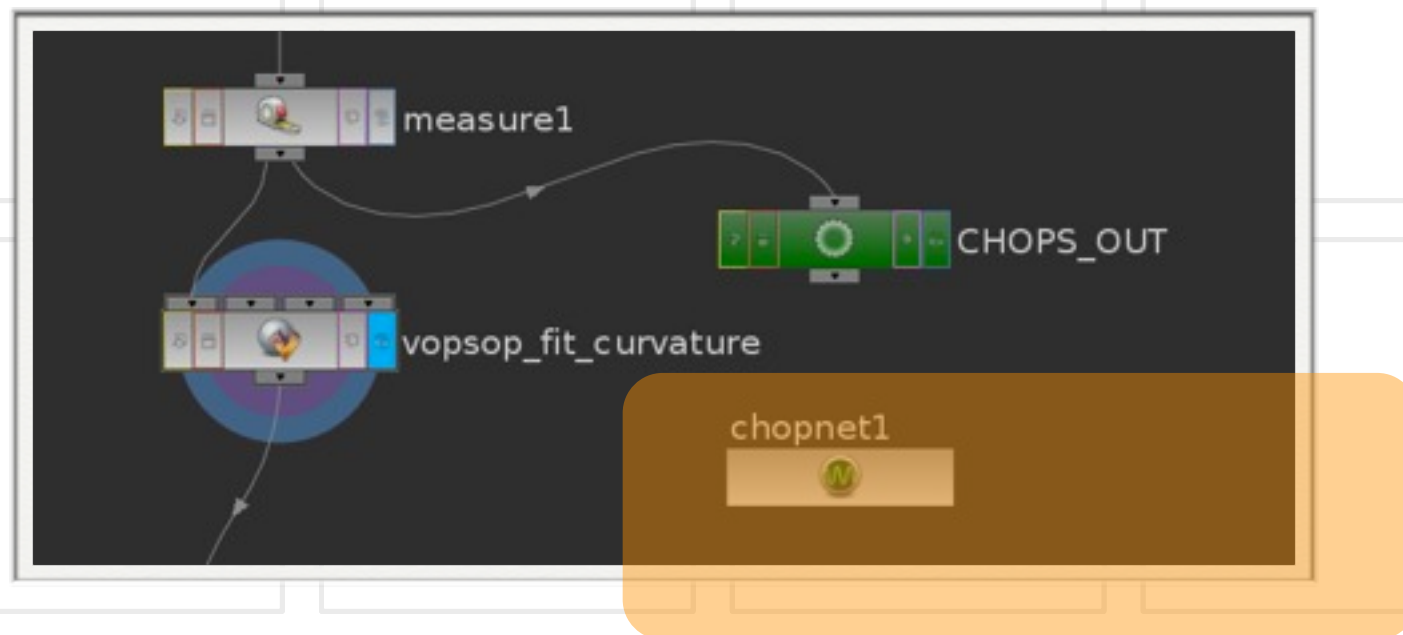
- ▶ Append a NULL to the Geometry CHOP
 - ▶ name - CHOPS_OUT
- ▶ Select “Edit Parameter Interface”
- ▶ We are going to create two float parameters for the minimum curvature and maximum curvature
 - ▶ First float
 - ▶ name - curvature_min, label Curvature Min
 - ▶ Second float - curvature_max, label - Curvature Max
- ▶ Click Accept

Automating Min and Max Values (cont.)



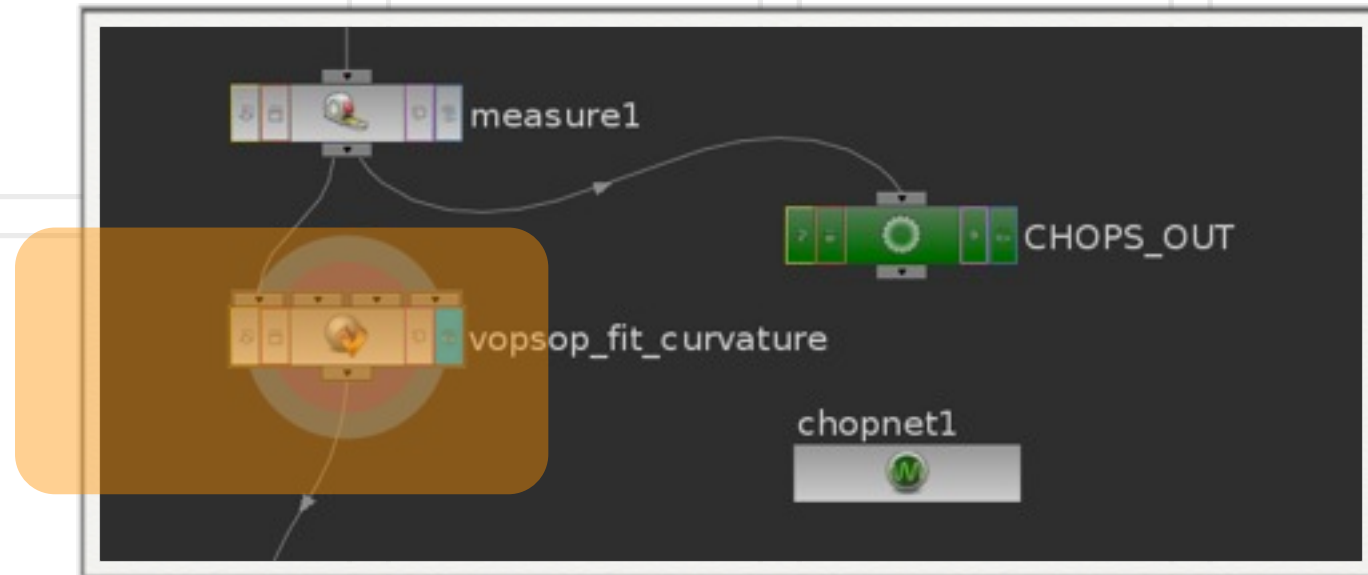
- ▶ In the parameters for the CHOPS OUT Node we are going to use the expressions icmin, and icmax to calculate the min and max values of the channel
- ▶ float icmin (float input_index, float channel_index)
- ▶ Evaluates a CHOP's input channel's minimum value
- ▶ Curvature Min - icmin(0,0)
- ▶ Curvature Max - icmax(0,0)
- ▶ Jump back up to the Geometry level

Automating Min and Max Values (cont.)

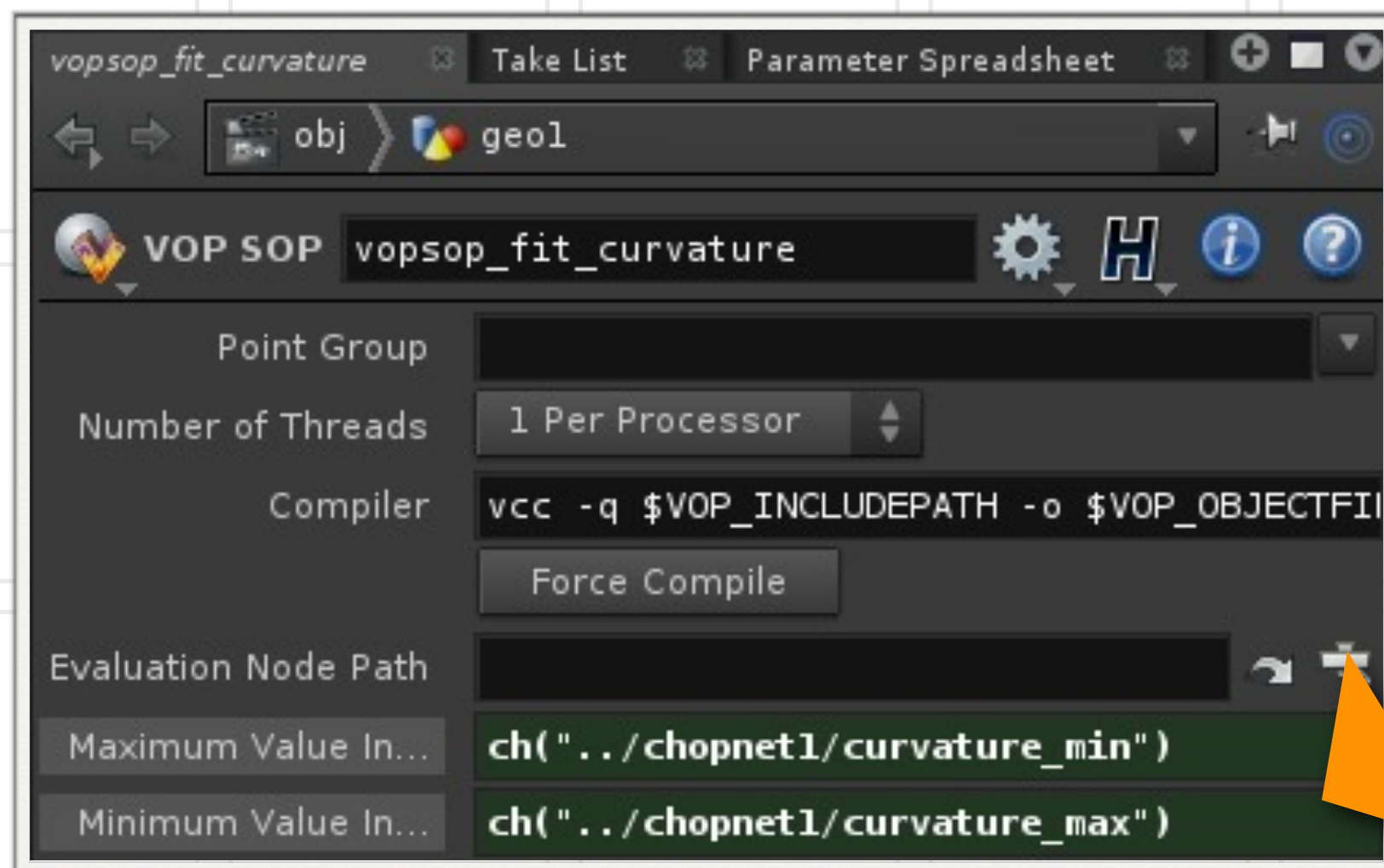


- ▶ Selecting the CHOPNET
- ▶ Select “Edit Parameter Interface”
 - ▶ We are going to promote the two values Curvature Min and Max to the Geometry level
- ▶ Create two floats like we just did at the CHOPs level but now at the Geometry level
- ▶ Click Accept
- ▶ Curvature Min - `ch("CHOPS_OUT/curvature_min")`
- ▶ Curvature Max - `ch("CHOPS_OUT/curvature_max")`

Automating Min and Max Values (cont.)



