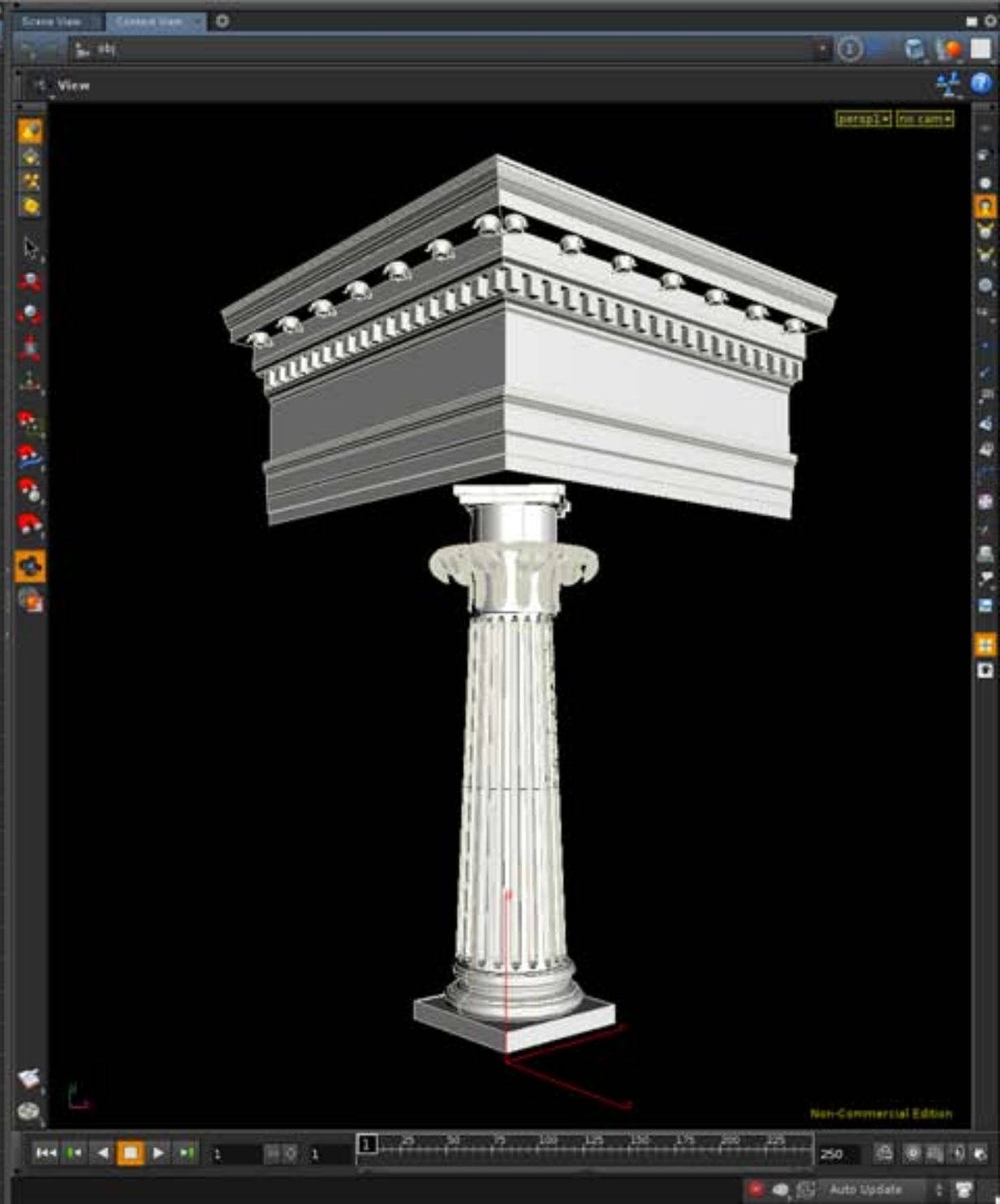
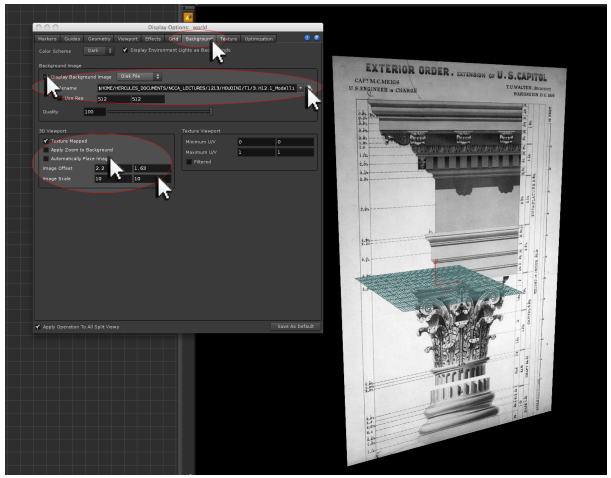
MY_TEMPLE


plinth
column
column_base
column_headpiece


View
persp1
no cam
Non-Commercial Edition

MODELLING A TEMPLE

In a **new Houdini scene**, activate the **Display Options** for the **Viewer** (**d with the mouse over the Viewer**), and from the **Background** section of the Display Options load in the **Corinth_Schematic.jpg** image as the reference image to model to.

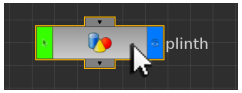


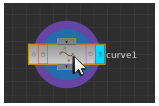
In the **3D Viewport** subsection of the Background Display Options, **specify**:

■	**Automatically Place Image**	
Image Offset	**2.2**	**1.63**
Image Scale	**10**	**10**

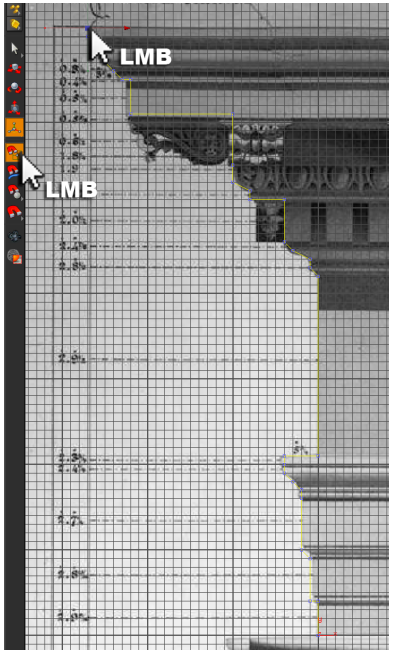
This will align the temple plinth area of the image with the Origin Point of the scene.

Close down the display options, and in the **Network Editor** create a new **Geometry Object** and rename it to **plinth**. Go inside this object, and **delete** the default **File SOP**. In its place create a **Curve SOP**.





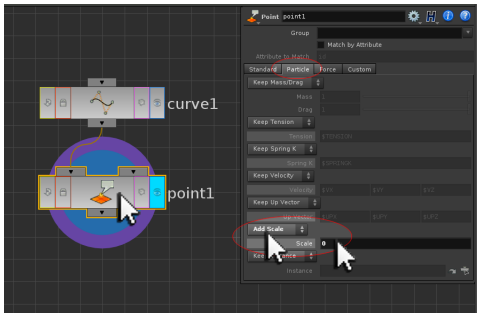
Set the Viewer to a **Front Orthographic View** (**SPACEBAR + 3** with the mouse over the Viewer), and activate **Grid Snapping** from the left hand side stow bar of the Viewer. Carefully draw the profile curve of the plinth, taking care to only draw around its base shape, rather than any extrusion plinth detail.



PREPARING THE CURVE

As with the Sea Horse organic polygon-modelling example, curves in Houdini need to be set not to appear at render time. This will prevent any unwanted modelling artefacts when the model is rendered.

Deactivate the **Background Display Image** and append to the Curve SOP a **Point SOP**. In the **Particle section** of its **parameters**, activate the **Add Scale** option from the Keep Scale drop down menu.

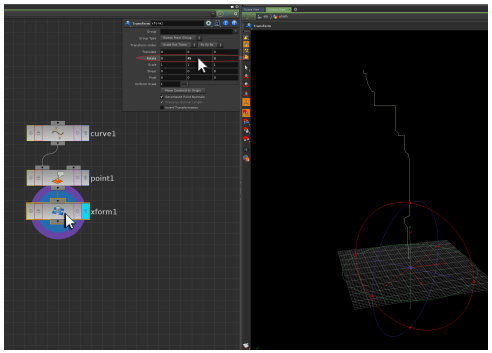


RMB on the **Scale parameter** and **Delete the Channels**. This will automatically set the scale of the curve to 0. A Render Region Preview of the scene will confirm the curve is no longer visible.

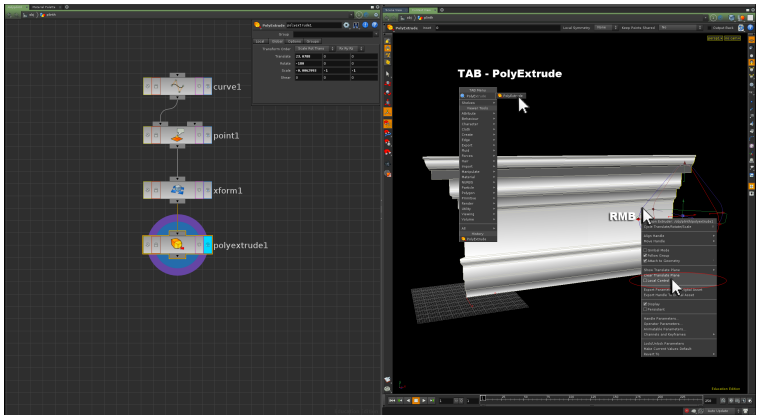
Append to the Point SOP a **Transform SOP**. In the **parameters** specify:

Rotate	**0**	**45**	**0**

This will allow for the curve to be extruded in both the X and Z axis to form a corner.

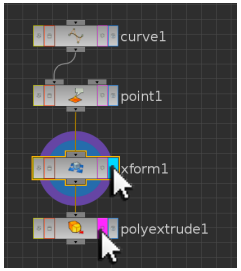


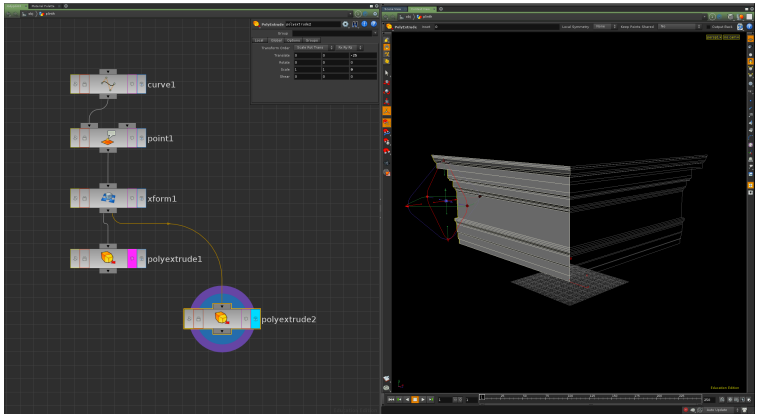
Select the curve in the Viewer, and activate a PolyExtrude Operation. **RMB** on the **PolyExtrude handle** in the **Viewer** and **deactivate Local Control**. Extrude the curve **25** units in the **X Axis**.



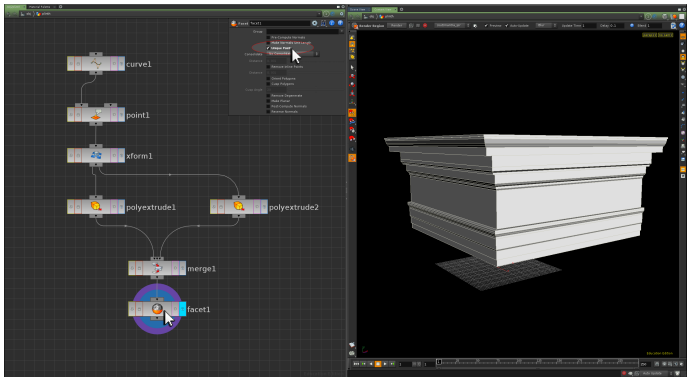
In the **Global** section of the **parameters** for the **PolyExtrude SOP**, reduce the **Scale X parameter** to **0**. This will create a flat end to the extrusion (this can also be done interactively in the Viewer by adjusting the X Axis scale controls of the PolyExtrude handle).

With the PolyExtrude complete, reset the **Display/Render Flag** to the **Transform SOP**, and **Template Display** the **PolyExtrude SOP**.





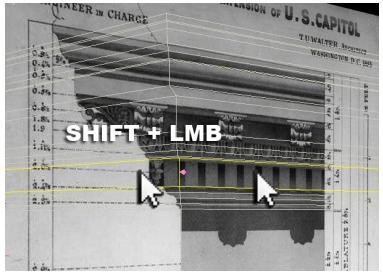
In the **Viewer**, select the curve again and activate a **second PolyExtrude Operation** using the **TAB Menu**. **Before LMB** on the **PolyExtrude tab menu listing**, hold down the **SHIFT** key. This will append the second extrusion as a new network branch. As before **deactivate Local Control**, and extrude the curve **-25** units in the **Z Axis**, reducing the Global **Scale Z parameter** to **0** to create a flat end to the extrusion.



