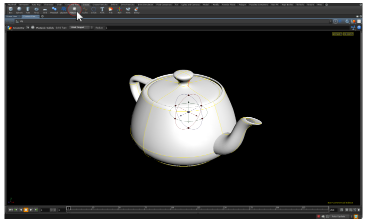
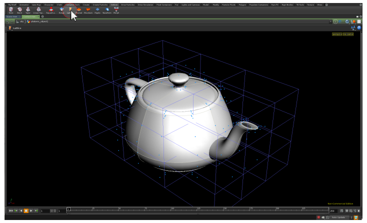
DEFORMERS

Deformers are tools that manipulate geometry in interesting and unusual ways. They often provide part of a solution for Digital Effects work. Good deformations are reliant upon good geometry topology. Among deformations tools are smaller operators such as **Soft Transform SOP**, **Soft Peak SOP**, **Sculpt SOP** and **Twist SOP**, which are also worth exploring.

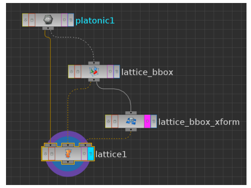
BOUND LATTICE DEFORMATIONS

In a **new Houdini scene**, **maximise** the **Viewer**, and create a teapot by **CTRL + LMB** pressing the **Platonic Solids** button located in the **Create Shelf**. In the **Viewer** parameters, set the **Solid Type** to **Utah Teapot**. Press **i** with the mouse over the Viewer to go to the **Geometry Level** for this object. From the **Deform Shelf**, activate a **Lattice** operation, and when prompted **select all** of the Utah Teapot **points** and press ENTER twice to confirm the operation.

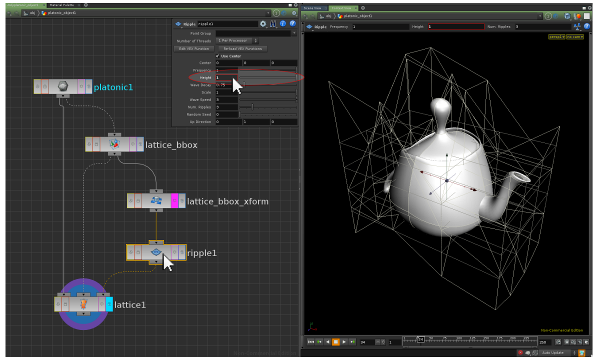




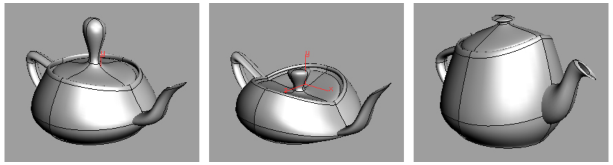
When the Network Editor is examined, a number of nodes have been automatically appended to the Platonic Solids SOP.



Put simply, a Lattice Cage (lattice_bbox) has been created, and wired into the second and third inputs of a **Lattice SOP** (with a Transform SOP also inserted into the third input stream). Any point deformation assigned to the Lattice Cage through the third input of the Lattice SOP, will automatically be inherited by the original geometry coming into the first input.



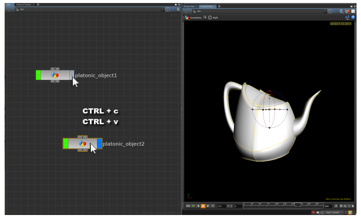
To the **Transform SOP** append a **Ripple SOP**. In the parameters set the **Height** to **0.5**. This will provide some instant animation to the points of the incoming geometry when **PLAY** is pressed.



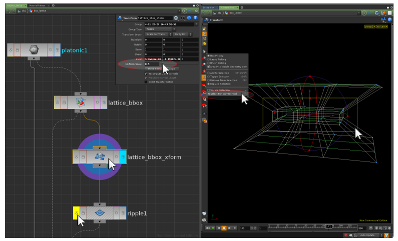
When the visibility flag of the Lattice SOP is turned on, the teapot will deform based on the automatic Ripple SOP animation whilst still retaining its overall structure. A **Null SOP** can be appended to the **Lattice SOP** in order to aid deformation visibility.

BOX LATTICES

A variation on the Bound Lattice example is where the teapot receives its deformation as it passes through the lattice on its way to another location. At **Object Level**, **Copy** (**CTRL + c**) and **Paste** (**CTRL + v**) the first Lattice example to create a second version of it. Turn off the Display Flag for the first lattice example.



At **Geometry Level** for the second lattice example, bypass the Ripple SOP and **select** the **Transform SOP** above it instead. **RMB** on the **Select Arrow**, choosing **Reselect for Current Tool**, ensuring **Point select mode** is activated. In the **Viewer**, **select all** of the **central points** for the **Lattice Cage** and press **ENTER** to confirm their selection.

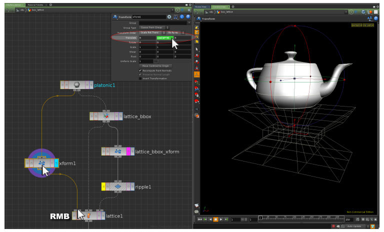


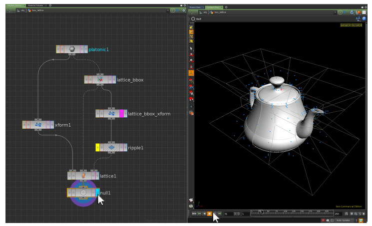
Set the **Uniform Scale** parameter of the **Transform SOP** to **0.5**. This will create a simple funnel to squeeze the teapot through.