- ▶ All that remains now is two create the expression in the VOPSOP to read the CHOPs min/max values instead of entering them manually





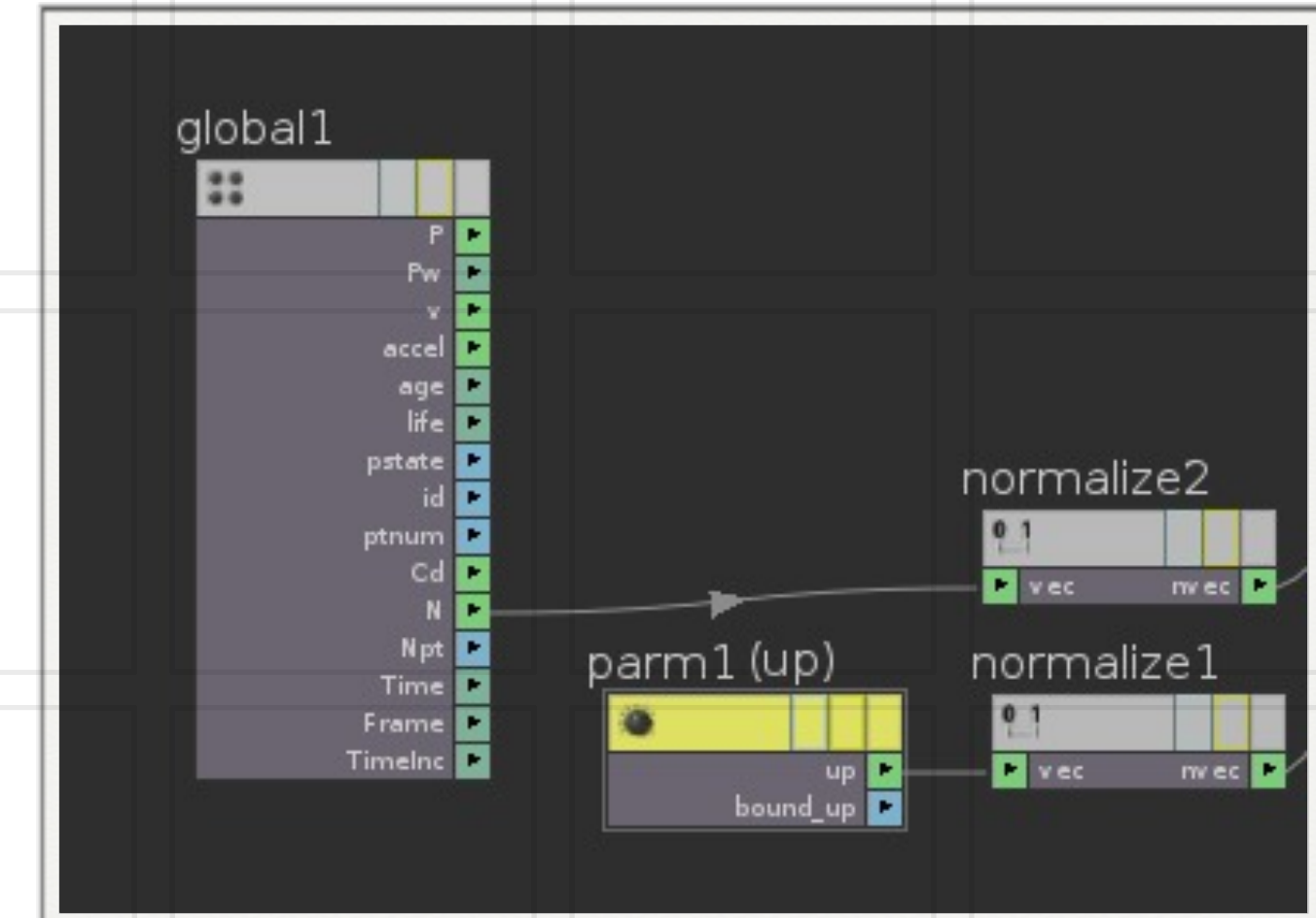
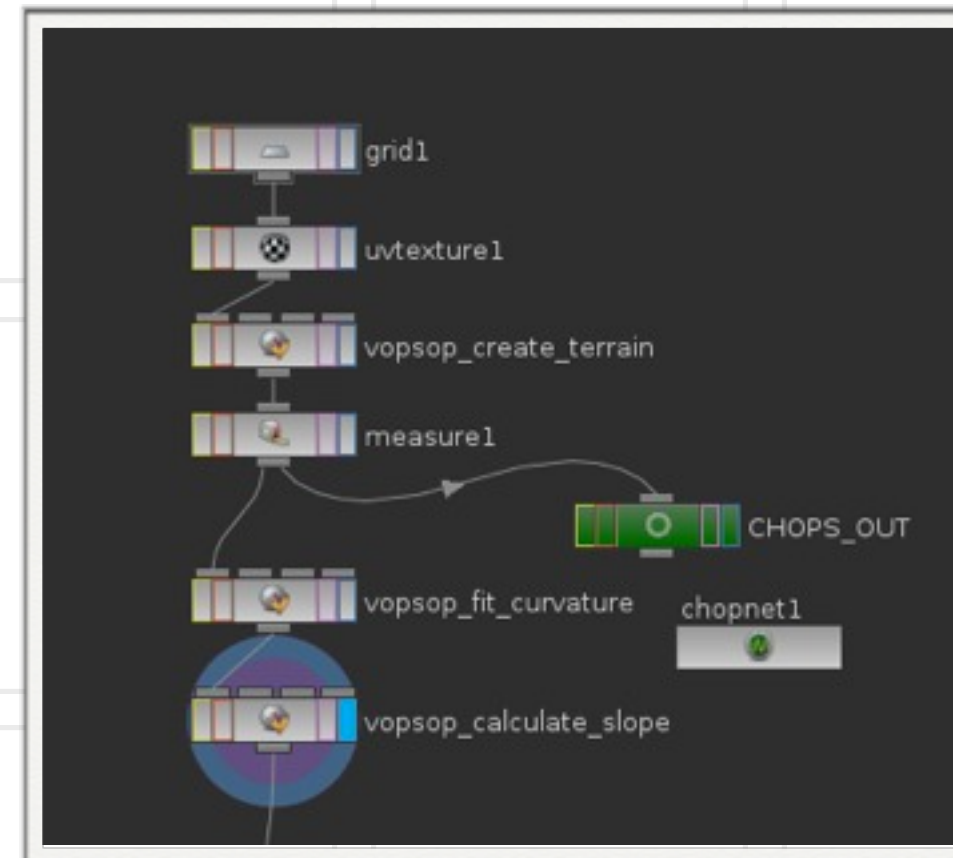
Calculating Slope and Peaks

Determining where grass can grow and snow can settle

**SIDE EFFECTS
SOFTWARE**

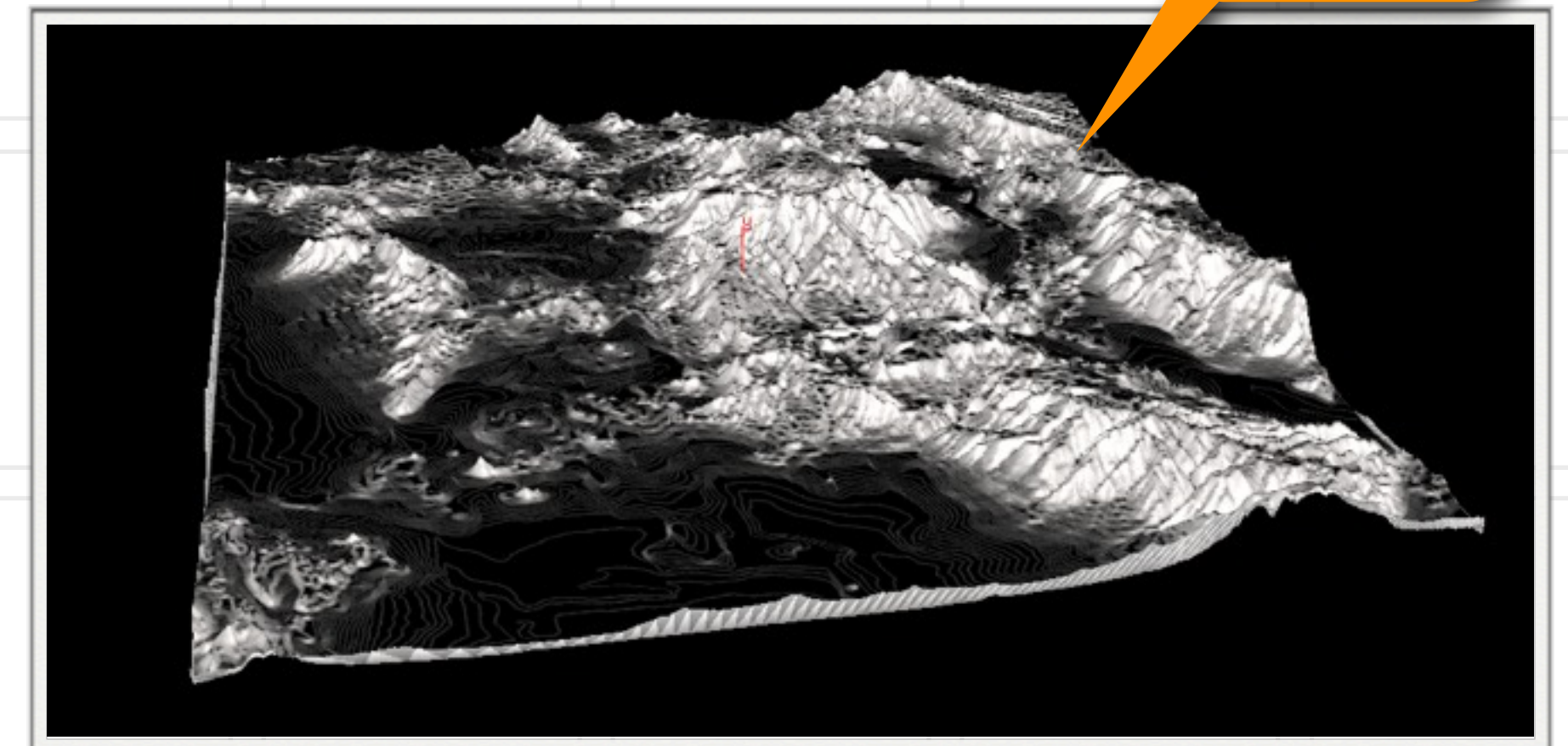
Calculating Slope

- ▶ Append another VOPSOP after the Curvature VOPSOP
- ▶ Dive inside. To calculate the slope all we need to know is the Surface Normals and a Up Vector
- ▶ Drop Down a Parameter VOP
 - ▶ name - up
 - ▶ label - up
 - ▶ Type - Vector
 - ▶ Vector Default - 0,1,0 (keep the parameter visible so the artist can override the value)
 - ▶ Append the Normalize VOP to it <-- we want unit vectors
 - ▶ From the Surface Globals Append a Normalize VOP to the Normals Parameter