Use a **Merge SOP** to combine **both outputs** of the two **PolyExtrude SOPs**, and append a **Facet SOP** activating the **Unique Points** option in its **parameters**. See file **temple_stage1.hipnc**

ADDING PLINTH DETAIL

In the **Viewer**, **SHIFT Select** the two faces of the plinth which require additional block extrusion detailing.

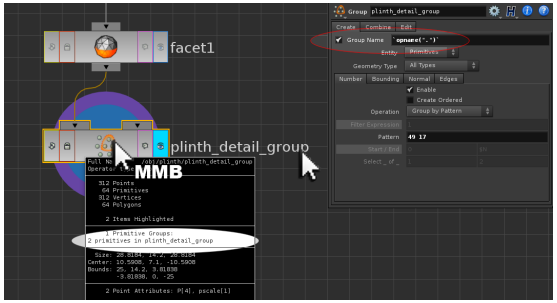


In the **Viewer**, press **TAB** and type **Group Geometry**. This will create a **Group SOP** combining these faces together as a group for easy accessibility.

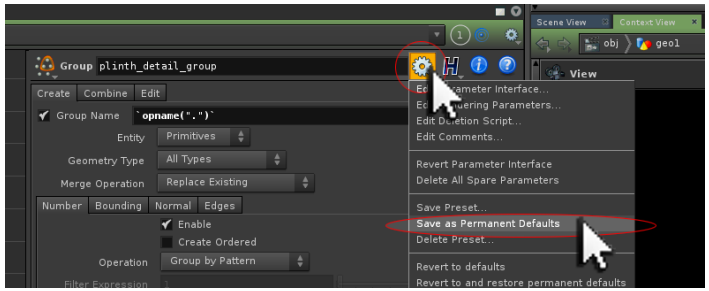
Rename the **Group SOP node** in the **Network Editor** to **plinth_detail_group**. In the **parameters** for this **Group SOP** specify:

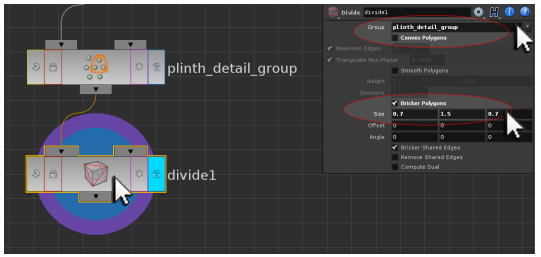
Group Name **`opname(“.”)`**

This is a Houdini native expression that will automatically set the name of the geometry group to the name of the node. This can be verified by **MMB** on the **Group SOP node** where the group name is revealed as part of the operator’s information card.



As this is a useful behaviour for the Group SOP (as it makes seeing group names very easy as they match the node name), this can be set as a **permanent behaviour** for the Group SOP by going to the **parameters Cog menu** and choosing **Save as Permanent Defaults**.

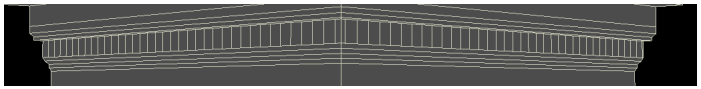




Append a **Divide SOP** to the **plinth_detail_group_Group SOP**. In the **parameters** of the **Divide SOP** specify:

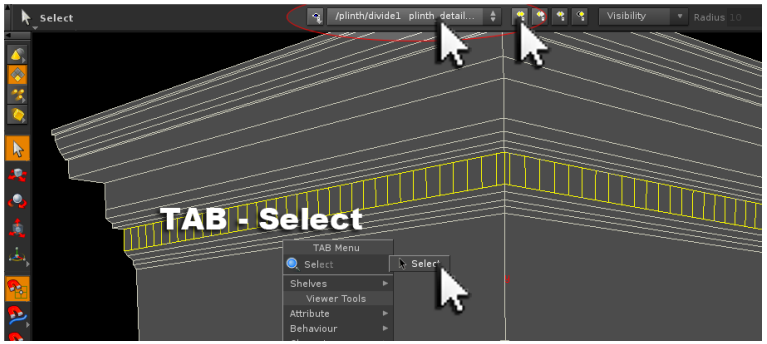
Group	**plinth_detail_group**		
☐	**Convex Polygons**		
☑	**Bricker Polygons**		
Size	**0.7**	**1**	**0.7**

This will internally divide the two faces in the plinth_detail_group at regular intervals without affecting its overall topology.



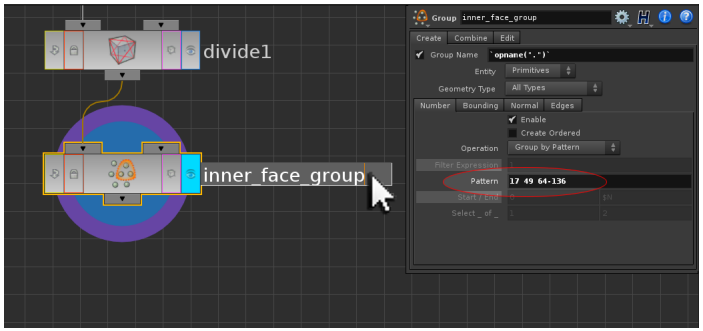
NOTE: These additional faces will automatically be included in the plinth_detail_group. **MMB** on the **Divide SOP** node to activate its information card will confirm this.

In the **Viewer**, press **TAB** and type **select**. Unlike using the Select Arrow button from the left-hand side stow bar of the Viewer; the Tab **Select Tool** also allows for the **reactivation of grouped geometry as the selection**.



From the **Tool Bar** at the top of the **Viewer**, activate the **plinth_detail_group** as the group to select, and press the first yellow button to **add the group to the selection**.

In the viewer, press **TAB** and type **Group Geometry**. This will create a new group that can be further edited to isolate every other primitive face for extrusion.



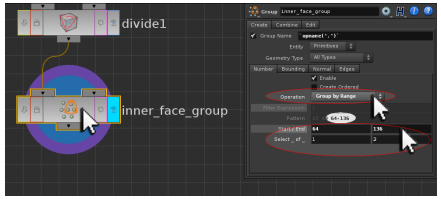
Rename this **Group SOP node** to **inner_face_group**, and note the **Pattern Parameter listing** (in this example, Primitive Face Number **17** & **49** (the original undivided faces) and **64-136** (the faces created by the Divide SOP) are being returned as the new grouped faces).

In the **parameters** for this Group SOP specify:

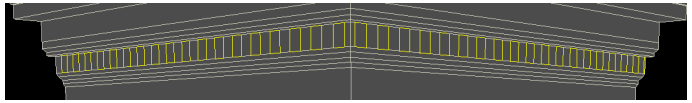
Operation	**Group By Range**	
Start / End	**64**	**136**

NOTE: these numbers are derived from the Group By Pattern parameter now greyed out.

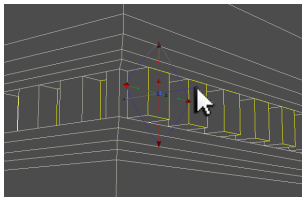
Select _ of _	**1**	**2**



This will select alternate faces from the Divide SOP for extrusion. **NOTE**: **Hidden Line Ghost shading mode** can make this selection easier to see in the Viewer.

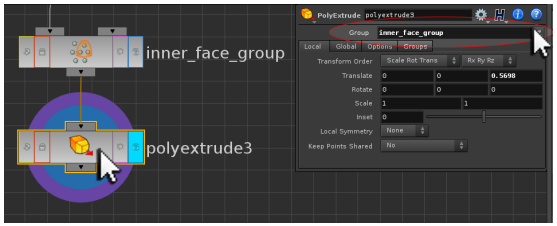


With these detail faces selected, activate a **PolyExtrude SOP** through the **Viewer** and extrude the faces outwards.



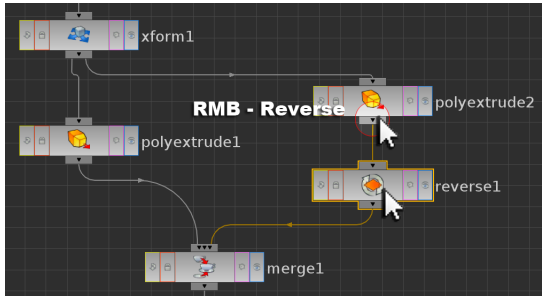
NOTE: the side faces extrude inwards rather than outwards... this can be easily fixed, so do not panic!

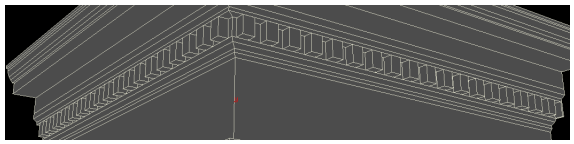
In order to keep the network as procedural as possible, ensure that the **Group Name parameter** of the **PolyExtrude SOP** is reset to **inner_face_group** rather than the Primitive Number listing created by default. This will allow any changes to the inner_face_group to be automatically passed into the PolyExtrude SOP without having to reselect any geometry.



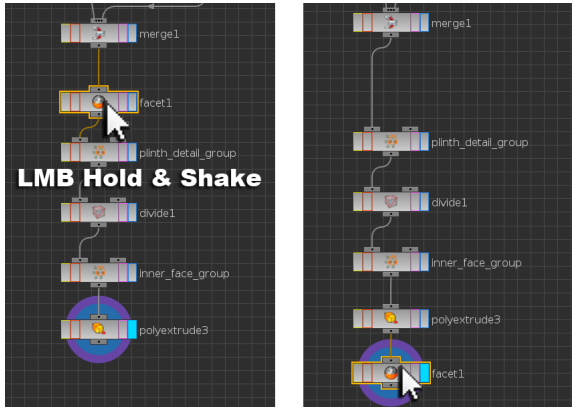
REVERSING FACES

To fix the inward facing extrusion problem return higher in the network to the **PolyExtrude SOP** responsible for **extruding the Z-Axis plinth face**. **RMB Append** to this node a **Reverse SOP**. This will reverse the direction of the vertices (ie which way the operator thinks is out or in).





As a final step **LMB Hold** and **Shake** the **Facet SOP** to lift it out of its current network position. **Move** the **Facet SOP** to the end of the network **activating** its **Display / Render Flag**. This will ensure that the entire plinth renders as a hard surface.



As the Facet SOP works on points rather than primitive faces, it can be moved around without affecting the primitive face modelling work done so far.

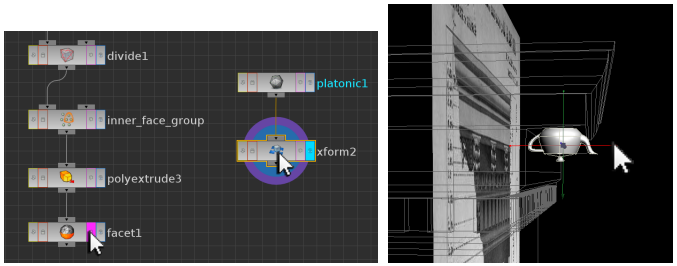
NOTE: The **Offset parameter** of the **Divide SOP** can also be further adjusted to align the extrude faces better with the reference image.

See file **temple_stage2.hipnc**

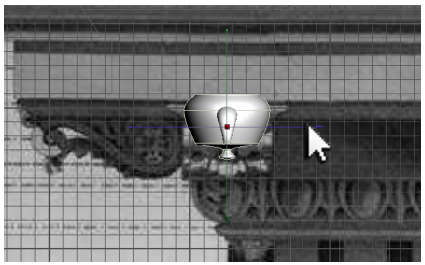
ADDITIONAL PROXY DETAIL

When modelling it is a good idea to utilise proxy geometry for detail that will take time to model. The coving on the plinth is one such area where proxy geometry could be used while the coving geometry is being created. As a **new network** create a **Platonic Solid SOP**. In its **parameters** specify:

Solid Type	**Utah Teapot**



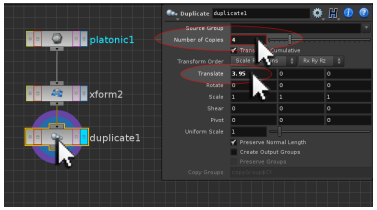
Append a **Transform SOP** to the Platonic Solid SOP, and using the **Reference Display Image as a guide**, **position the teapot** so it sits on the underside of the plinth top area.



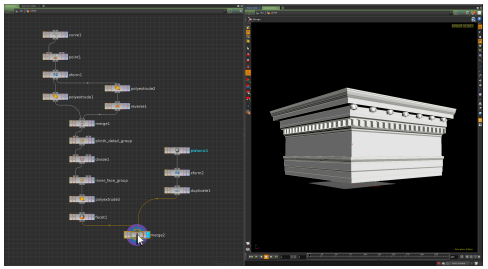
The **Front Orthographic View** (**SPACEBAR + 3** with the mouse over the Viewer) can be used to check the alignment of the teapot with the coving of the reference image.

Append a **Duplicate SOP** to the Transform SOP. In its **parameters** specify:

Number of Copies	**6**
Translate	**3.95**



This will create a line of teapots along the front face of the plinth. **NOTE:** proxy geometry can also be created for the other side of the plinth.

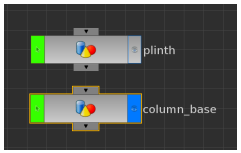


As a final step, create a **Merge SOP** and wire the two networks together. Do not worry about the yellow/black warning stripes that appear on the Merge SOP. **MMB** on the **Merge SOP** reveals this warning is as a result of a mis-match of attributes from the input network chains. This is where the first network chain has an additional geometry attribute (pscale) created by the Point SOP, whereas the second network chain does not.