RMB on the **first input** of the **Lattice SOP** to **insert append** a **Transform SOP** to the Platonic Solids SOP. In the **parameters** for the **Transform SOP** specify:

Translate	**0**	**cos($F*3)**	**0**

This will create a teapot which cyclically animates in the Y axis, passing through the simple funnel of the lattice cage.



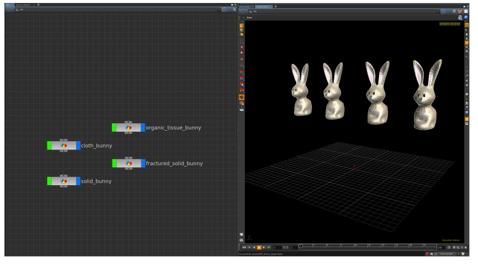


As before, a Null SOP can be appended to the end of the network to better see the deformation effect.

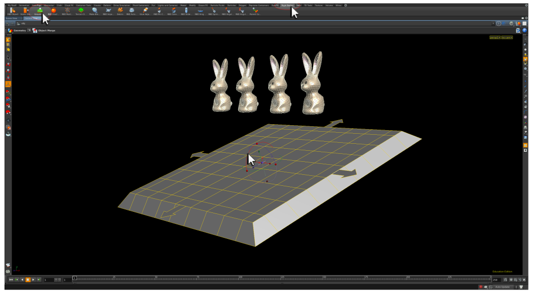
See file bound_and_box_lattices.hipnc

DYNAMIC DEFORMATIONS

The **DOPs Level** of Houdini has a number of deformation tools all driven by the **Finite Element Solver**. This solver can create cloth and organic deformation effects, capable of dynamic interaction with other solvers such as Rigid Bodies, Particles, Volumes and Wires as well as dynamic forces such as gravity. **Open the scene DOP_bunny_begin.hipnc.**

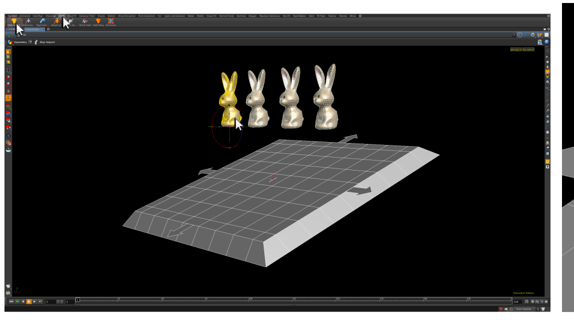


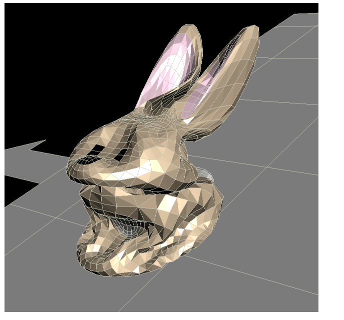
This scene contains four bunny objects positioned above the construction plane that can be used to demonstrate the different Finite Element Solver capabilities. **Maximise** the **Viewer** and **reveal the Shelves**. From the **Rigid Bodies Shelf**, activate the **Ground Plane Shelf Button**. This will create an AutoDopNetwork and a Ground Plane that the deforming bunnies can interact with.



Rotate the **ground plane** object to create a slope.

In the **Viewer**, **select** the **first bunny** and from the **Cloth Shelf** activate the **Cloth Object Shelf Button**.





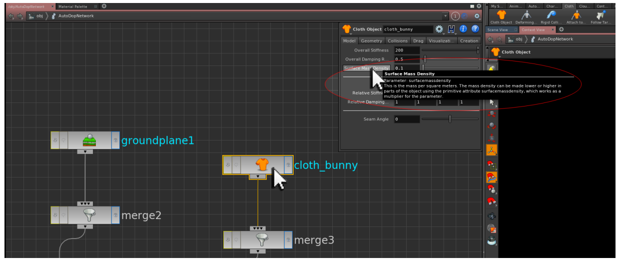
This will convert the first bunny object into a **cloth object** in the **AutoDopNetwork**. When **PLAY** is pressed, the first bunny drops onto the ground plane under the influence of gravity and crumples accordingly. By default, a cloth object is configured to automatically collide with itself.

DYNAMIC CLOTH BEHAVIOUR

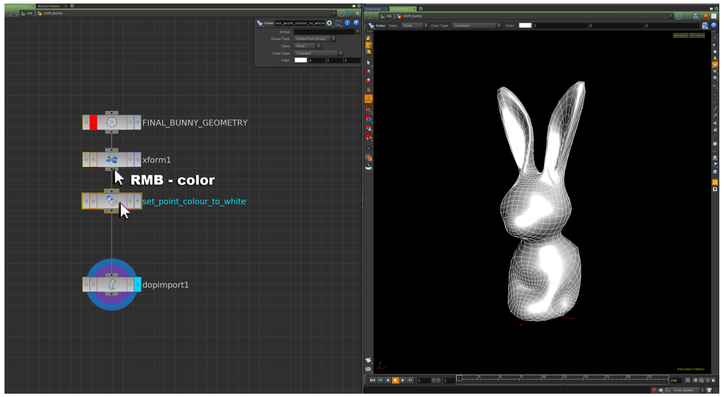
While cloth in the formal sense is normally associated with clothing and drapery, in the context of deformers, cloth objects can also be used to create **crumpling** and **denting effects** akin to **metal objects** (if its dynamic cloth object parameters are set to a higher than normal values).