-
- The diagram illustrates a computational graph for a neural network layer. The graph consists of the following blocks and their connections:
- global1**: The initial input block on the left, containing a list of variables: P, Pw, v, accel, age, life, pstate, id, ptnum, Cd, H, Npt, Time, Frame, and TimeInc.
 - normalize2**: Receives input from **global1** and **parml (up)**. It contains a 'vec' field and an 'nvec' field.
 - parml (up)**: A parameter block containing 'up' and 'bound_up' fields.
 - normalize1**: Receives input from **parml (up)**. It contains a 'vec' field and an 'nvec' field.
 - dot1**: Receives input from **normalize2** and **normalize1**. It contains a 'vec1' field, a 'dot...' field, and a 'vec2' field.
 - complement1**: Receives input from **dot1**. It contains a 'val' field and a 'co...' field.
 - output1**: The final output block on the right, containing a list of variables: P, Pw, v, accel, age, life, pstate, id, Cd, and N.
- Arrows indicate the flow of data from left to right, showing the sequence of operations in the network layer.

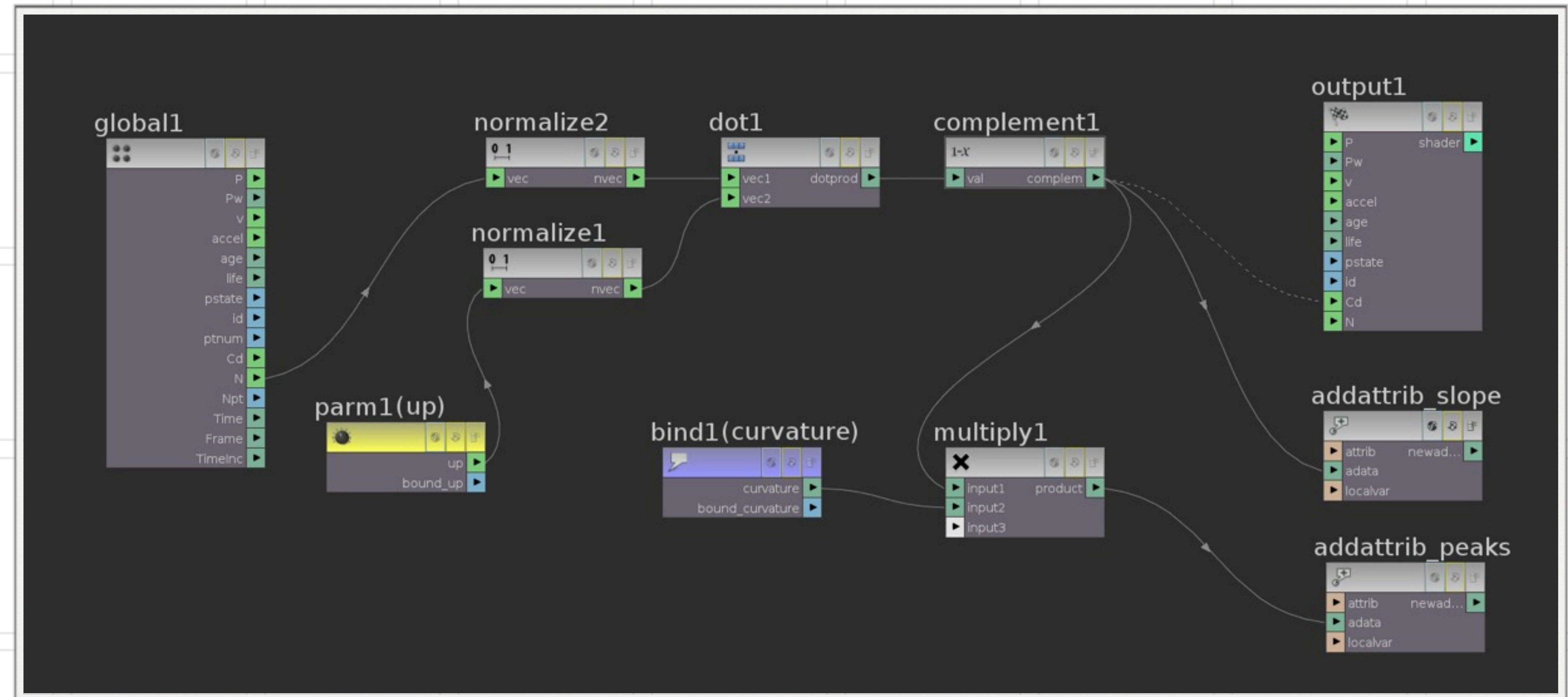
Correct



Thursday, April 25, 13

Calculating Peaks

- ▶ In addition to the slope let us calculate peaks. It might come in useful if you want the peaks be granite or covered in snow
- ▶ Peaks = Curvature times Slope





Creating the Grass Attribute

Where do I want to place grass?

**SIDE EFFECTS
SOFTWARE**

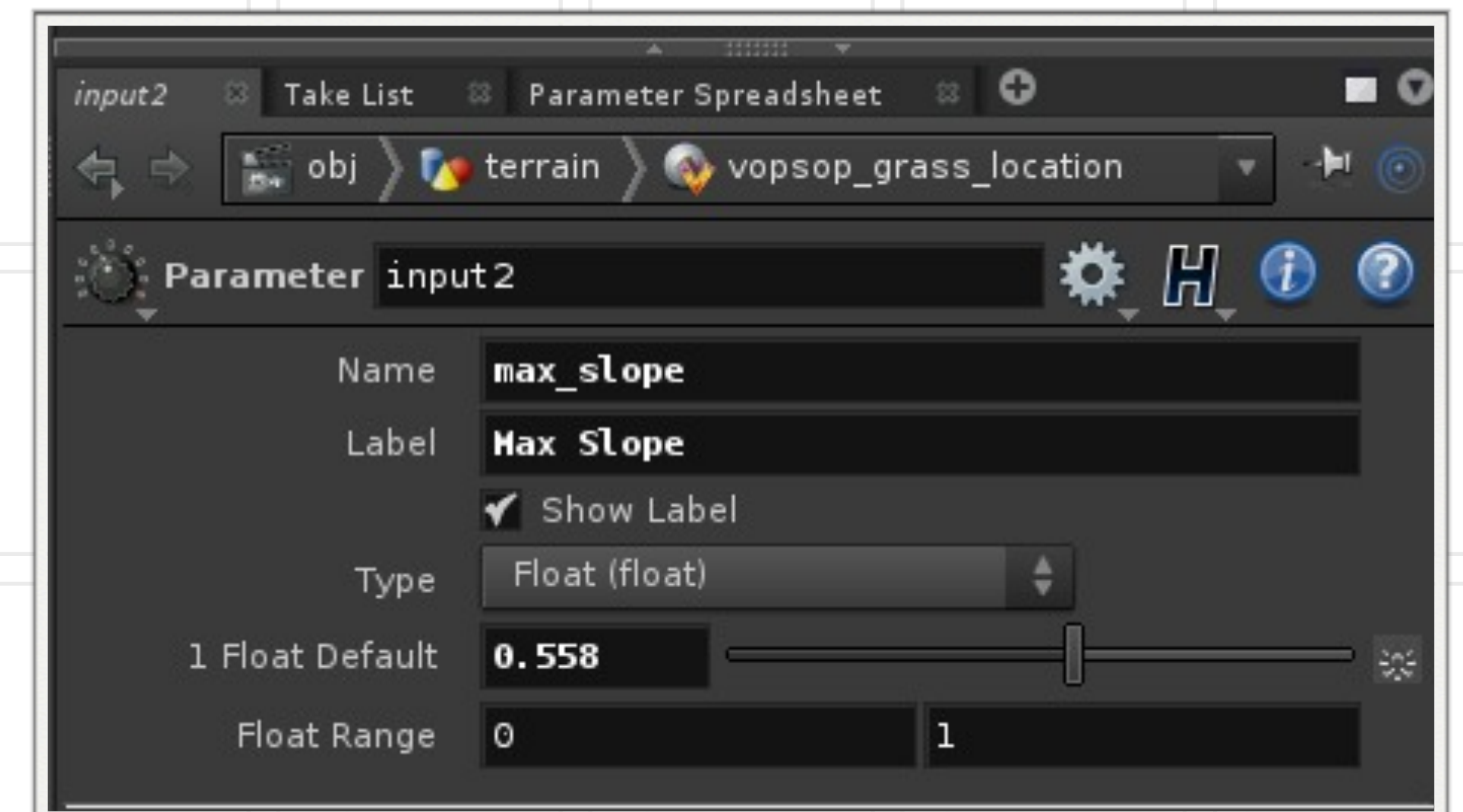
Considerations for the Grass Attribute

Where do I want to place grass?