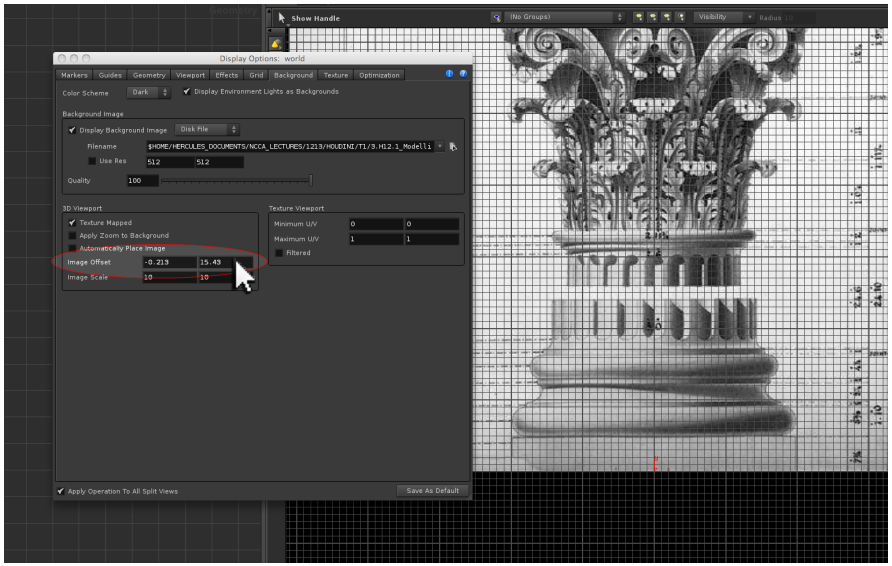
See file **temple_stage3.hipnc**

CREATING THE COLUMN BASE

Return back up to Object Level, turn off the blue Display Flag for the plinth object, and create a new piece of geometry.



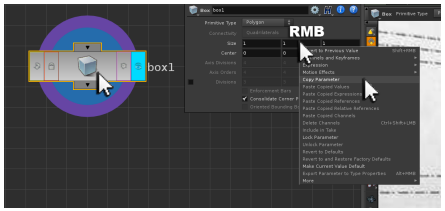
Rename this object to **column_base**. Inside this new object, **delete** the **default File SOP**.



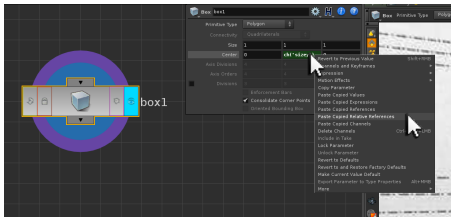
Using the **Viewer Display Options**, modify the **Image Offset** position of the reference image so that the base of the column aligns with the World Origin.

COPYING PARAMETERS & CHANNEL REFERENCING

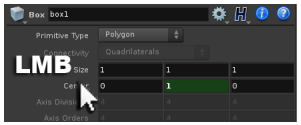
In the Network Editor, create a Box SOP. RMB on its Size Y parameter and choose Copy Parameter from the resulting menu.

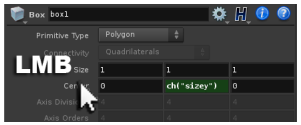


RMB on the **Center Y parameter**, and from the resulting menu choose **Paste Copied Relative References**.



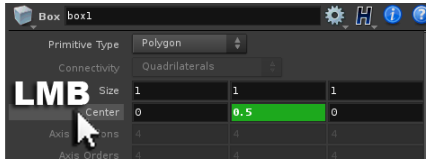
Doing this will automatically set the Center Y position of the box to the same value as the Size Y parameter. This is visually indicated by the **expression ch("sizey")**. This is known as a channel reference. **LMB** on the **Center parameter name** will toggle between the **expression view** of the channel reference, **and its** resulting **numeric value**.



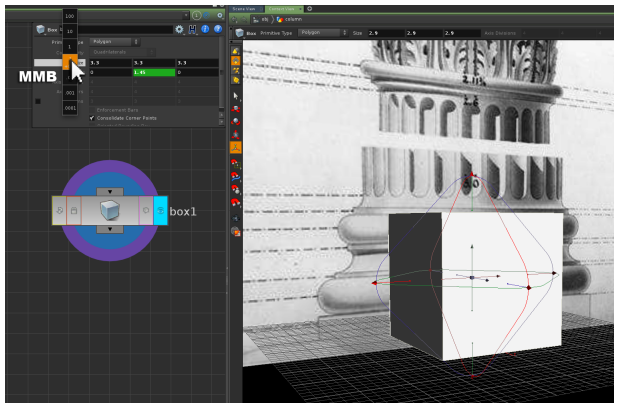


Modify the **channel reference expression** to read:

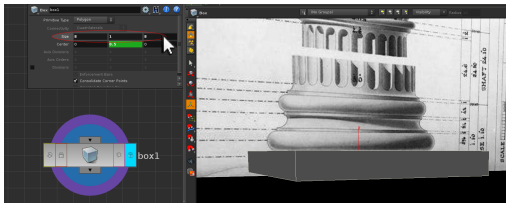
Center	**0**	**ch(“sizey”)/2**	**0**



This will halve the value being returned by the channel reference. Again, **LMB** on the **Center parameter name** will toggle between the channel reference expression, and its resulting value.



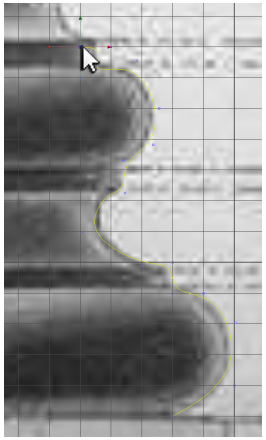
The visual effect of this channel reference is how the box grows in size. Normally if the Size parameter of the Box SOP was adjusted, the box would grow in all directions from the origin point of the scene. The channel reference however now automatically controls the Centre Y position of the box, so that it grows upward from the origin point instead.

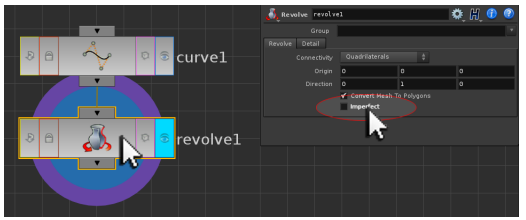


To create a base for the column, modify the **Size parameter** of the **Box SOP** to:

Size	**8**	**1**	**8**

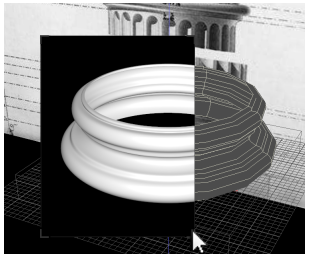
In the **Network Editor**, create a **Curve SOP** as a new network chain. In the **Front Orthographic View** (**SPACEBAR + 3**) draw a **NURBS** profile curve for the cylindrical base above the box. A **Revolve SOP** can then be **appended** to the **NURBS curve**.



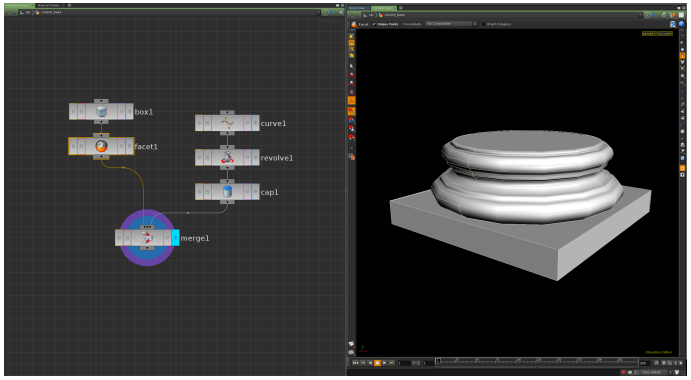


In the **parameters** for the **Revolve SOP**, **deactivate** the **Imperfect** option so that the resulting shape matches exactly the original profile curve.

Do not worry about how the revolved shape looks in the Viewer. A Render Region preview reveals that the shape will render as expected.



A Cap SOP can be appended to the Revolve SOP to close the ends of the revolved shape For accuracy only the top hole needs to be capped, as the lower hole is hidden by the box. This can be achieved by modifying the parameters of the Cap SOP.



The two network chains can then be merged together. The box network chain can also be enhanced by a Facet SOP (set to Unique Points) to ensure a hard surface render. See file **temple_stage4.hipnc**.

CREATING THE COLUMN

The temple column can be created in a number of ways depending upon production context. In its simplest form this can be done using Primitive modelling techniques; but can also be achieved using both NURBS and Polygon Subdivision modelling techniques.

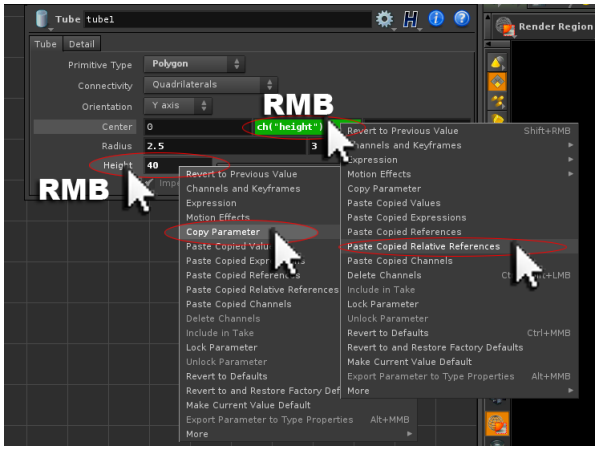
CREATING THE COLUMN USING PRIMITIVE GEOMETRY

Return back to **Object Level** and create a new piece of **Geometry**. **Rename** this new object to **column_primitive**.

Inside this object, **delete** the default **File SOP**, and in its place create a **Tube SOP**. In the **parameters** of the Tube SOP specify:

Primitive Type	**Polygon**	
Radius	**2.5**	**3**
Height	**40**	

RMB on the **Height parameter** and from the resulting menu choose **Copy Parameter**.



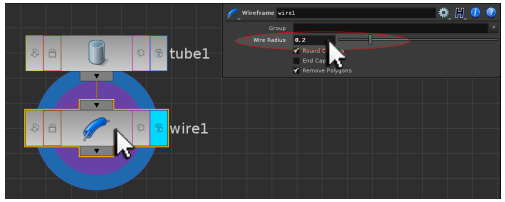
RMB on the **Center Y parameter** and from the resulting menu choose **Paste Copied Relative References**. **Modify** the **channel reference expression** from:

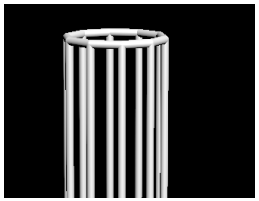
ch(“height”) to **ch(“height”)/2**

Now when the height of the tube is adjusted, it grows from the Origin Point upwards.

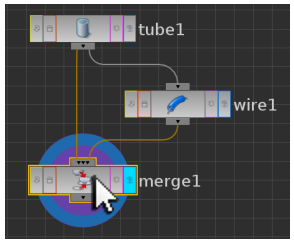
Append to the Tube SOP a **Wireframe SOP** (**NOTE**: A **PolyWire SOP** can also be used for this step). In the **parameters** for the **Wireframe SOP** specify:

Wire Radius **0.2**



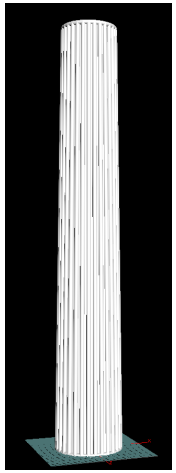


This will convert the tube into a wireframe version of itself based upon the polygon faces of the input geometry.



Wire the outputs of both the **Tube SOP** and the **Wireframe SOP** into a **Merge SOP** (**NOTE: MMB** on the **output** of a **node** will activate a new operator as a **new network branch**).

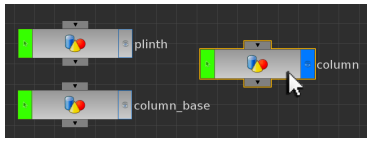
The **Detail > Columns parameter** of the **Tube SOP** can be further adjusted to create a primitive column effect.



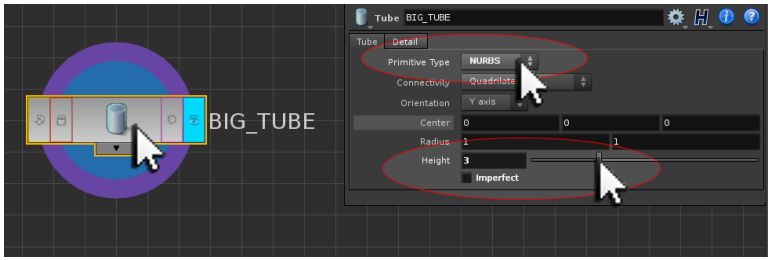
See file **temple_stage4_a.hipnc**

CREATING THE COLUMN USING NURBS MODELLING

Return back to Object Level and create a new piece of Geometry. Rename this new object to column.