All **Dynamic Object parameters** can also be **overridden** as **geometry attributes** if required, so that specific areas of the geometry can behave differently to other areas. Go **inside** the **AutoDopNetwork** and locate the **Cloth Object DOP** reading in the cloth_bunny object.

Hover the **mouse over** the **Model > Surface Mass Density parameter**. This reveals a small Help Card denoting the internal name for this parameter (**surfacemassdensity**), as well as instructions about how it can be overridden as a **primitive attribute**.

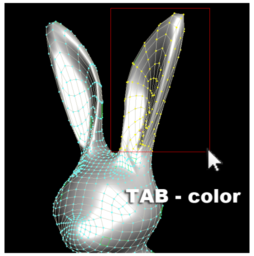


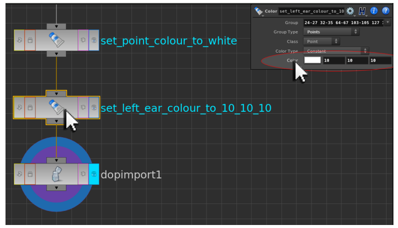
NOTE: A full list of associated dynamic attributes can be found in the parameter pane **Help Card** for any Object DOP node.



Go to the **Geometry Level** for the **cloth_bunny object** and **RMB insert** a **Color SOP before** the automatically generated **DOP Import SOP**. By default, this will set the base point colour of the bunny geometry to white.

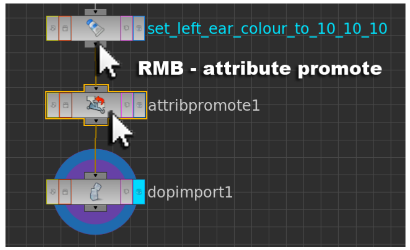
In the **Context Viewer**, **select** all of the **points** of the bunny geometry's **left ear**, and interactively create a second **Color SOP**. In the **parameters** for this second Color SOP, specify a **Color** value of **10, 10, 10**.





This will create a higher colour value for only the left ear that when combined with the white base colour of the geometry, can be used as a multiplier for the **Surface Mass Density parameter**.

NOTE: If necessary, the colour value of the first Color SOP can be temporarily decreased to see the effect of the higher colour values assigned to the left ear.

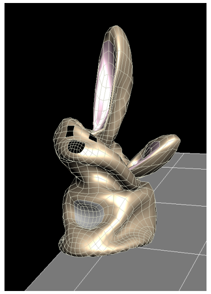


RMB append an **Attribute Promote SOP** to the second Color SOP. This node can be used to convert the **point colour attribute** (**Cd**) into a **primitive attribute** and assign it a **new name**.

In the **parameters** for the **Attribute Promote SOP** specify:

Original Name	**Cd**
New Class	**Primitive**
Change New Name	
New Name	**surfacemassdensity**

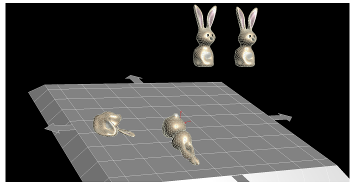
NOTE: As the original bunny geometry colour was specified as a primitive attribute rather than a point attribute, the original bunny colour will reappear as a result of the Delete Original [point attribute] option active on the Attribute Promote SOP.



Now when **PLAY** is pressed, the left ear of the cloth_bunny object is now much heavier than the rest of the bunny geometry. This principle of being able to override parameters using attributes is a key aspect of Houdini workflow, and will be explored further in a later lecture. **See file DOP_bunny_stage1.hipnc**

THE SOLID OBJECT DOP DEFORMER

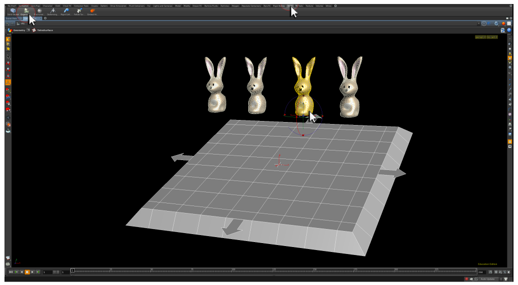
Maximise the **Viewer**, and select the **second bunny object**. From the **Solid Shelf**, activate the **Solid Object** button. This will create a **semi-rigid object** again driven by the **Finite Element Solver**. When **PLAY** is pressed, the second bunny also drops onto the ground plane, this time as a rubbery object.

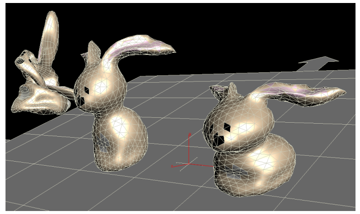


NOTE: The Solid Object shelf tool will automatically rebuild the bunny geometry topology to work correctly with the **Finite Element Solver**. These nodes are automatically assigned at Geometry Level, and can be modified to increase the accuracy of the topology rebuild.