- ▶ Where water accumulates - no steep slopes
- ▶ Above a certain minimum elevation
- ▶ Below a certain maximum elevation

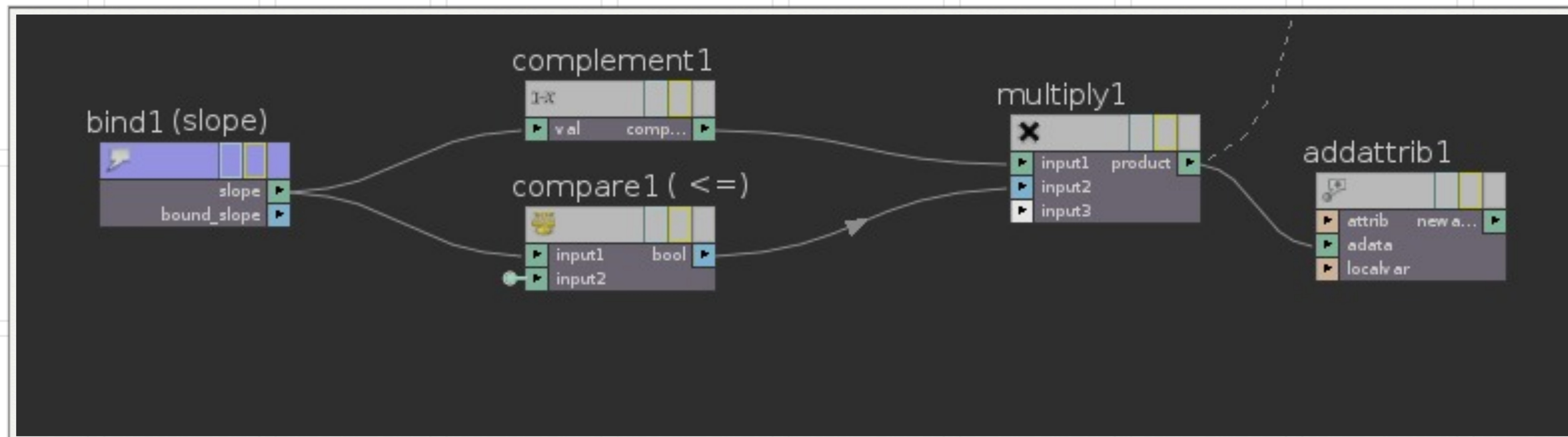
Placing Grass by Slope

- ▶ Drop down another VOPSOP and dive inside
- ▶ Drop down a Bind SOP with name “slope”
- ▶ Append a Compare VOP to the Bind
 - ▶ Test - Less then or Equal to
 - ▶ Promote the second input
 - ▶ name - max slope
 - ▶ label - Max Slope
 - ▶ default - 0.5



Placing Grass by Slope

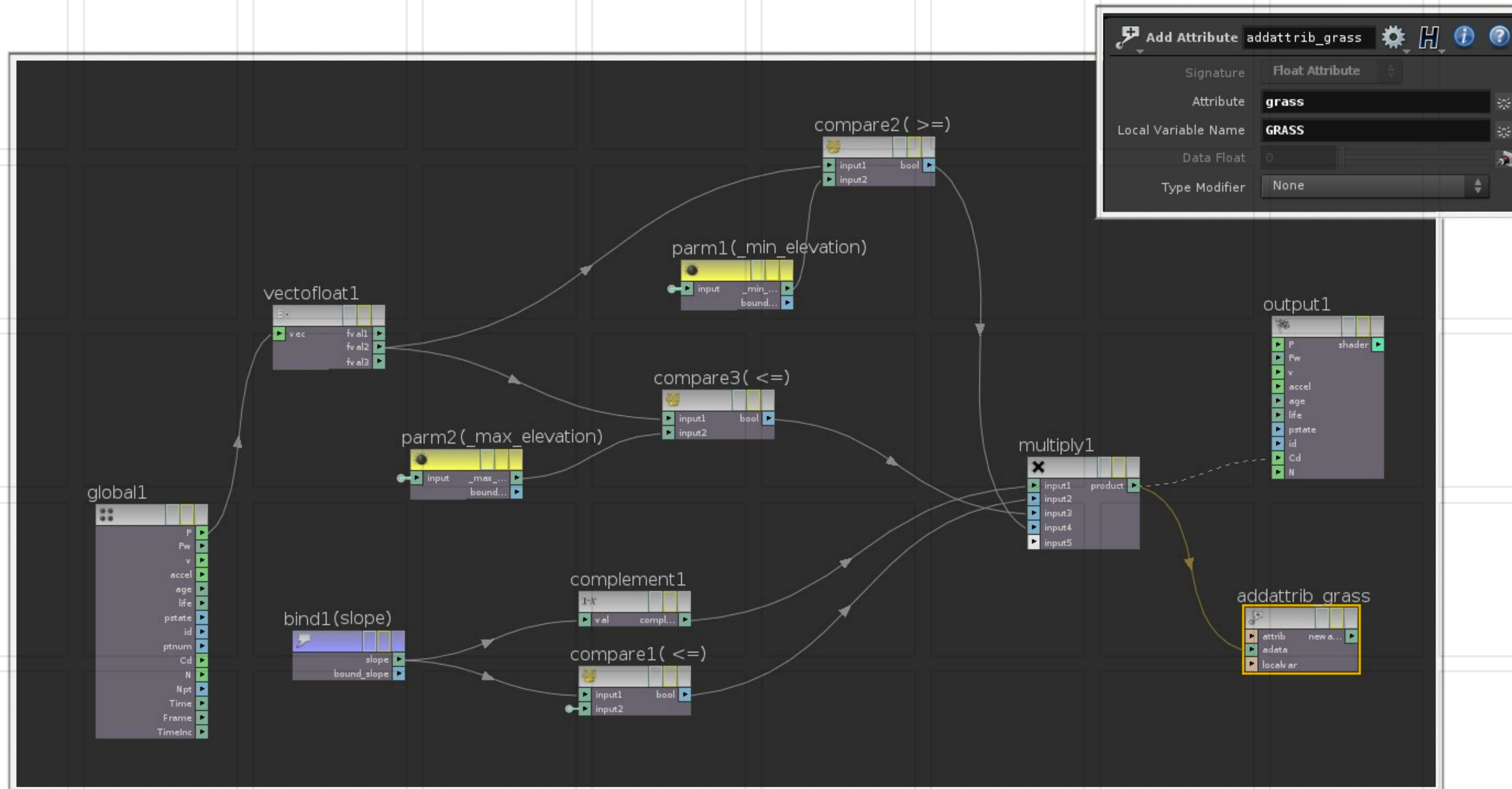
- ▶ Now the output of the compare will be a 0 or 1. 0 if the slope is too great, 1 if the slope is right for grass
- ▶ We want to multiply the value by the slope to get less grass as the slope increases. But we want the complement of the slope because grass should be more dense in flatter terrains



Placing Grass by Elevation

- ▶ Now lets do similar operation to create a range where grass can grow
- ▶ Drop down a Vector to Float VOP and wire the Global Position to it (We just want the y-component)
- ▶ To the y-channel of the float to vector append to Compare VOPs. One we will test the minimum elevation with a \geq test the other a maximum elevation with a \leq test
- ▶ Now we just multiply them all together
- ▶ Finally append to MULTIPLY VOP an ADDATTRIBUTE
 - ▶ name - grass
- ▶ Network image on next page...