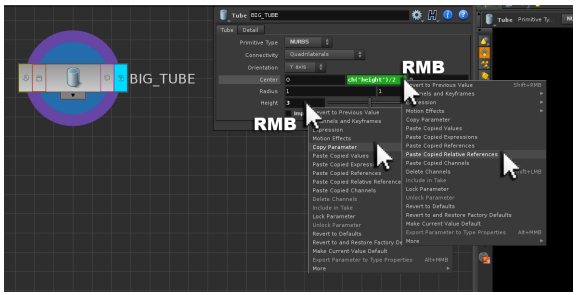
Go inside this object and **delete** the **default File SOP**. In its place create a **Tube SOP** and rename it to **BIG_TUBE**.



In the **parameters** for the **Tube SOP** specify:

Primitive Type	**NURBS**
Height	**3**
☐	**Imperfect**

As with the creation of the column base, channel referencing can be used to control the height of the tube relative to the Origin Point of the object. **RMB** on the **Height parameter** and from the resulting menu choose **Copy Parameter**.



RMB on the **Center Y parameter** and from the resulting menu choose **Paste Copied Relative References**.

Modify the **channel reference expression** from:

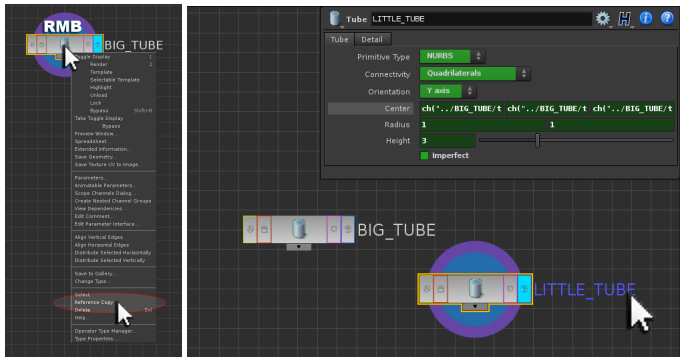
ch(“height”) to **ch(“height”)/2**

Now when the height of the tube is adjusted, it grows from the Origin Point upwards.

REFERENCE COPIES OF NODES

Sometimes it is useful to create **reference copies** of certain nodes. This is where channel referencing for all parameters are automatically set in the copied node; meaning that any adjustments to the original node also affect the reference copy. In the example of the column, a reference copy of the BIG_TUBE can be created to cut away the indentations detail of the column shape. By creating a **reference copy** of the BIG_TUBE, any modification of its parameters will automatically be passed onto its reference copy.

RMB on the **BIG_TUBE** node and from the **resulting menu** choose **Reference Copy**.

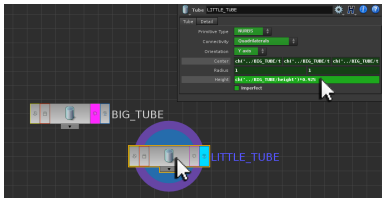


Rename this reference copy node to **LITTLE_TUBE**. **NOTE**: The name of a reference copy node is displayed as purple to indicate that it is a reference copy. The parameters of a reference copy node will be green indicating their channel references to the original node.

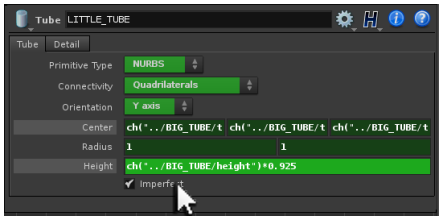
MODIFYING REFERENCE COPIES

While having a channel referenced copy of a node can prove useful, the channel references themselves can be further modified to create a more dynamic relationship between the two nodes. In the **parameters** for the **LITTLE_TUBE**, modify the **Height** parameter channel reference from:

ch("../BIG_TUBE/height") to **ch("../BIG_TUBE/height")*0.925**



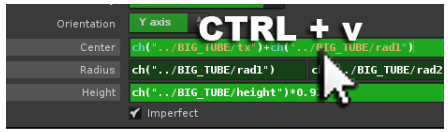
Template the BIG_TUBE node, and **RMB** on the **Imperfect parameter** of the **LITTLE_TUBE**, and from the resulting menu choose Delete Channels. This will remove the Channel Reference from this parameter allowing it to be modified independently from the BIG_TUBE node.



With the Channel Reference deleted, this parameter can now be **tick-box activated**. Doing this will help improve the accuracy of how the LITTLE_TUBE can cut a shape out of the BIG_TUBE.

AUTOMATIC POSITIONING AND SCALING THE LITTLE_TUBE

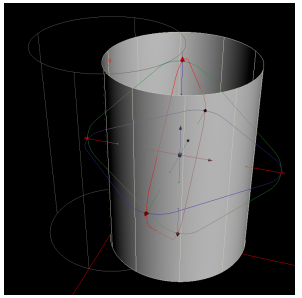
LMB on the **Radius parameter name** of the LITTLE TUBE node to reveal its channel reference. **Select** the first radius **channel reference expression** as text, and use **CTRL + c** to copy it into memory.



Activate the **Center X** parameter as text, and add a **+** to the end of the expression. Use **CTRL + v** to paste append the copied channel reference expression. The final expression for the Center X parameter should now read:

Center X **ch("../BIG_TUBE/tx") + ch("../BIG_TUBE/rad1")**

Adding these channel references together visually results in the LITTLE_TUBE automatically being positioned on the edge of the BIG_TUBE.



Further modify both of the LITTLE_TUBE **Radius** parameter **Channel References** dividing the expressions by 10:

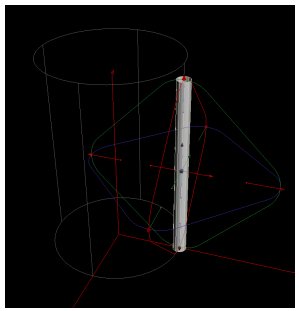
Radius ch("../BIG_TUBE/rad1")/10 ch("../BIG_TUBE/rad2")/10

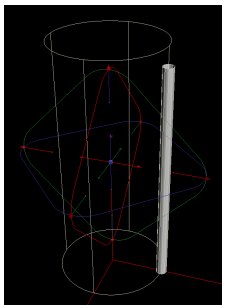
This will automatically resize the radius of the LITTLE_TUBE to 1/10th of the size of the BIG_TUBE radius, no matter what size the BIG_TUBE's radius is set to.

As a final step, **RMB** on the **Height Parameter** of the **LITTLE_TUBE** and from the **resulting menu** choose **Copy Parameter**. **RMB** on the **Center Y** parameter and from the **resulting menu** choose **Paste Copied Relative References**. **Modify** the resulting **channel reference expression** to:

Center Y ch("height")/2

This will cause the LITTLE_TUBE to grow from the origin point upwards. Now whenever the BIG_TUBE's parameters are adjusted, the LITTLE_TUBE will automatically reposition and scale itself correctly relative to the BIG_TUBE.



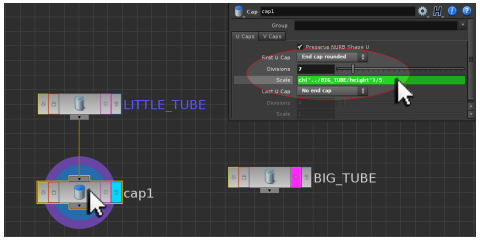


See file **temple_stage5.hipnc**

CREATING THE CUT OUT

Append a **Cap SOP** to the output of the **LITTLE_TUBE**, and in its **parameters** specify:

First U Cap	**End cap rounded**
Divisions	**7**
Last U Cap	**No end cap**



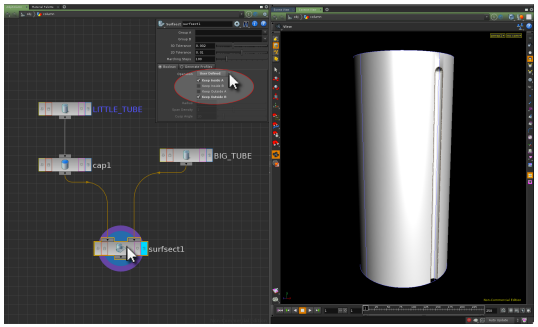
Channel Copy the **Height parameter** from the **BIG_TUBE**, and paste its **relative references** into the **Scale parameter** of the **LITTLE_TUBE**. Modify this channel reference expression to:

Scale ch("../BIG_TUBE/height")/5

This will automatically rescale the roundness of the First U Cap relative to the height of the BIG_TUBE.

Append a **Surfsect SOP** to the output of the **Cap SOP**, wiring the output of the **BIG_TUBE** as its **second input**. In the **parameters** of the **Surfsect SOP** specify:

Operation	**User Defined:**
☑	**Keep Inside A**
☑	**Keep Outside B**

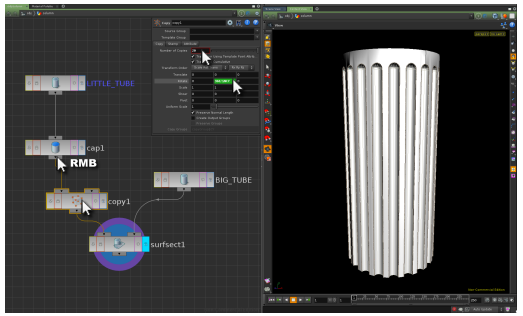


This will cut out the LITTLE_TUBE from the BIG_TUBE. **NOTE**: The Surfsect SOP allows for cut away operations between NURBS Surfaces. For cutting polygon shapes from each other, use a **Cookie SOP** instead (which works especially well when both inputs are **Subdivided Geometry** shapes).

RMB on the output of the **CAP SOP** and insert a **Copy SOP** into the network chain. This can be used to repeat the cut out shapes around the BIG_TUBE. In the **parameters** for the **Copy SOP** specify:

Number of Copies	**20**		
Rotate	**0**	**360 /$NCY**	**0**

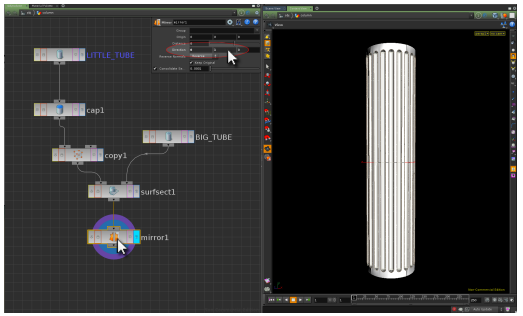
$NCY is a **local variable** of the **Copy SOP** that automatically returns the **Number of Copies** parameter value. A full list of Copy SOP local variables can be found in the ? Help Card for the Copy SOP.



NOTE: a channel reference based expression would also work for this step; for example **360/ch("ncy")**. **NOTE**: for speed, the **Number of Copies can be reduced until Render time**.

Append a **Mirror SOP** to the **output** of the **Surfsect SOP**. In its **parameters** specify:

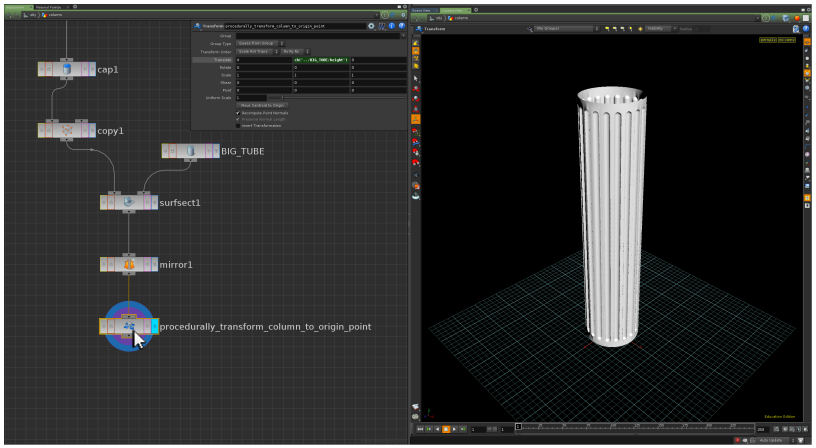
Direction	**0**	**1**	**0**



This will mirror the geometry in the Y Axis, completing the column. See file **temple_stage6.hipnc**

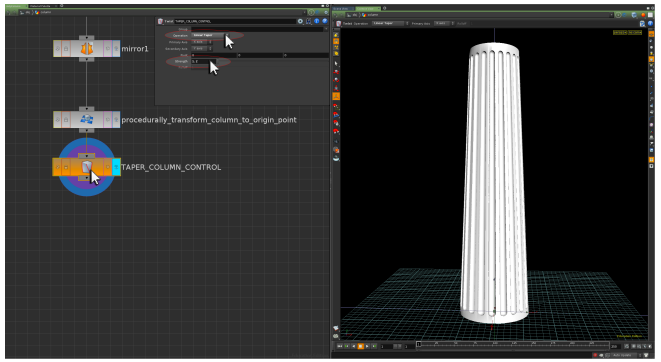
OPTIONAL COLUMN FUNCTIONALITY

Additional nodes can be added to the column network to improve its overall functionality. A simple first step is to append a Transform SOP to the network that will automatically position the column so that is grows from the origin point of the scene upwards.