THE ORGANIC TISSUE OBJECT DOP DEFORMER

Rewind the **simulation** and **maximise** the **Context Viewer**. **At Object Level, select** the **third bunny** object, and from the **Solid Shelf** activate the **Organic Tissue** shelf button. This will convert the bunny geometry into a more organic rubbery surface.



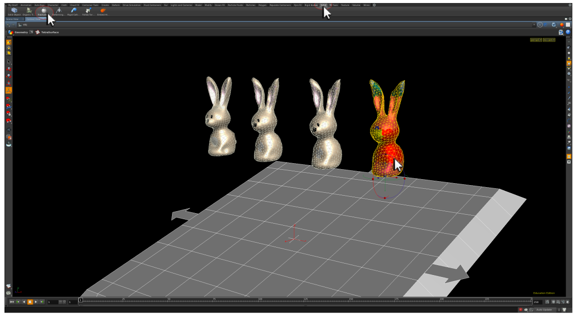


When **PLAY** is pressed the effect of the default organic tissue settings can be seen.

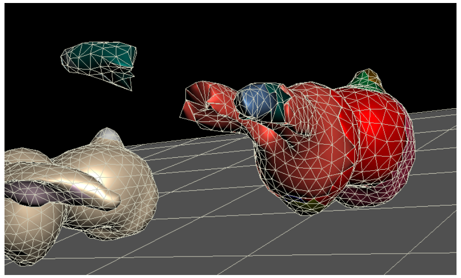
THE FRACTURED SOLID OBJECT DOP DEFORMER

Finally, return to **Object Level** and **select** the **fourth bunny object**. From the **Solid Shelf**, activate the Fractured Solid Object shelf button. This will create an object that is both rubbery and capable of breaking apart.

When this tool is activated, the bunny geometry will become coloured to represent the areas of the geometry that will fracture.



Go inside the **AutoDopNetwork** and **specify** a **Fracture Threshold** value of **0.00000075** for the **fractured_solid_bunny node**. When **PLAY** is pressed, the effect of this tool can be seen.

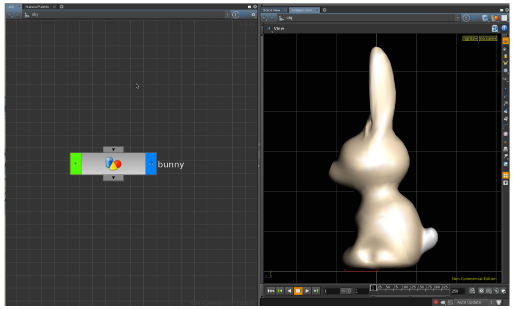


NOTE: Aside from the cloth_bunny, the solid_bunny, organic_tissue_bunny and fractured_solid_bunny are the same Solid Object DOP configured to different settings.

See file DOP_bunny_complete.hipnc

BONES ON A CURVE

One useful deformation technique is the ability to generate bones along a predefined curve. This technique passes deformations of a curve onto a bone chain, and then from the bone chain onto any captured geometry associated with them. Open the scene **bunny_bones_begin.hipnc**

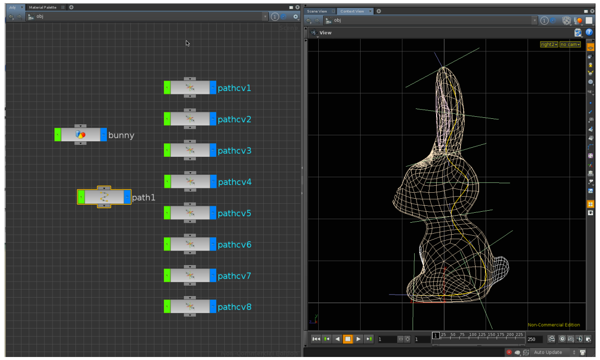


This scene contains a simple piece of geometry to demonstrate the Bones from Curve operation. With the mouse over the View Pane press **SPACEBAR + 4** to switch the view from **Perspective** to **Right Orthographic View**.

CURVES VS PATHS

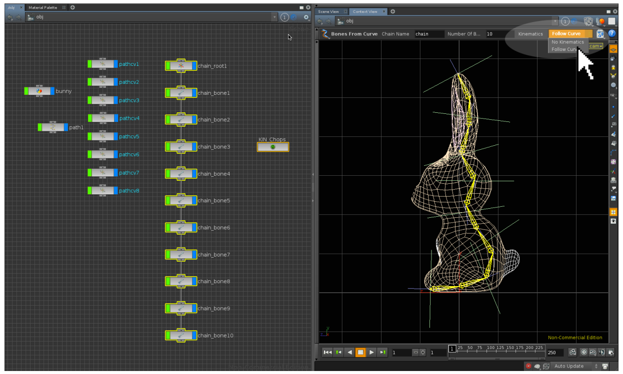
With the mouse over the **Viewer**, press **TAB** and type **Path (Utility)**. A Path is an Object Level operation where a series of Null Objects (Path CVs) are created and then Object Merged into a separate Path Object to create a curve. This is similar to a Curve SOP. The disadvantage of Paths over Curves is the number of objects generated as a result; however Path CV objects can act as keyframe-able animation controllers for positioning the points of the curve.