Grass Attribute Network





Assignment at Home

Create the Networks for Snow and Granite

**SIDE EFFECTS
SOFTWARE**



Exporting a Grass Map for Games

Creating a grayscale map to export to a game engine

**SIDE EFFECTS
SOFTWARE**

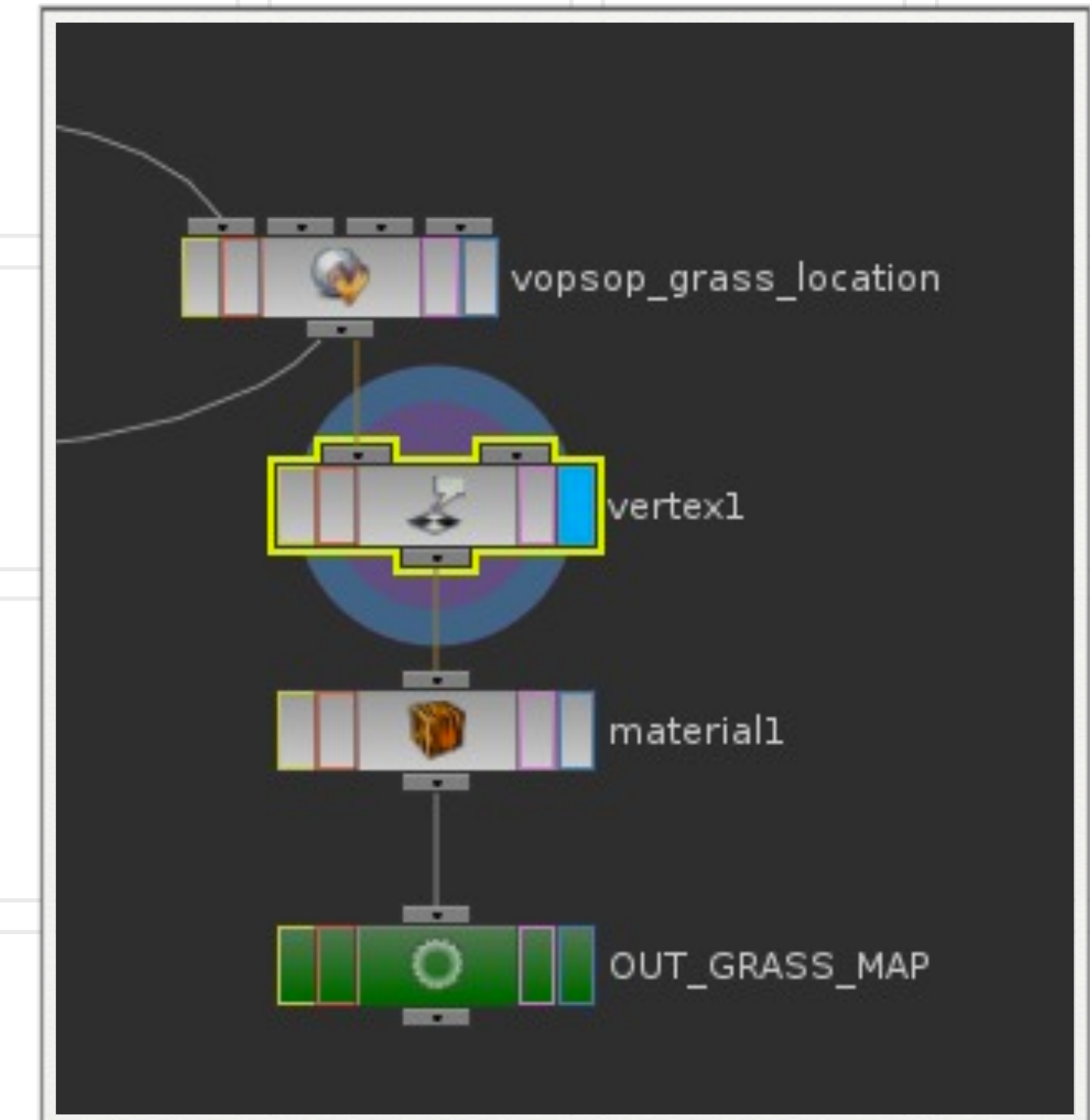
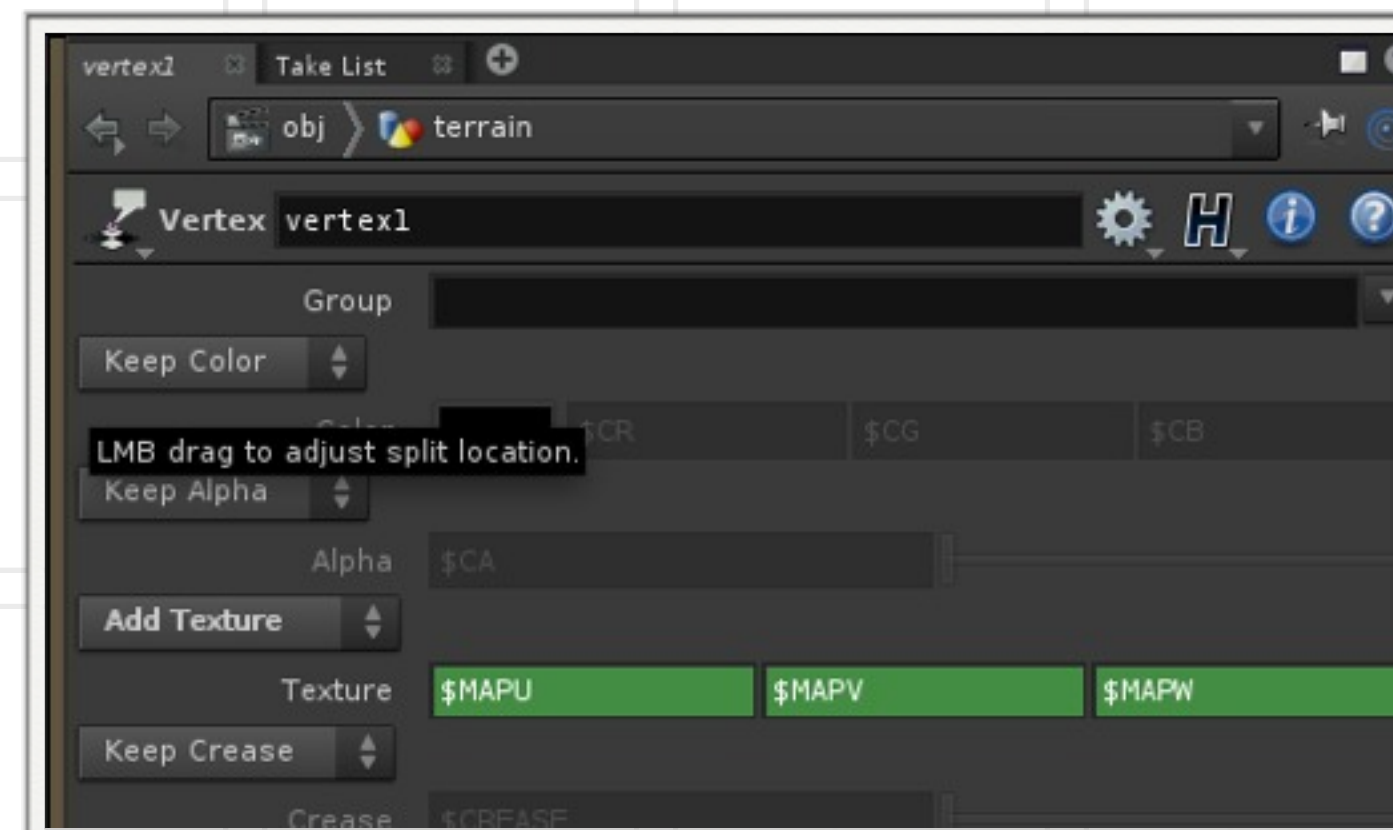
Grass Map Considerations

The Density of the terrain is too much for a game engine

It would be great to be able to export a uv map of the grass distribution

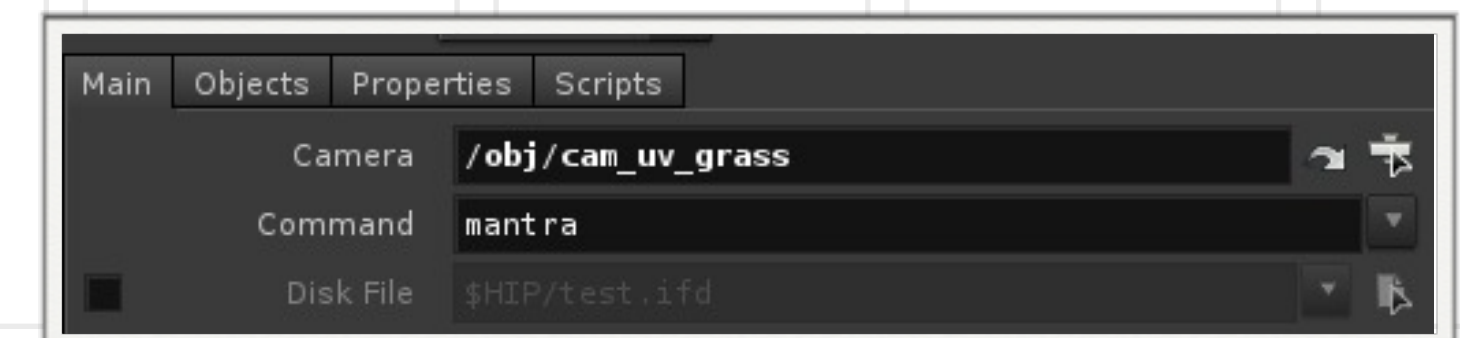
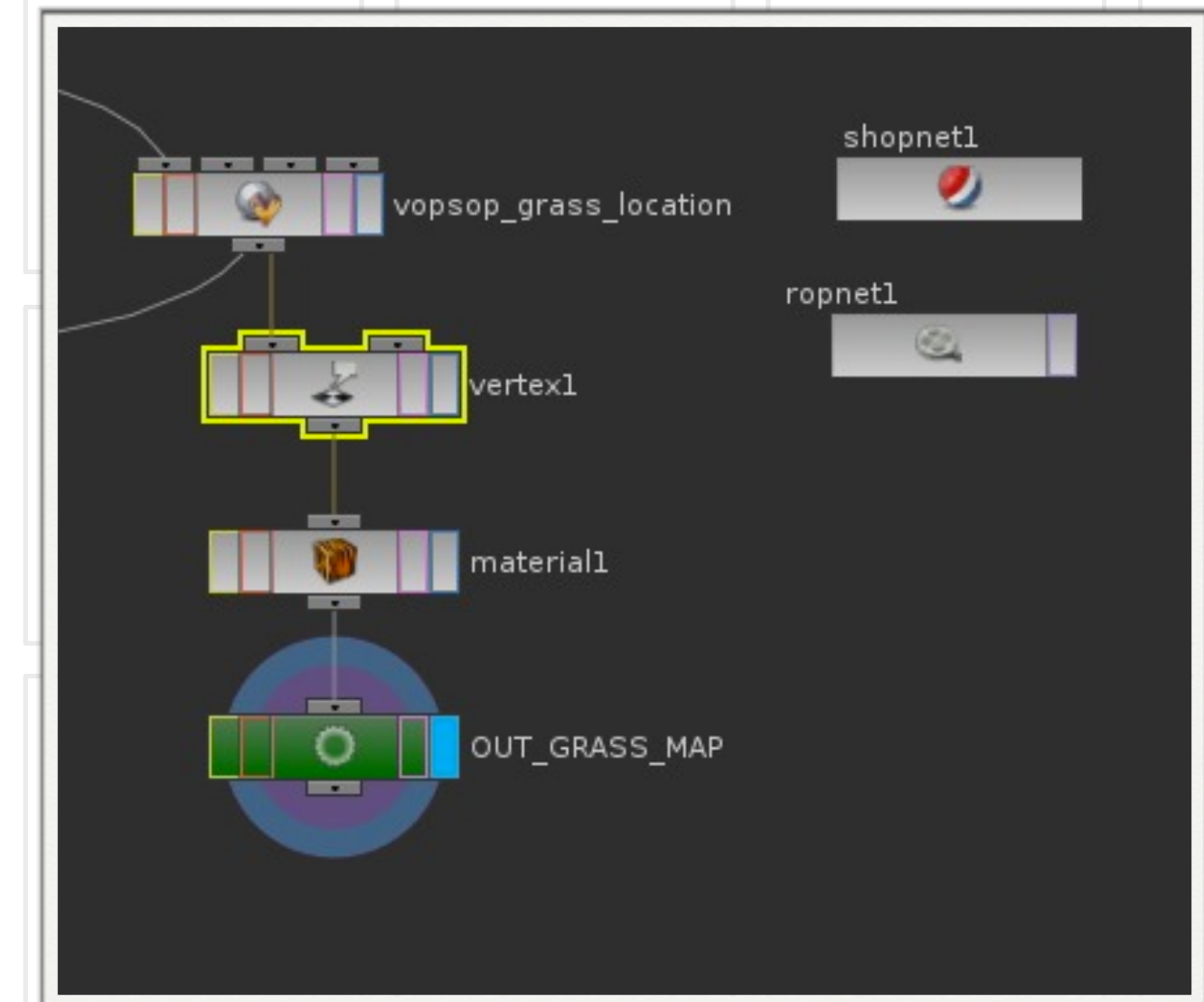
Creating a Vertex Based UV Map for Export

- ▶ Append a VERTEX SOP to the grass location vopsop
 - ▶ Add Texture
- ▶ Append a Material SOP to the Vertex SOP
 - ▶ You will need to link the Material to a Constant Shader
- ▶ Append a NULL
 - ▶ name - OUT_GRASS_MAP