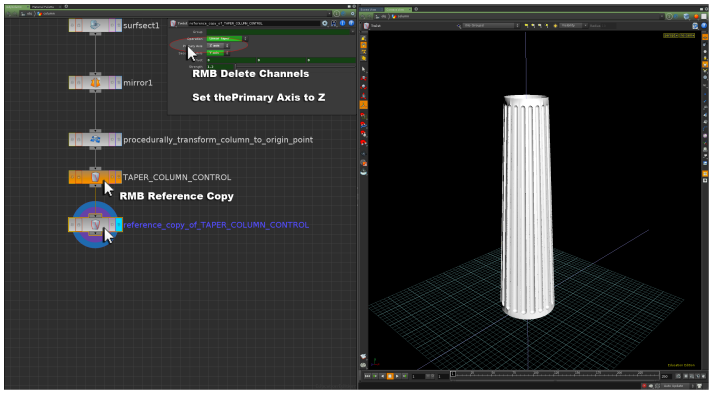


In the **parameters** for the **BIG_TUBE** Tube SOP, **RMB** on the **Height** parameter and choose **Copy Parameter** from the resulting menu. This can be set as a **channel reference** in the appended **Transform SOP** by **RMB** on the Transform SOP's **Translate Y** parameter and choosing **Paste Copied Relative References**.

The ability to **taper the column geometry** is another useful piece of functionality for this network. This can be achieved by appending a **Twist SOP** to the end of the network. In the **parameters** of the **Twist SOP**, specify an **Operation** of **Linear Taper**, with a **Strength** value of **1.2**. This will cause a tapering of the column in the default axis of X and Y only.



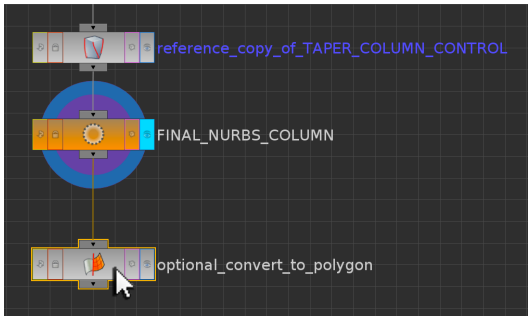
In order to make the column also taper in the Z axis as well, a **Reference Copy** of the Twist SOP can be created to achieve this.



RMB on the **Primary Axis parameter** of the **Twist SOP Reference Copy**, and choose **Delete Channels** from the resulting menu. This will allow for the **Primary Axis** of the Twist SOP Reference Copy node to be set to the **Z Axis** independently of the original Twist SOP.

IMPORTANT NOTE: When using Reference Copies of nodes, adjusting the parameters of either the original node or its reference copy will result in both nodes having that specified value (IE parameter value adjustments between a node and its Reference Copy flows both ways).

As a final step, a Null SOP can be appended to the network to denote the completion of the NURBS column.



An **optional Convert SOP** can also be added, to allow an end user to convert the NURBS column geometry into Polygons if necessary. This node can simply be added to the end of the network, and if a Polygonal output is required, its **Display** / **Render Flag** can be activated accordingly.

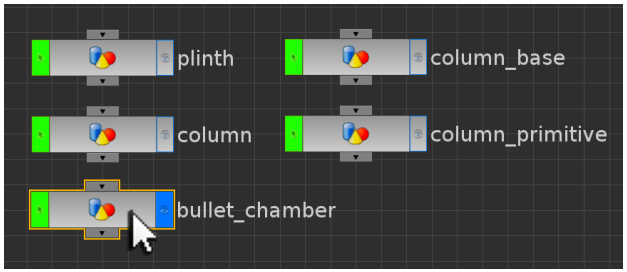
SEE FILE temple_stage7.hipnc

INORGANIC POLYGON SUBDIVISION COOKIE MODELLING

A variation on the NURBS column network is to recreate the same effect using Polygons instead of NURBS. The same methodology of cutting one piece of geometry away from the other can be achieved by using a **Cookie SOP** for polygons (as opposed to a Surfsect SOP for NURBS based geometry). The **Cookie SOP** works especially well on **subdivided polygonal geometry** rather than default polygons, as subdivided polygonal geometry can help prevent Cookie SOP errors as well as allowing for multiple cookies to take place.

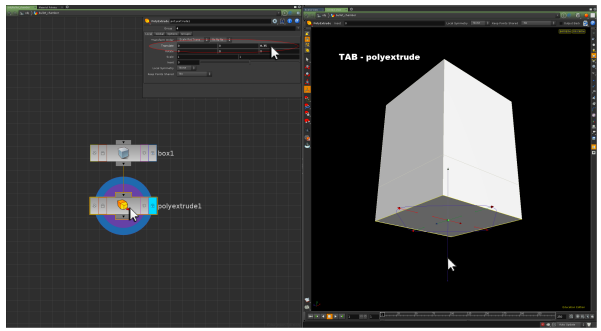
Both the NURBS Surfsect SOP and the Polygon Cookie SOP work on the artistic principle of **positive** and **negative space**. In both cases geometry can be created to cut away its shape from another geometry source.

Rather than recreating the column as polygons, a bullet chamber shape can be created instead. The principles outlined in this exercise can then be applied to the creation of a column if required.



Return to **Object Level** and create a new **Geometry Object**. Rename it to **bullet_chamber** and enter into its **SOP Level**. Delete the default **File SOP** and with the mouse over the Network Editor, press **TAB** and type **box** to create a **Box SOP**. Leave the Parameters for the **Box SOP** at their default settings.

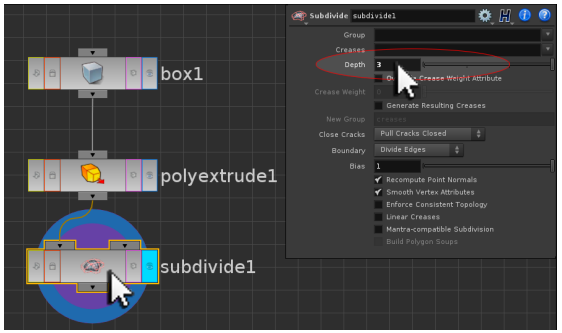
Interactively select the bottom face of the box and with the mouse over the **Viewer**, press **TAB** and type **polyextrude**. This will activate a **PolyExtrude SOP**.

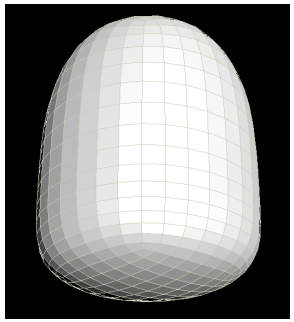


PolyExtrude the bottom face of the **Box SOP** down. In the **Parameters Pane** for the **PolyExtrude SOP**, specify the **Translate** values of:

Translate	**0**	**0**	**0.35**

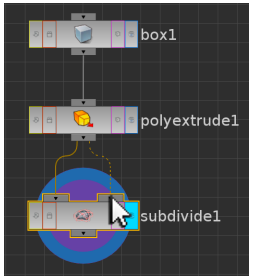
To the **PolyExtrude SOP** append a **Subdivide SOP**. In the **Parameters** for the **Subdivide SOP** increase the **Depth Parameter** to **3**. This will form the basis of the bullet chamber.

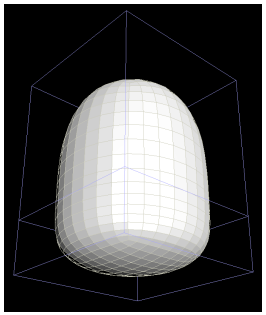




SUBDIVISION CREASES

The **second input** of the **Subdivision SOP** can be used to initiate creases. Subdivision creases can be used to create sharp edges on a subdivided model. In the **Network Editor**, additionally wire in the output of the **PolyExtrude SOP** to the second input of the **Subdivide SOP**. A blue wire frame version of the un-subdivided geometry will appear. These are the currently inactive crease faces.

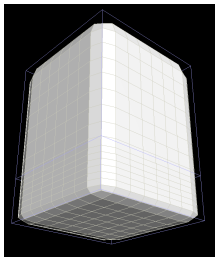




In the **Parameters** for the **Subdivide SOP**, activate the **Override Crease Weight Attribute** and increase the **Crease Weight** Parameter to:

Crease Weight	**2**

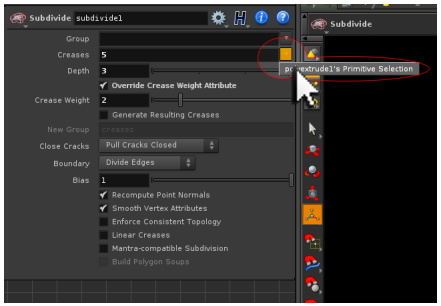
The subdivided mesh will inflate to the boundaries of the blue wire frame.

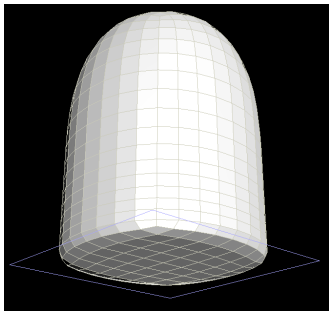


SPECIFYING A SPECIFIC CREASE FACE

At present all of the un-subdivided faces are being used to create creases; however individual or groups of faces can be used instead. This can be done using the **Input Button** associated with the **Creases Parameter**. **LMB** on this Input Button will list any specific data that is being fed into the second input of the Subdivide SOP. In this example, the list will read:

polyextrude1's Primitive Selection





Selecting this will add the number **5** to the **Creases Parameter**. This is the Primitive Number of polyextruded face at the base of the un-subdivided mesh. The result of this action is that only the base face is used to inflate the subdivide mesh onto. This visually results in the body of the bullet chamber being formed.

NOTE: Primitive Numbers of geometry can be displayed in the Viewer activating the Display Primitive Numbers button () found on the right hand side Viewer stow bar.

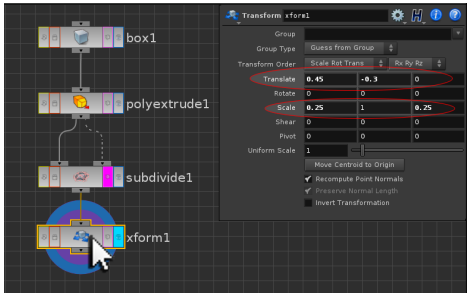
CREATING THE BULLET CHAMBER DETAIL

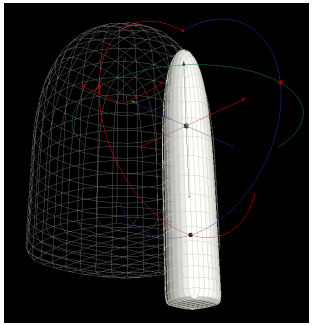
In the **Network Editor**, activate the **Template Flag** of the **Subdivide SOP** and append to it a **Transform SOP**.

Set the **Transform SOP Parameters** to:

Translate	**0.45**	**-0.3**	**0**
Scale	**0.25**	**1**	**0.25**

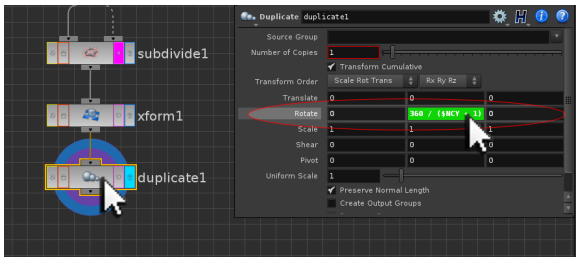
This will create a geometry shape that can be used to cut into the original shape.

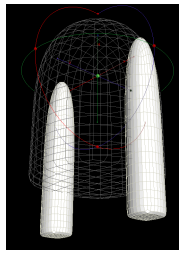




Append to the **Transform SOP** a **Duplicate SOP**. Specify the **Number of Copies** to be **5** and enter into the **Rotate Y Parameter** the following expression:

Rotate	**0**	**360 / ($NCY + 1)**	**0**





With a Number of Copies parameter value of 1, two sets of geometry are produced (the original and its duplicate). The default behaviour of the Duplicate is therefore to automatically create a duplicate of its input.

In Houdini most operators come with a set of Local Variables (similar to $HOME or $PATH) that can be entered into its numeric parameters. In the case of the **Duplicate SOP** the value of the **Number of Copies** Parameter can be automatically returned by the Local Variable **$NCY** (which is also the parameter's internal name).

The expression **360 / ($NCY + 1)** therefore equates to 360 degrees divided by the total number of copies + 1.

NOTE: A full list of Local Variables for an operator can be seen by activating the ? Help Card of the Parameters dialog.

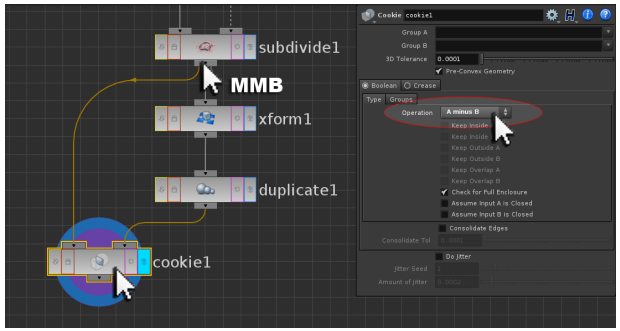
See file temple_stage7_a.hipnc

THE COOKIE SOP

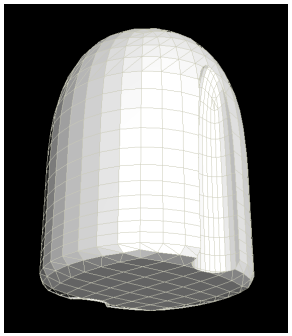
In order to use the duplicates to cut into the original shape, the **Cookie SOP** can be used. The **Cookie SOP** is named after a baking technique of cutting rolled out cookie dough with a metal shape in order to create biscuits (cookies). In terms of Houdini, a **Cookie SOP** operation requires two inputs (A + B). Either one of them can be chosen as the cutter.