Switch to a Wire-frame view and **LMB** draw Path CV points along the bunny. A Path curve will automatically be generated between each Path CV point.

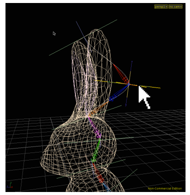


GENERATING THE BONES

In the **Network View**, select the **path1 Object** and with the mouse over the **View Pane** press **TAB** and type **Bones From Curve**. This will by default generate 10 bones along the length of the curve.

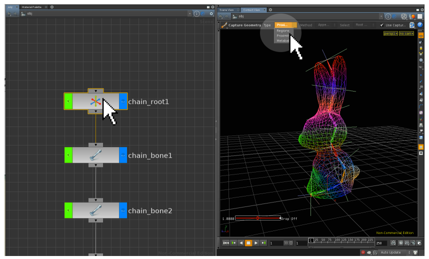


With the **Bones From Curve** tool mode still active, set the **Kinematics** parameter located at the top of the View Pane from **No Kinematics** to **Follow Curve**. This will automatically create a **CHOP Network** in the **Network View** that locks the movement of the bones to the position of the path curve. This can be verified by using the **Pose Tool** (TAB – pose) in the View Pane to transform one of the **PathCV** Objects.

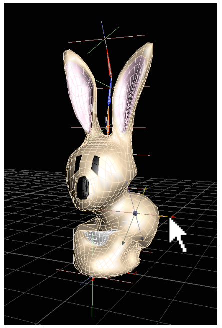


CAPTURING THE BUNNY GEOMETRY

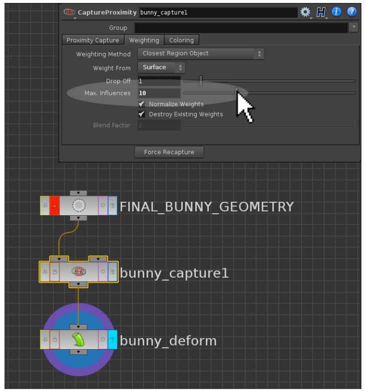
Undo any PathCV changes, and in the select the **bunny** Object. With the mouse over the **Viewer**, press **TAB** and type **Capture Geometry**. As prompted by the help bar at the bottom of the View Pane, select the chain_root1 object in the **Network View**, and press **ENTER** with the mouse over the **View Pane** to confirm the operation. The geometry will now be captured to the bone chain.



Before exiting the Capture Geometry tool, change the **Type** parameter to **Proximity** (located in the top stow bar of the Viewer) and slide the **HUD** slider in the viewer window all the way to the end. This will create some approximate capture regions for the deformation.



When the **Pose Tool** is reactivated, the bunny geometry will move in accordance to any transformations given to a **PathCV** Object. At present however the deformation of the surface is crude.



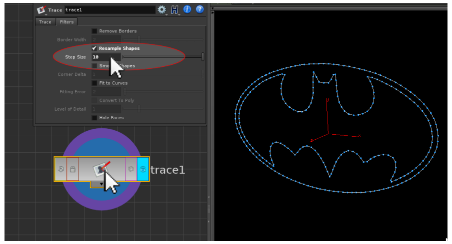
This can be fixed by going to the **Geometry Level** of the **bunny** Object and locating the **Capture Proximity SOP** automatically generated by the Capture Geometry process. Increasing the **Max. Influences parameter** to **10** will increase the number of bones influencing the deformation of the geometry. When the Pose Tool is reactivated at Object Level, the bunny deformation will be much smoother. See **bunny_bones_end.hip**

WIRE DEFORMATIONS

In a new Houdini scene, generate a **Grid Object** and go inside it to Geometry Level. In the **parameters** for the Grid SOP, specify:

Size	**1.5**	**1.5**
Rows	**100**	
Columns	**100**	

Create a **Trace SOP** alongside the Grid SOP. This operator will read in a simple graphic or sequence of images and generate a curve from it. In the **parameters** for the **Trace SOP**, set the **Image Input** parameter to point to the **batman_logo.jpg**.

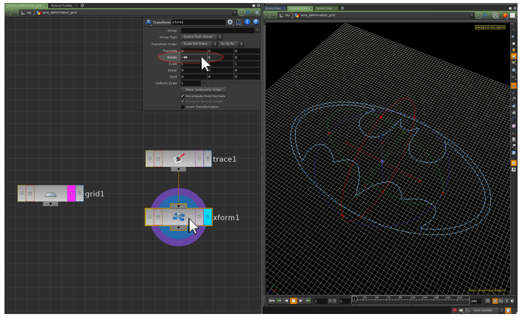


As the number of points generated by the Trace SOP is very high, the curve can be resampled by activating the **Resample Shapes** option located in the **Filters** section of the parameters. Increasing the **Step Size** parameter to **10** will create a more usable curve.

To the **Trace SOP** append a **Transform SOP**. In the **parameters** for the Transform SOP specify:

Rotate	**-90**	**0**	**0**

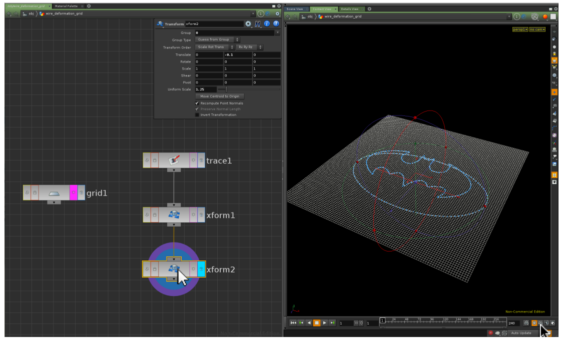
This will align the logo to the grid.