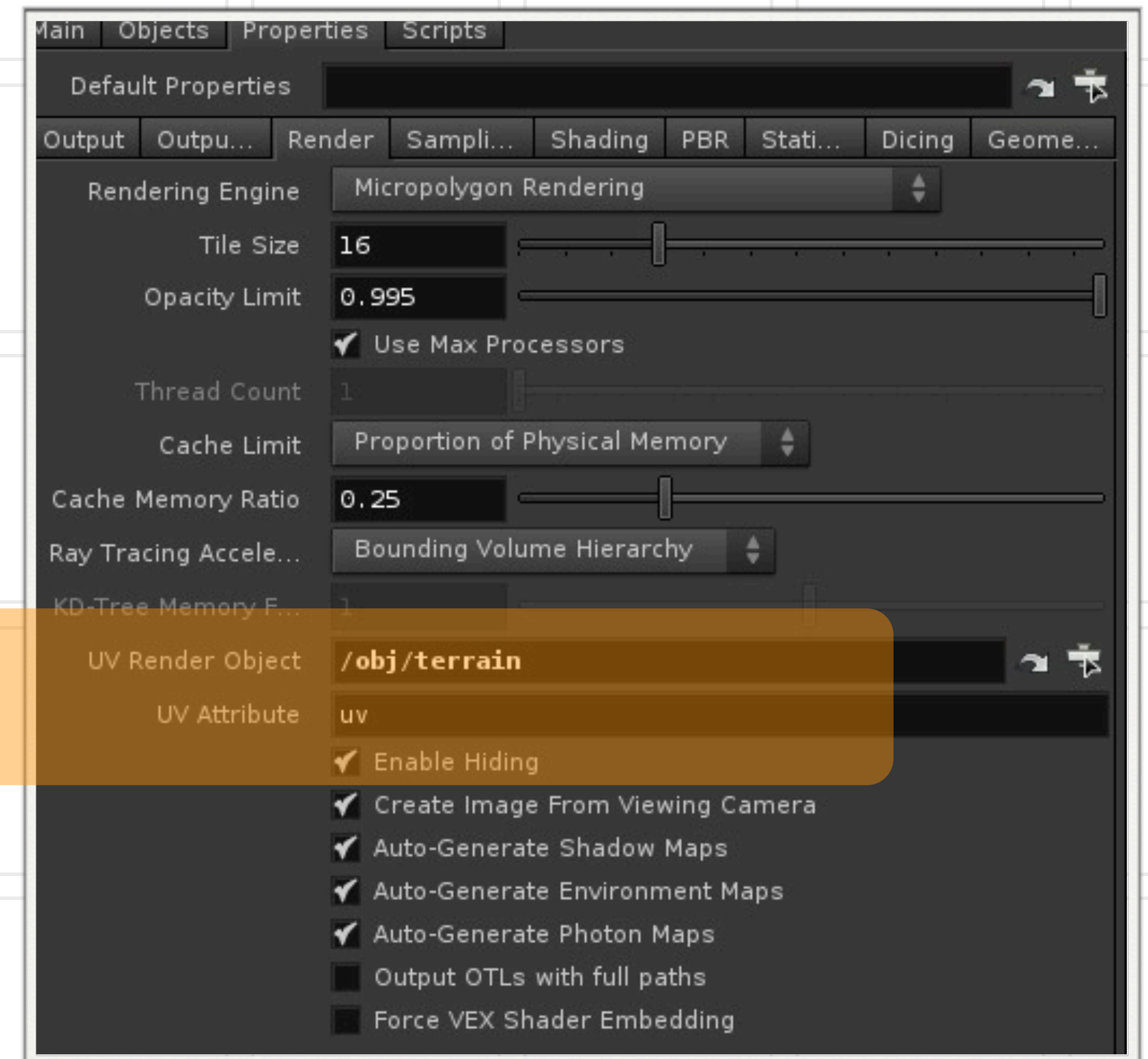
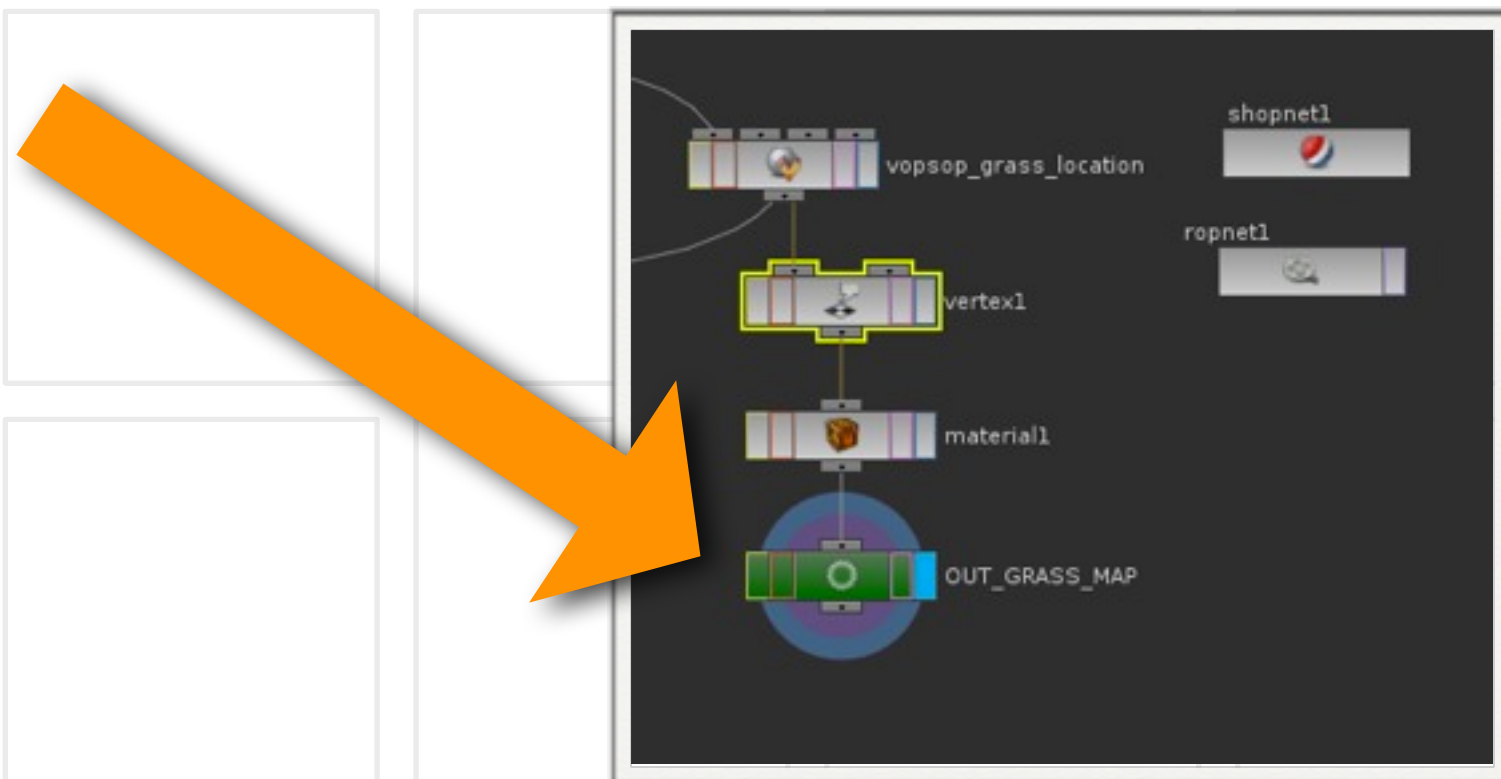
Creating a Vertex Based UV Map for Export (cont.)

- ▶ Now we need a SHOPNET to place a Constant Shader and a ROPNET for our Mantra Node
- ▶ Drop down a SHOPNET and place a Constant Shader Inside
- ▶ Drop down a ROPNET and dive inside
 - ▶ Drop down a Mantra Node
 - ▶ Any camera will do
- ▶ Images shown on next page