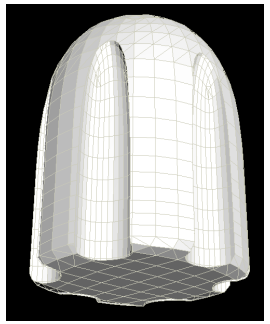
In the **Network Editor**, **MMB** on the output of the **Subdivide SOP** to create a **Cookie SOP** as a new network branch. Wire the output of the **Duplicate SOP** into the **second** input. In the **Parameters** of the **Cookie SOP**, specify the **Operation** parameter as:

Operation **A minus B**



The result of the Cookie SOP can be seen in the Viewer (turn off the Template Flag of the Subdivide SOP).



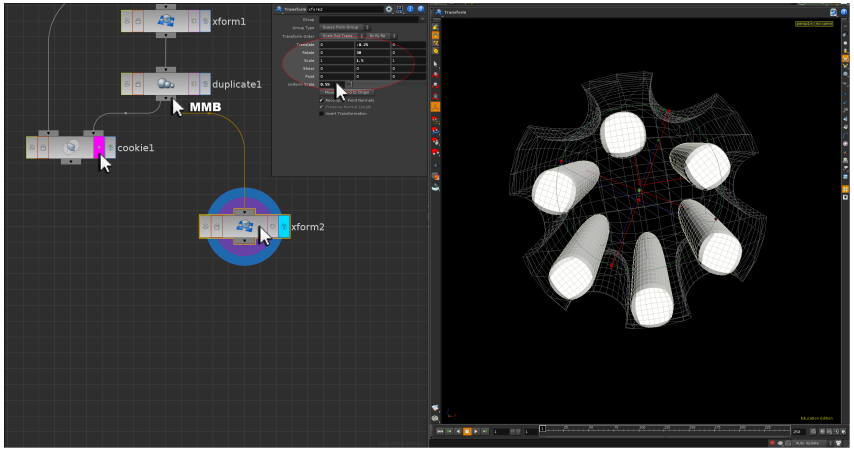


Increasing the **Number of Copies** parameter of the **Duplicate SOP** to a value of **5** will complete the bullet chamber exterior.

ADDING MORE DETAIL

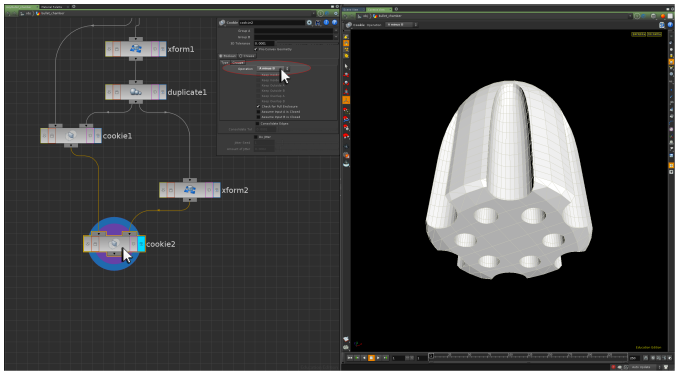
The output of the **Duplicate SOP** can be used to create the holes for the bullet chamber. To do this, **MMB** append as a new network branch a **Transform SOP** to the Duplicate SOP. In the parameters for this second **Transform SOP** enter the following values:

Translate	**0**	**-0.25**	**0**
Rotate	**0**	**30**	**0**
Scale	**1**	**1.5**	**1**
Uniform Scale	**0.55**		



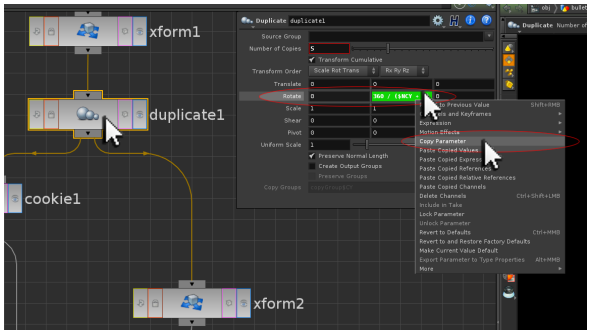
This will create another piece of geometry on which another cookie can be performed using the output of the first Cookie SOP and the second transform SOP.

To the output of the Cookie SOP append **a second Cookie SOP**, wiring the output of the second Transform SOP as its second input. As before set the **Cookie SOP Operation** to **A minus B**.

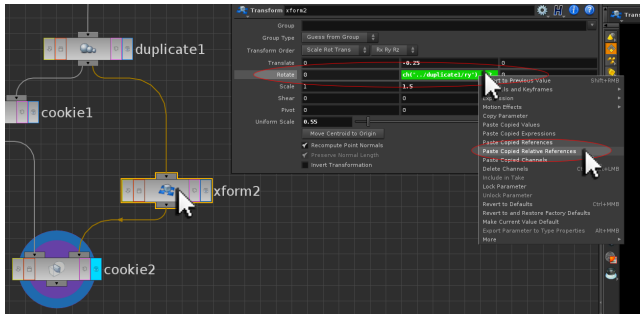


COPYING PARAMETERS

In Houdini sometimes it is useful to reference a value from one operator into another. This is done by **copying parameters**. For the example of the bullet chamber, this would allow the Rotation Y parameter of the second transform to be procedurally controlled. **RMB** on the **Rotate Y Parameter** of the **Duplicate SOP**, and from the resulting contextual menu choose **Copy Parameter**.



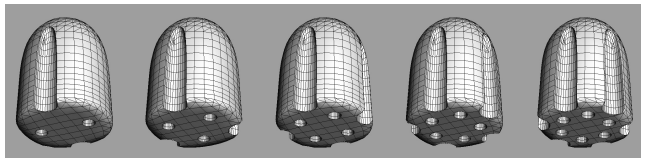
Locate the **Rotate Y Parameter** of the second **Transform SOP**, and again **RMB** on it to bring up the contextual menu. This time select **Paste Copied Relative References**.



The value contained in the Rotation Y parameter of the Duplicate SOP is now being passed into the Rotation Y parameter of the second Transform SOP. The **ch()** function returns the value of the path specified. In this case the path is looking in the parent directory (**../**) for a node called **duplicate1** and a parameter inside it called **ry**. Amend this channel reference from:

ch("../duplicate1/ry") to **ch("../duplicate1/ry") / 2**

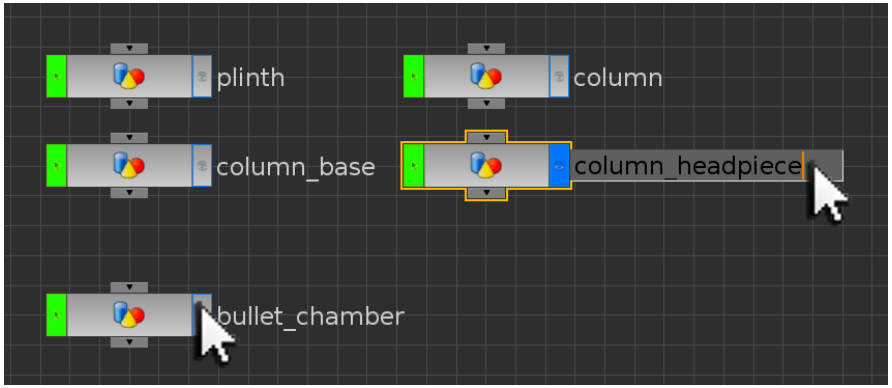
The result of this modification is that this Rotation Y parameter will always be half of the original rotate value. In visual terms, the **Number of Copies** parameter located in the **Duplicate SOP** now automatically controls the rotational placement of both the cookie operations. Increasing or decreasing the Number of Copies parameter will affect the model accordingly.



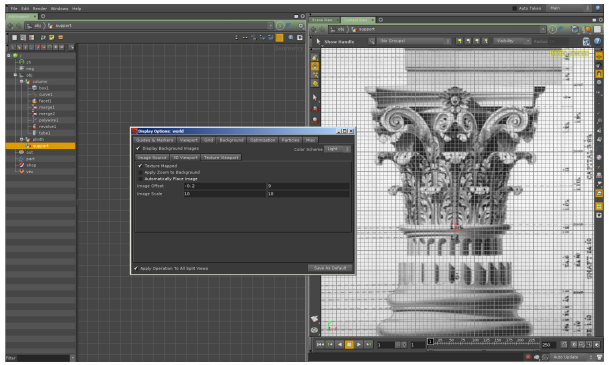
See file temple_stage7_b.hipnc

CREATING THE COLUMN HEADPIECE

Return back up to **Object Level** and **turn off** the **Blue Display flag** for the bullet_chamber object. Create a **new piece of geometry**, and **rename** it to **column_headpiece**.

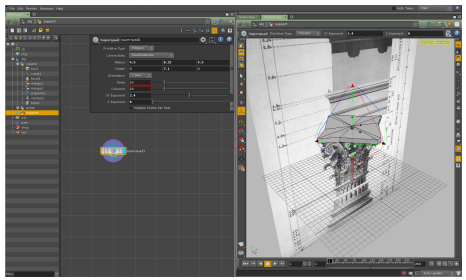


Go inside this object and **delete the default File SOP**. Using the **Viewer Display Options**, move the reference image so that the base of the headpiece is at the World Origin (**Image Offset 0 9; Image Scale 10 10**).



In the **Network Editor** create a **Super Quad SOP**. In its **parameters** specify:

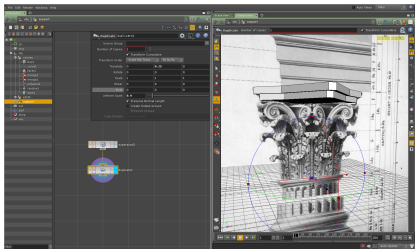
Radius	**4.5**	**0.25**	**4.5**
Center	**0**	**7.1**	**0**
XY Component	**2.4**		
Z Component	**0**		



This will create a suitable shape for the top of the headpiece.

Append a **Duplicate SOP** to the **Super Quad SOP**. In its **parameters** specify:

Translate	**0**	**0.25**	**0**
Uniform Scale	**0.9**		

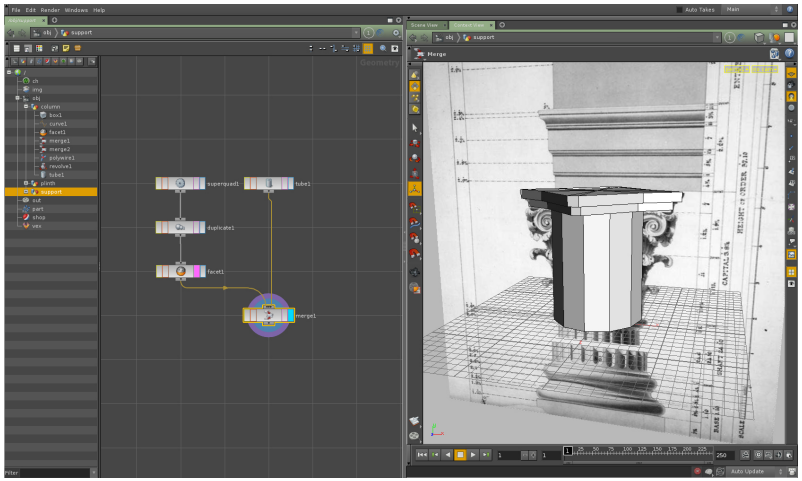


This will complete the top part of the plinth. Append a **Facet SOP** (with **Unique Points** activated in its parameters) to the **Duplicate SOP** to ensure the geometry renders as flat surfaces.

As a **new network chain** create a **Tube SOP**. In its **parameters** specify:

Primitive Type	**Polygon**		
Center	**0**	**3.35**	**0**
Radius	**2.5**	**2.5**	
Height	**6.1**		
Details >			
Columns	**12**		

This Tube SOP can then be **merged** with the **duplicated Super Quad geometry**.