To the output of the **Transform SOP** append a **second Transform SOP**. In the **parameters** for this node specify:

Group	**0**		
Translate	**0**	**-0.1**	**0**
Uniform Scale	**1.25**		

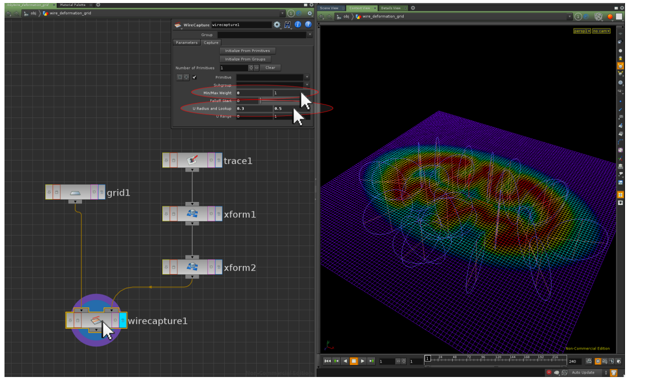
This will transform only the outer curve of the logo (primitive number 0), moving it away from the inner curve and increasing its size.



WIRE DEFORMER SETUP

In order to deform geometry based upon curves, the geometry has to be captured relative to the deformation curves. This is done by using a **Wire Capture SOP**.

In the Network Editor create a **Wire Capture SOP** and assign the **output** of the **Grid SOP** as its **first input** and the **output** of the **second Transform SOP** as its **second input**. Pressing **ENTER** with the mouse over the Viewer will activate the Wire Capture SOP visualization.



Under the **Capture** section of the Wire Capture's **Parameters**, specify:

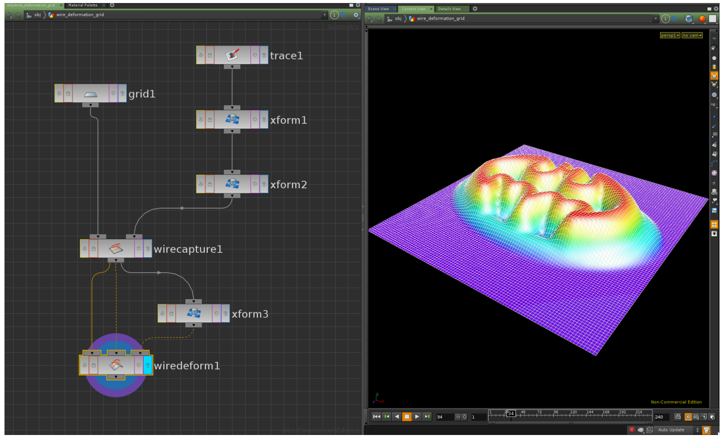
Min/Max Weight	**0**	**1**
U Radius and Lookup	**0.3**	**0.5**

This will reduce the size of the capture circle radius to something making the logo more legible.

NOTE: The scale of each circular capture region can also be scaled relative to its position along the curve. **This is done by key-framing the Lookup value**.

WIRE DEFORM SOP

To the **Wire Capture SOP** append a **Wire Deform SOP**. Wire the output of the Wire Capture SOP as both the first, second and third input. **RMB** on the **third input** of this operator will allow for the insertion of a **third Transform SOP**.



As with Lattices; animating, deforming or editing the third input of the Wire Deform SOP will create the wire deformations. In this example, creating a wire based deformation can be done by adding an expression to the inserted **Transform SOP**. In the **parameters** for the inserted Transform SOP specify:

Translate	**0**	**-sin($F*20)/3**	**0**

This will animate the deformation in accordance to a sine wave.

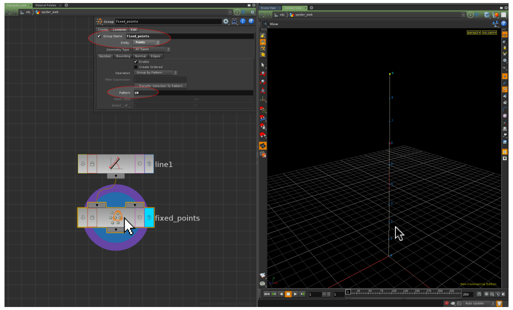
See file Wire_Deformer.hipnc

A DYNAMIC SPIDER'S WEB

In a new Houdini scene, create a **Geometry Object**, and rename it to **spider_web**. At its **SOP Level**, replace the File SOP with a **Line SOP**. In the **Parameters** for the **Line SOP** specify:

Length	**3**
Points	**10**

To the **Line SOP** append a **Group SOP**. Set the **Group Name** to **fixed_points** and the **Entity** parameter to **Points**.