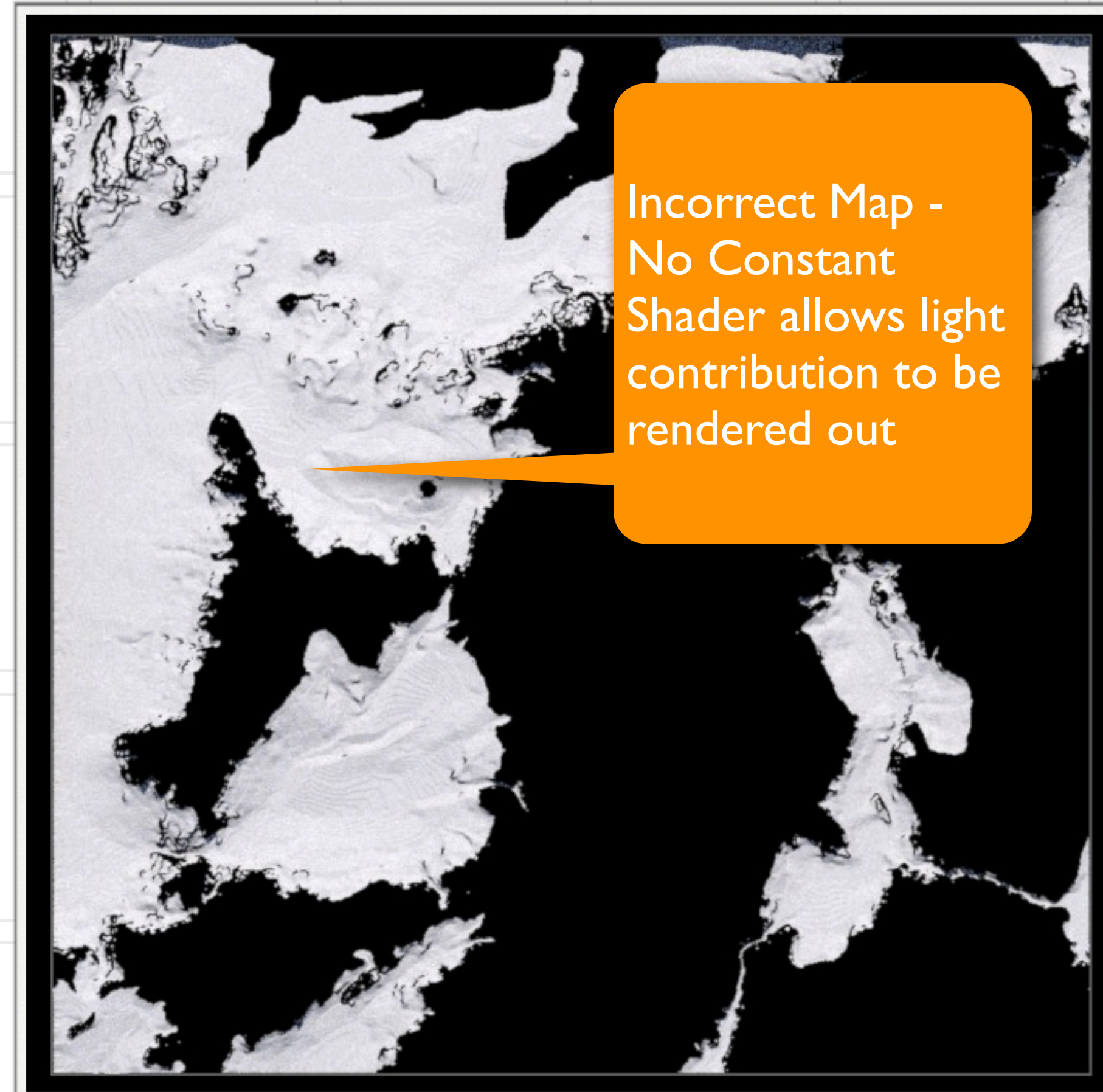
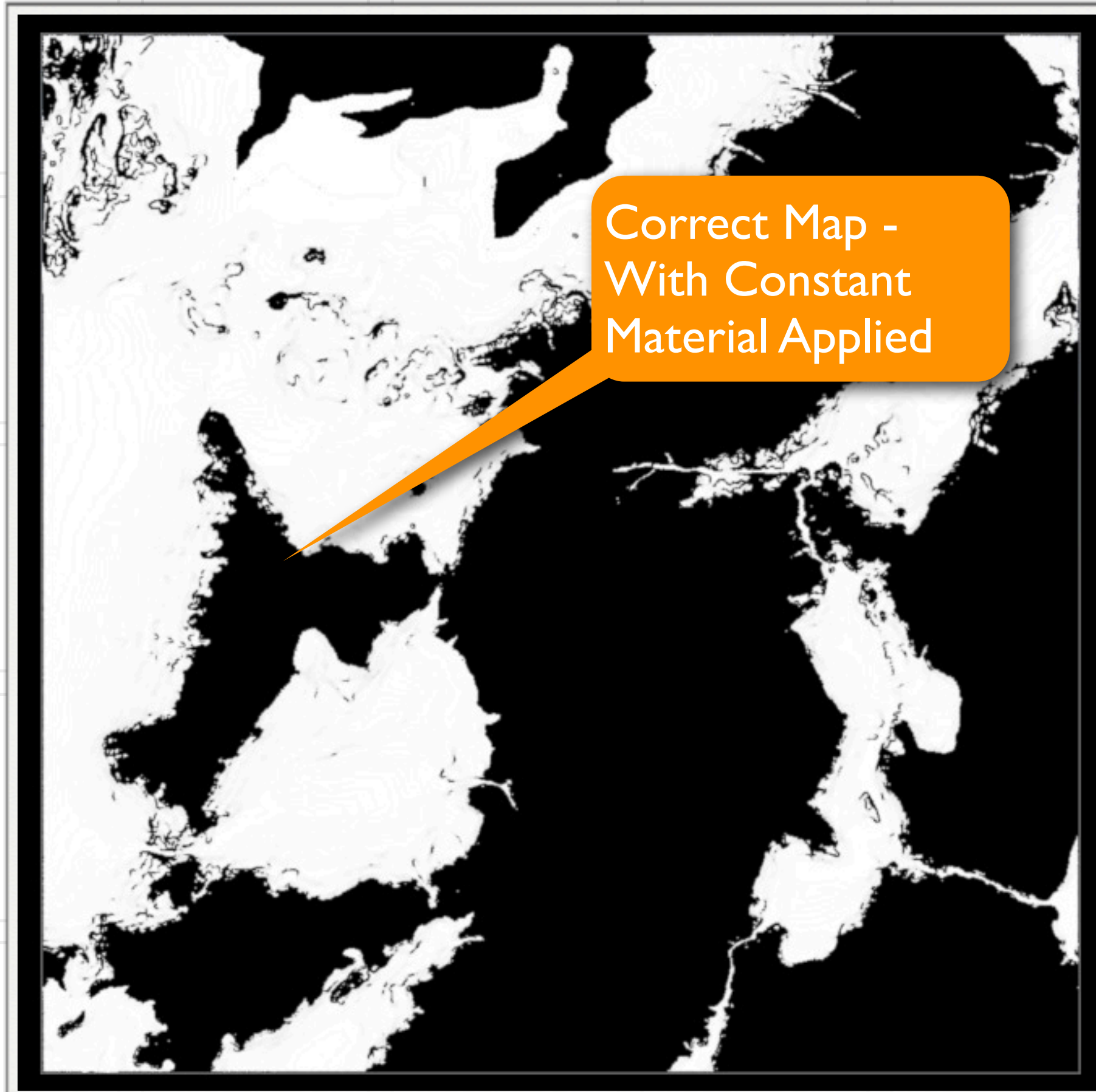


Creating a Vertex Based UV Map for Export (cont.)

- ▶ In the Render tab you must enter the object you want to uv render.
- ▶ **IMPORTANT** - In the object you want to render the render flag must be set on the NULL after the VERTEX texture Map

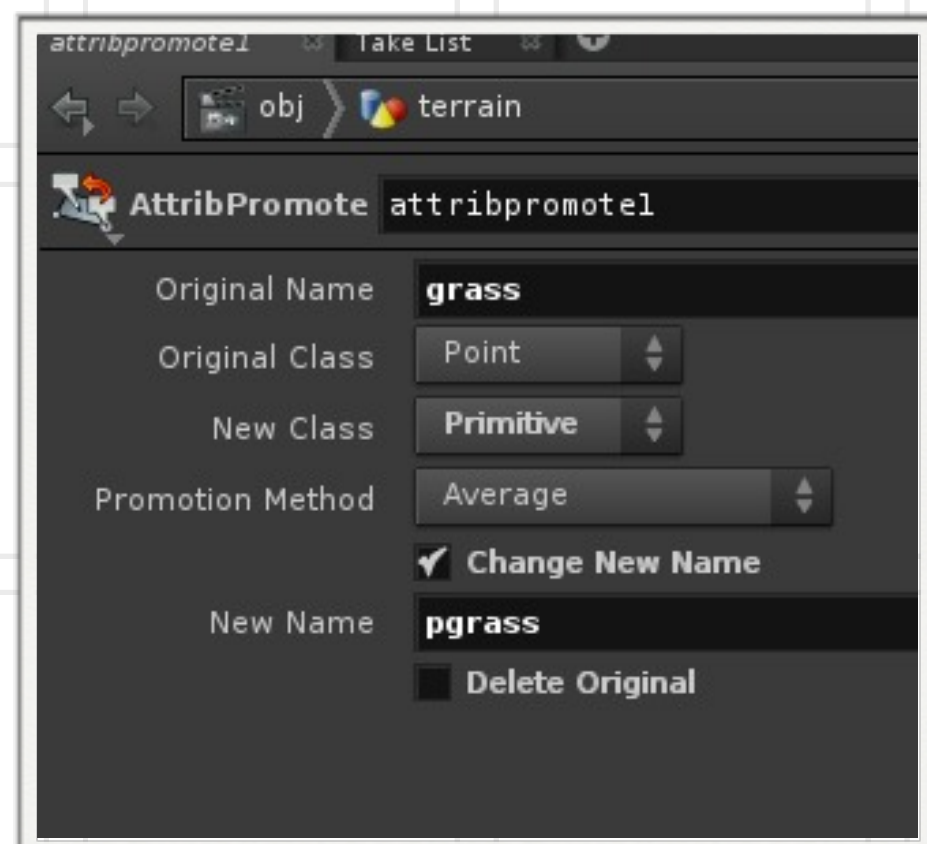
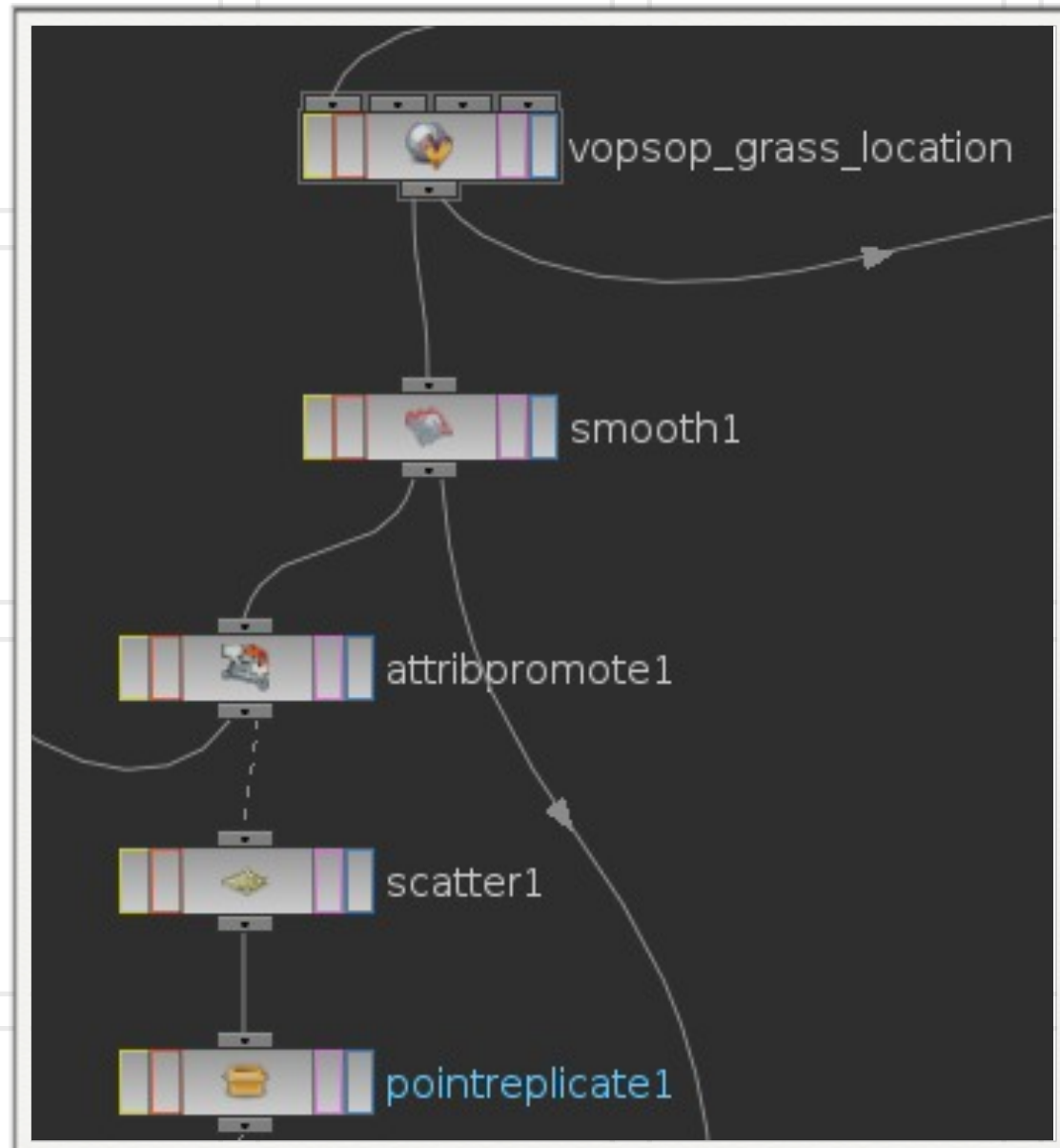