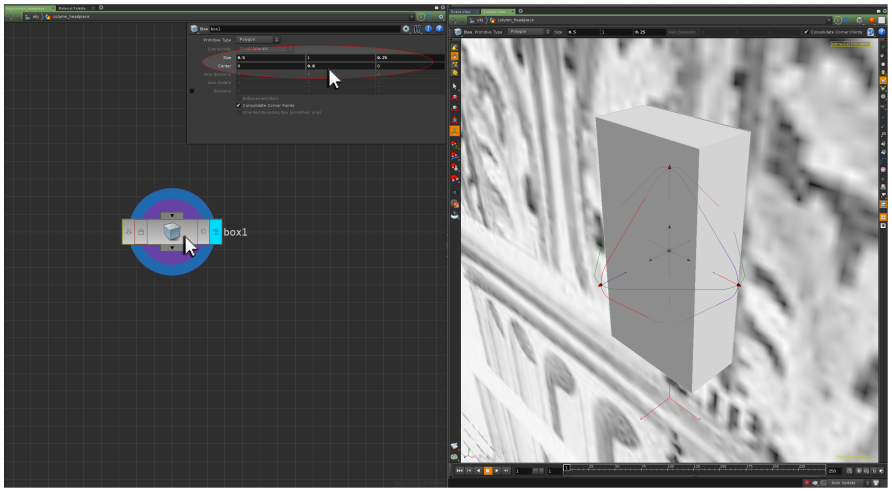
See file **temple_stage8.hipnc**

CREATING HEADPIECE DETAIL – SUBDIVISION BOX MODELLING

Still inside the column_headpiece object, place a **Box SOP** in a **new part** of the **Network Editor**. In its **parameters** specify:

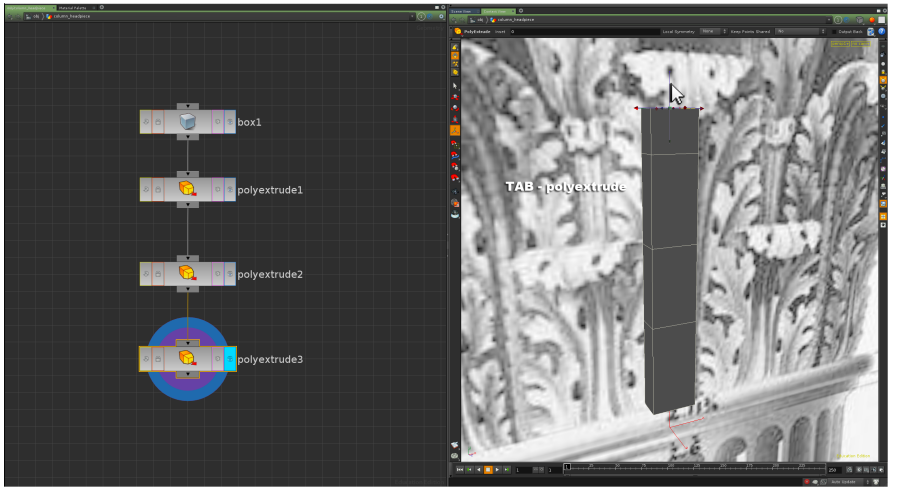
Primitive Type	**Polygon**		
Size	**0.5**	**1**	**0.25**
Center	**0**	**0.8**	**0**

This will create a small flat box from which a column filigree can be generated.



In the **Viewer**, **select** the **top face** of the **Box SOP**, and press **TAB** to activate a **PolyExtrude SOP**. Using the reference image as a guide, **translate up** the **top face** to created a second set of box faces on top of the first. The height of this first extrusion should be roughly the same as the height of the original box.

Press **q** with the **mouse over the Viewer** to **repeat this step twice more**, creating a stack of box faces.

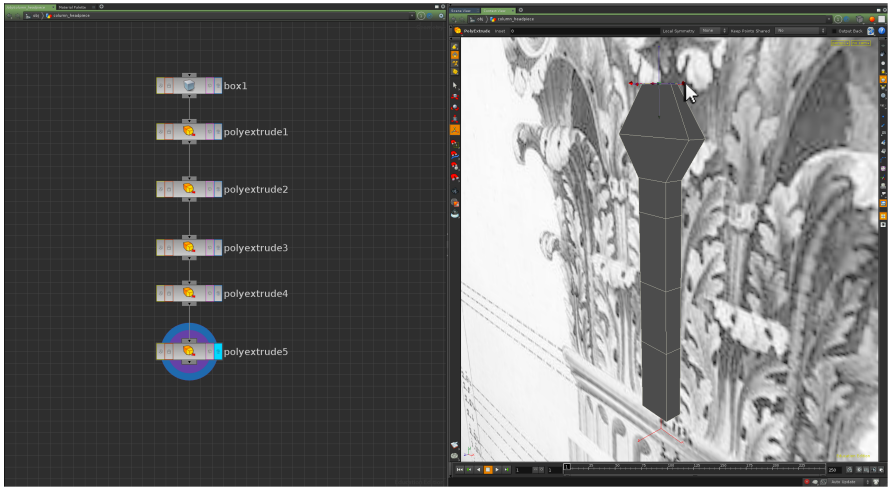


Set the height of the third PolyExtrude operation to roughly 1/3 the height of the original box.

NOTE: Setting the **Viewer Display** to **Hidden Line Ghost** can make this process easier to see against the reference image.

It does not matter how rough these polyextrusions are, as the nodes generating them can be further adjusted after the basic filigree geometry has been mapped out.

Press **q** with the mouse over the **Viewer** to **repeat this step twice more**, this time creating a slightly tapered paddle shape at the top of the stack.

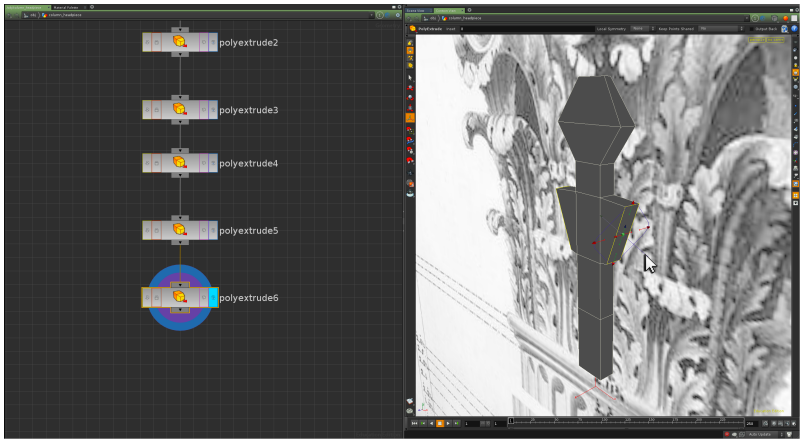


See file temple_stage8_a.hipnc

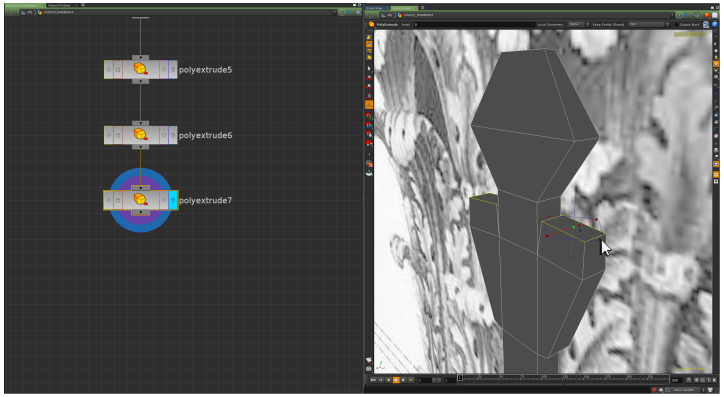
Creating the Filigree Arms

In the **Viewer**, select the **two side faces** of the **third box stack**, and again perform a **PolyExtrude** operation.

Translate out these polyextruded faces to create a pair of arms for the filigree. Pressing **r** with the **mouse over the Viewer** will activated a rotate handle that can make these arm extrusions slightly angled. Pressing **y** with the **mouse over the Viewer** will **restore** the original **polyextrude transform handle**.



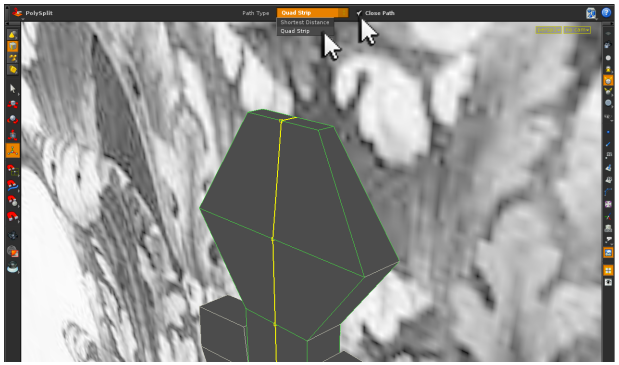
In the Viewer, select the two top faces of these arm extrusions, and create a new PolyExtrude operator, this time translating these faces upwards.



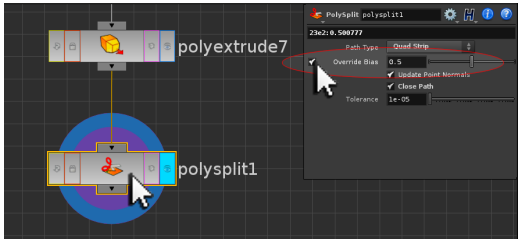
These extruded faces can also be scaled in slightly to create the filigree arms.

POLYSPLIT QUAD STRIPPING

With the **mouse over the Viewer**, **activate a PolySplit** operation. **Select** the **middle** of a **centre top edge** of the shape, and press **Enter** to confirm the operation. **NOTE:** By default, a single selection polysplit will simply create a single point at the centre of the top edge.



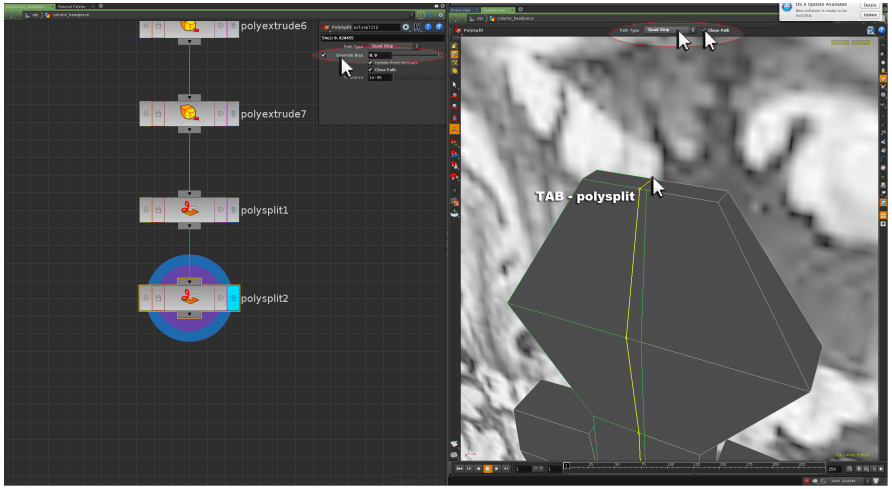
Using the **Viewer based parameters** of this PolySplit operation, set the **Path Type** to **Quad Strip**, and **tick** the **Close Path** option. This will **automatically create a ring of splits** around the vertical centre of the shape.



The **Override Bias** parameter of the **PolySplit SOP** can also be activated to ensure that this ring of splits is exactly in the middle of the shape.

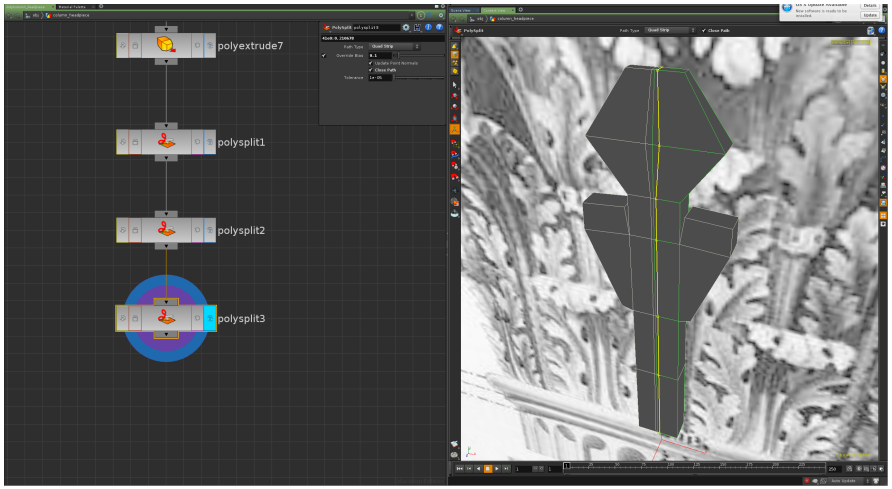
Repeat this quad strip process, this time creating a new quad strip on the left top edge of this centre line.

As before, activate the **Override Bias** parameter of this **second Polysplit SOP**, this time setting a value of **0.9**. This will create a new vertical ring of splits close to the centre quad strip.



Repeat this process, this time creating a **Quad Strip** to the **right** of the **centre line**. In the **parameters** for this **third PolySplit** set the **Override Bias** value to **0.1**. This will create an equivalent ring of splits to those on the left side of the centre ring.

NOTE: The Override Bias parameter is based upon the edge length where the Quad Strip is occurring. This edge length is considered as a value between 0 and 1, with 0 at one end of the edge, 0 at the other end of the edge, and 0.5 in the centre of the edge.



See file temple_stage8_b.hipnc

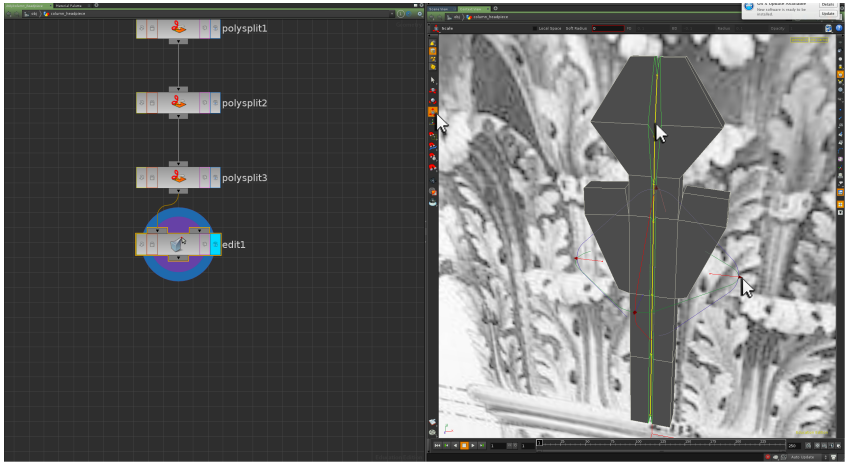
SELECTING EDGE LOOPS

In the **Viewer**, activate the **Select arrow**, and **select one** of the **edges** of the **centre ring**. **Pressing f** with the **mouse over the viewer** will step forward the selection by one edge, relative to the small pink arrow at the end of the edge selection. Pressing SHIFT + r will reverse the direction of the next edge selection (IE put the small pink triangle at the other end of the edge). Pressing **l (L)** with the mouse over the Viewer will automatically select the entire ring of edges.