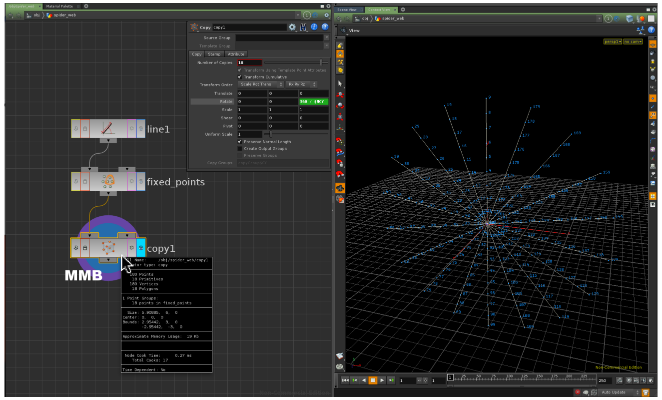


In the **Pattern** parameter enter the expression **$N**. This will procedurally group the very last point of the line.

To the **Group SOP** append a **Copy SOP**. In its **parameters** specify:

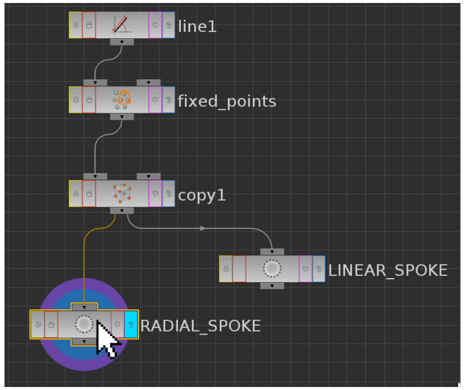
Number of Copies	**18**		
Rotate	**0**	**0**	**360 / $NCY**

This will create the spokes for the web where the rotation of each spoke is automatically governed by the total number of copies (the local variable **$NCY**).



When **MMB** is pressed on the **Copy SOP node**, the numbers of points present in the fixed group has risen from 1 to 18 as a result of each duplicated web spoke.

To the output of the **Copy SOP** append **two Null SOPs** as separate network branches. Rename one of the Null SOPs to **RADIAL_SPOKE** and the second to **LINEAR_SPOKE**.



CREATING THE RADIAL SPOKE

To the **radial_spoke Null SOP** append a **Fuse SOP**. This will fuse together the centre points of the web. To the **Fuse SOP** append a **Delete SOP**. In its **parameters** specify:

Pattern *

☑ **Keep Points**

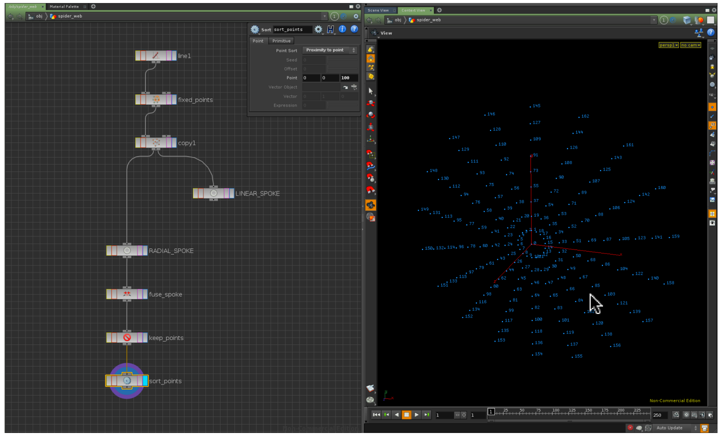
This will delete all of the geometry except for the point information.

To the **Delete SOP** append a **Sort SOP**. In the **Point** section of the **parameters**, specify:

Proximity to Point

Point	**0**	**0**	**100**

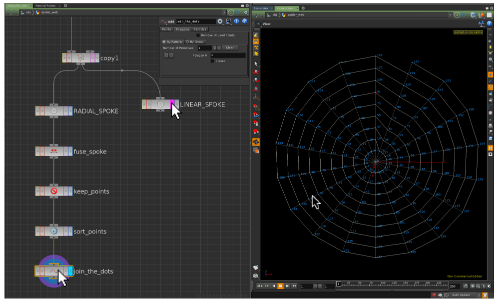
This will perform a radial sorting of the point numbers from the centre outwards forming a spiral.



To the **Sort SOP** append an **Add SOP**. Under the **Polygon** section of its **Parameters** specify:

Polygon 0 *****

The ***** wildcard will draw a polygon line between all of the points. Activating the pink **Template Flag** of the **linear_spokes Null SOP** will help visualise the spider's web.

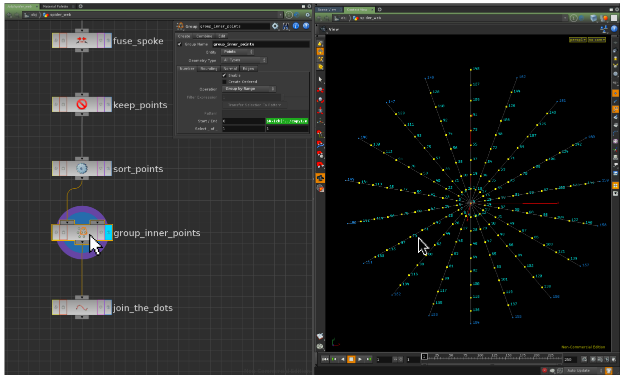


CONTROLLING THE DRAWN POINTS

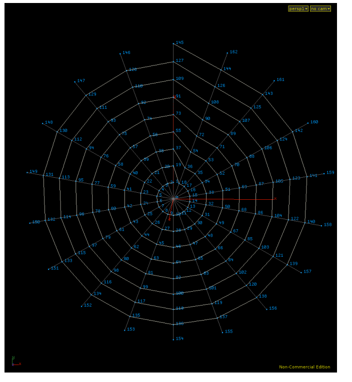
At present the polygon line is being drawn through all of the points. A spider web would however need the outer points to attach to the environment. An expression can be created that will automatically group the entire inner web points, no matter how many spokes are assigned to the web.