Final Grass Map



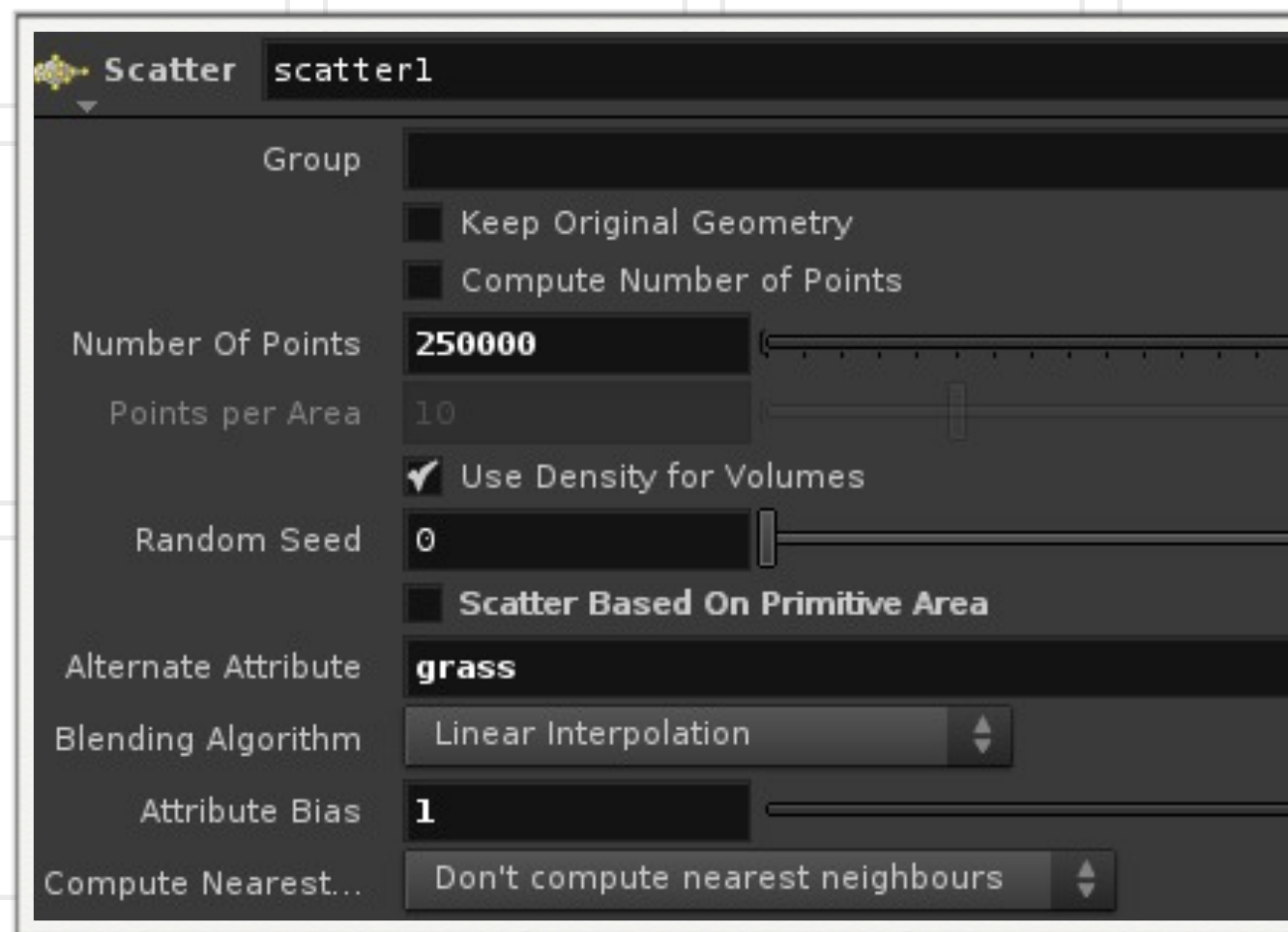
**SIDE EFFECTS
SOFTWARE**

Scatter Points Based on Grass Attribute



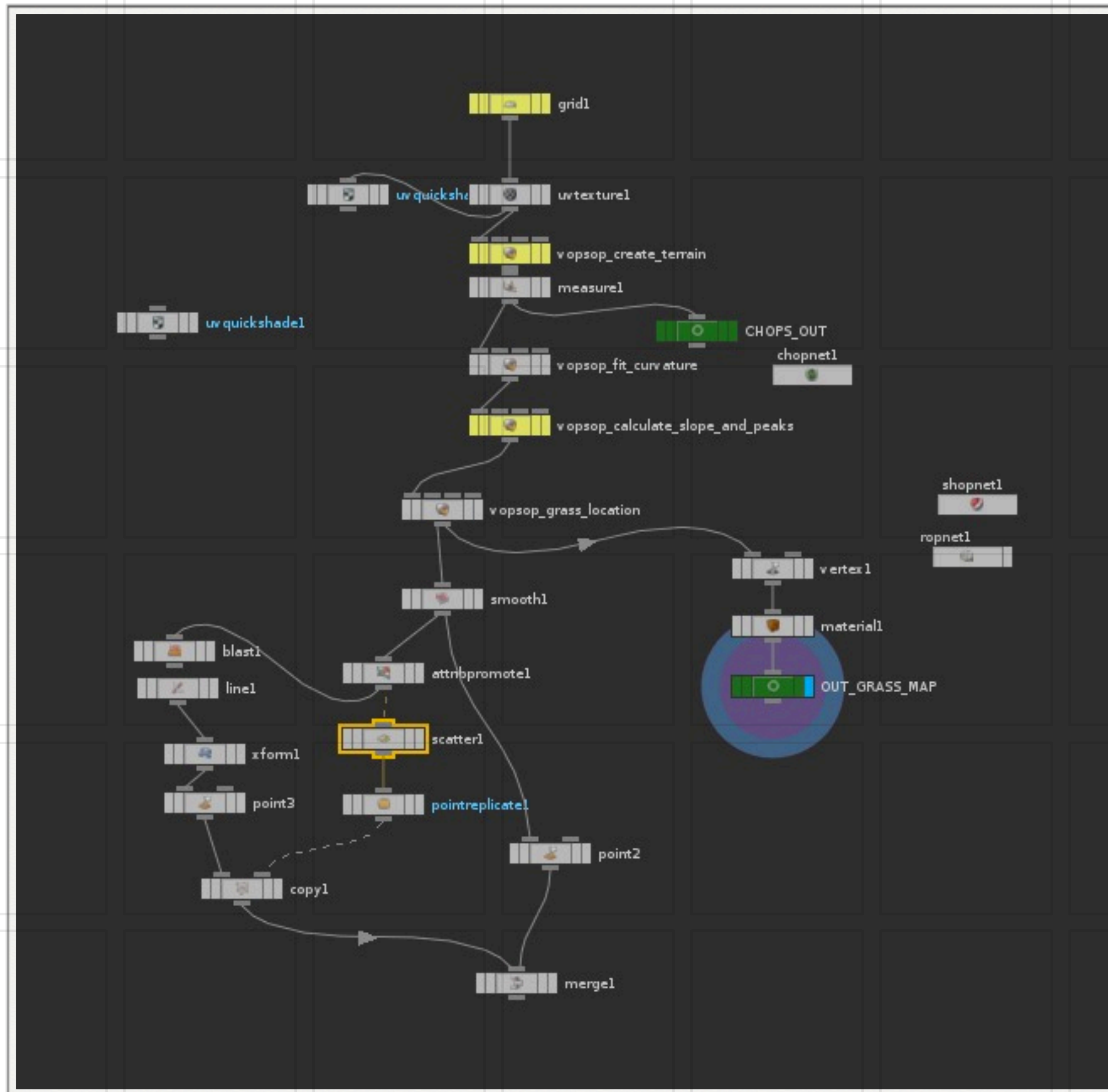
- ▶ Append a SMOOTH SOP to the VOPSOP
 - ▶ You can smooth attributes
 - ▶ apply to - Other Attribute
 - ▶ attribute name - grass
- ▶ We do not want to work with points so
 - ▶ Append an Attribute Promote
 - ▶ original name - grass
 - ▶ original class - point
 - ▶ new class - primitive

Scatter Points Based on Grass Attribute (cont.)



- ▶ Append a Scatter SOP
 - ▶ Number of Points - 250000
 - ▶ Turn off “Scatter Based On Primitive Area”
 - ▶ Alternate Attribute - grass
- ▶ If you want more points in clusters try using the point replicate SOP

Final Network



Of course you can use
the Grass OTL we
created a few weeks
ago :)



End Module 07

**SIDE EFFECTS
SOFTWARE**