THE VIEWER TRANSFORM BUTTONS

On the left hand side of the Viewer are a number of Transform buttons (Translate, Rotate, Scale). These buttons have different functions depending upon the level of Houdini. At Object Level, these Transform Buttons can be used to perform Object Level transformations on scene objects (for example move an object around). When working inside an Object (at its Geometry Level), these Transform Buttons will automatically insert an Edit SOP into the active network chain. This behaviour difference can take a while to get used to when you are new to Houdini; and in the context of the Geometry Level, using these Transform SOPs is exactly the same as pressing TAB and typing Edit.

Use the **Select Arrow** to select the **central ring of edges**, and then **press** the **Scale Transform Button** located on the **left hand stow bar** of the **Viewer**. This will **automatically insert an Edit SOP** into the network.

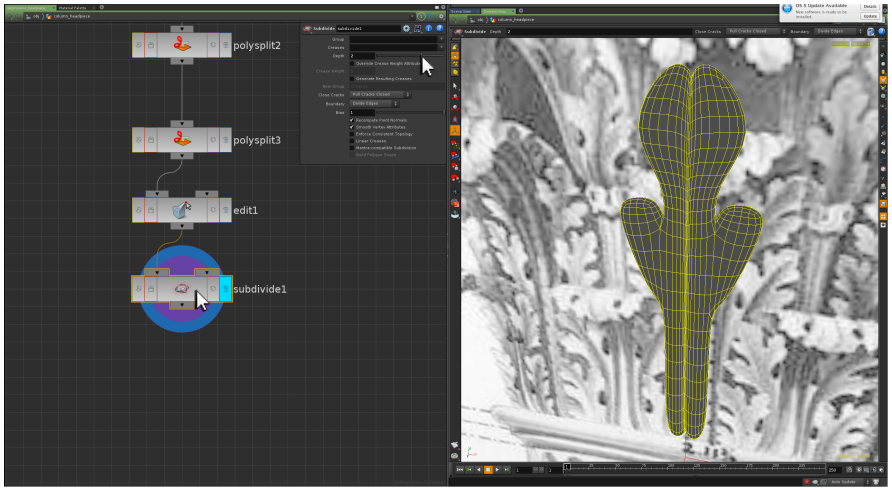


Using the Scale handles, scale down this central ring of edges to create an indentation in the filigree shape.

IMPORTANT NOTE: The Scale Tool (or any Transform Button Mode) will remain active in the Viewer even if a new operation is created in the Network Editor. Pressing ESC + ENTER with the mouse over the Viewer can reset the Viewer mode to any new operator created this way.

SUBDIVIDING THE SHAPE

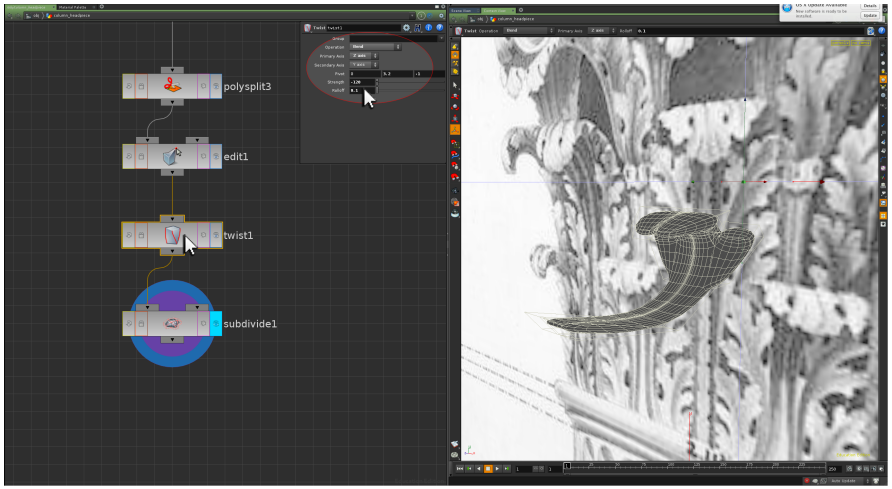
As with the seahorse example, when the basic polygonal geometry is established it can be subdivided to produce a smoothed version of it. This can be done either by appending a **Subdivide SOP** directly in the **Network Editor**, or by **selecting all the geometry** in the **Viewer** and pressing **TAB – subdivide**.



The nodes higher in the network can also be revisited with a Subdivide node in place. This will create a preview cage of the non-subdivided geometry around the smoothed version, allowing for other shape editing to take place.

BENDING THE SHAPE

Insert a Twist SOP before the Subdivide SOP. This operator can be used to fold over the shape to create the bending filigree leaf shape. x



In the **parameters** for the **Twist SOP** specify:

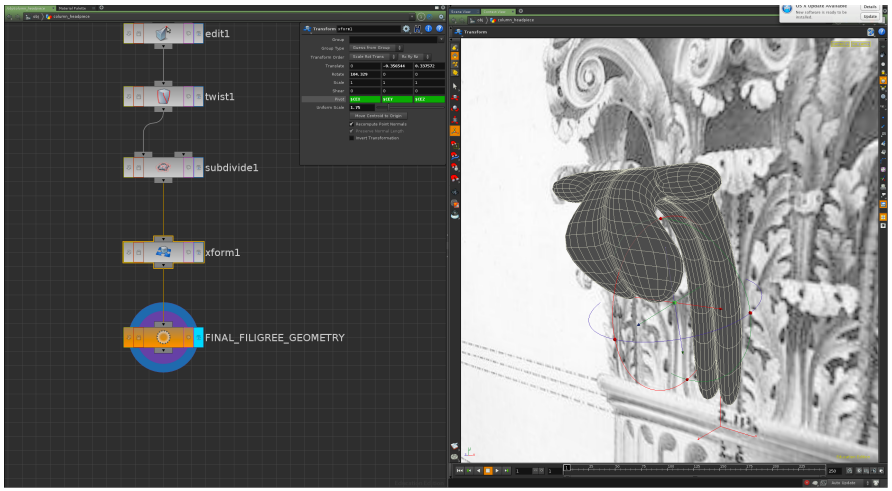
Operation	**Bend**		
Primary Axis	**Z axis**		
Secondry Axis	**Y Axis**		
Pivot	**0**	**3.2**	**-1**
Strength	**-120**		
Rolloff	**0.1**		

TRANSFORMING THE SHAPE

This curved filigree can now be transformed back into position relative to the reference image. **To the output of the Subdivide SOP append a Transform SOP.** In the **parameters** for the **Transform SOP** specify:

Pivot	**$CEX**	**$CEY**	**$CEZ**

This will automatically set the pivot position of the Transform handle to the centre of the geometry.



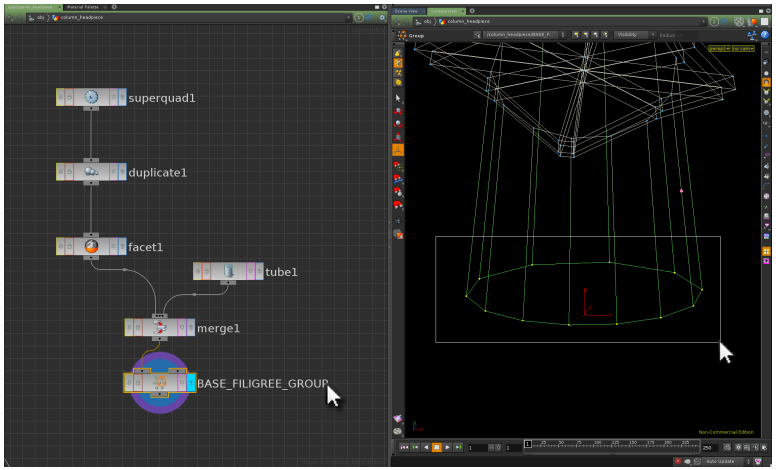
The filigree shape can then be rotated and scaled to match the reference image.

SEE FILE temple_stage9.hipnc

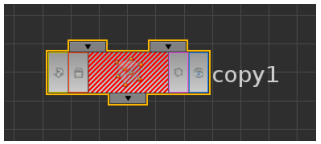
NOTE: This filigree shape can also be saved out to disk as a .bgeo file. Doing so can save Houdini having to process all of its construction nodes when the scene gets rendered.

COPYING GEOMETRY ON OBJECTS

Reactivate the **Display/Render Flag** for the main **headpiece network Merge SOP**, and in the **Viewer** select all the **points** around the **base of the support** and group them using a **Group Geometry SOP**.



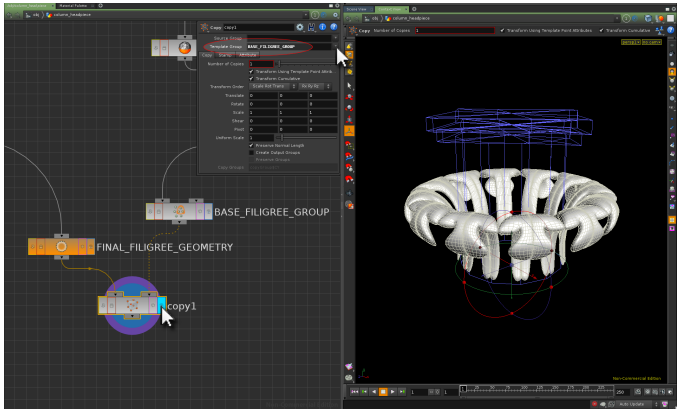
Rename the Group Geometry node to **BASE_FILIGREE_GROUP**. A Copy SOP can now be used to copy the filigree geometry onto the grouped points.



In the **Network Editor**, create a **Copy SOP** as a new node. Do not activate its Display or Render Flag; and ignore the Red Error Striping.

A Copy SOP has **two inputs**; the first is the geometry to be copied; the second is an optional input for points to copy the geometry onto.

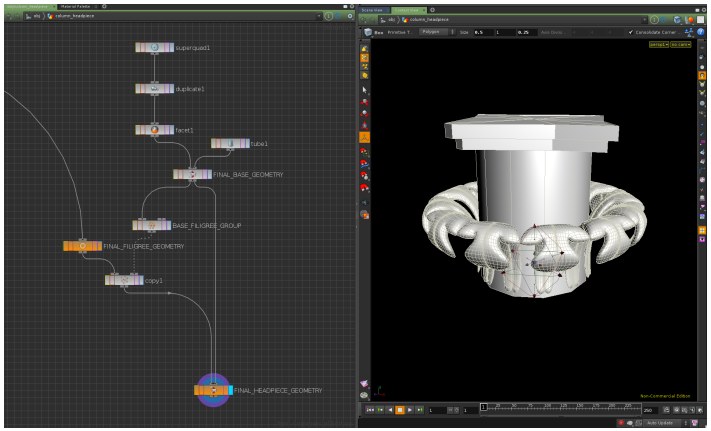
Wire the **output** of the **FINAL_FILIGREE_GEOMETRY** network as the **first input** of the **Copy SOP**. Wire the **output** of the **BASE_FILIGREE_GROUP** node as the second input. This will copy the filigree geometry onto all points (which may take time to calculate).



In the **parameters** of the **Copy SOP**, set the **Template Group Parameter** to **BASE_FILIGREE_GROUP**. This will limit the geometry being copied to only the base points.

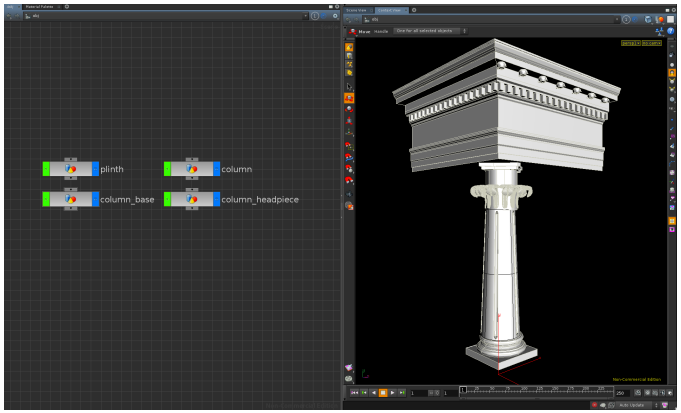
NOTE: Geometry being copied in this way should be facing the Z Axis to copy correctly to the template geometry. If geometry to be copied is not facing the right way, a Transform SOP can be inserted before the Copy SOP's first input.

The components of the headpiece can now be merged together to create the final geometry.



See file temple_stage10.hipnc

The temple components can now be positioned relative to each other at Object Level.



See file temple_complete.hipnc