Between the **Sort SOP** and the **Add SOP** insert a **Group SOP**. In its **Parameters** specify:

Group Name	**group_inner_points**	
Entity	**Points**	
Operation	**Group by Range**	
Start / End	**0**	**$N-(ch("../copy1/ncy")-1)**
Select_of_	**1**	**1**



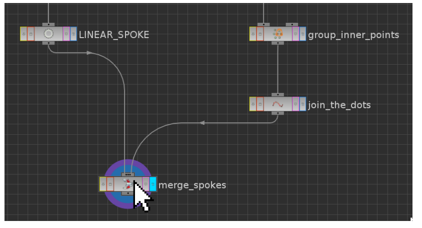
This will procedurally group all of the points that will make up the radial spoke. The **$N – (ch("../copy1/ncy")-1)** expression returns the total number of points minus the total number of copies specified in the Copy SOP. Visually this equates to all of the points being grouped apart from all but one of the anchor points.



The **group_inner_points** can now be referenced in the **Add SOP**.

COMPLETING THE WEB CONSTRUCTION

The two network chains can now be reunited using a **Merge SOP**. For this example, the order in which the inputs of the merge occur is important. To the output of the **linear_spokes Null SOP** append a **Merge SOP**. Set the output of the **Add SOP** of the radial_spoke network chain as its second input.



The correct input ordering for the Merge SOP will ensure the correct point order will occur after these merged spokes are fused together.

As a final step append to the **Merge SOP** a **Fuse SOP** to fuse any overlapping points together. The construction of the spider's web is now complete. See **spiderweb_stage1.hipnc**

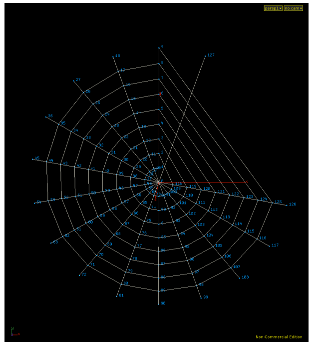
CREATING AN ANIMATED WEB

Before making the web dynamic in terms of movement, the web itself can be grown by animating an inverted deletion of the points. This can be done by utilising a Delete SOP.

To the **Fuse SOP** append a **Delete SOP**. In its parameters specify:

Entity	**Points**	
Operation	**Delete by Range**	
Start / End	**$F / 3**	**$N – 1**
Select _ of _	**1**	**1**

By specifying an increasing Start Parameter ($F / 3) only points above this value will be deleted. As this number increases the number of points being deleted gets less giving the impression of growth.



Increasing the timeline frame range to 500 will allow for all the web growth to be seen.

DYNAMICALLY ANIMATING THE WEB

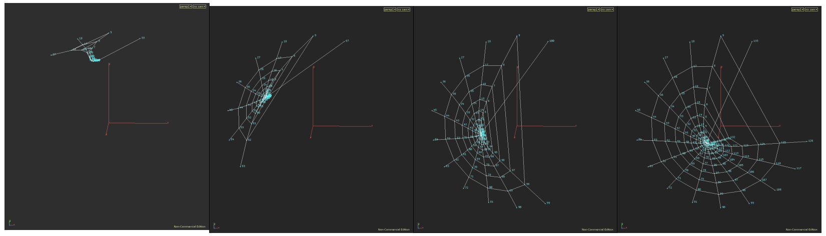
To the **Delete SOP** append a **Spring SOP**. This node can be utilised to add simple dynamics to the web. Under the **Forces** section of its Parameters, specify:

External Force	**0**	**-1**	**0**

This will give the web some gravity. Under the **Nodes** section, specify:

Fixed Points	**fixed_points**
Spring Behaviour	**Normalize Displacement**
Initial Tension	**30**

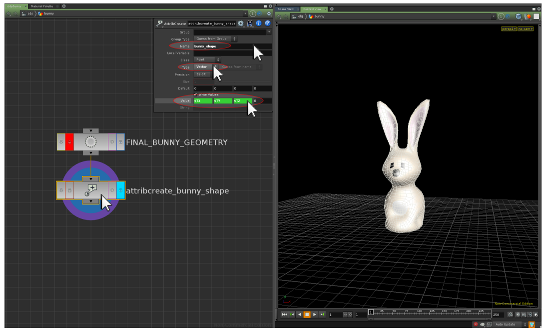
This will ensure that the points grouped at the very start of the web construction will not be dynamic. Now when **PLAY** is pressed the spider's web will appear to be spun.



The render aesthetic of the web can be controlled by appending a **Point SOP** to the Spring SOP. Under the **Particle** section of the Point SOP **parameters**, set the **Scale** attribute from **Keep Scale** to **Add Scale**. A new scale value can now be set by **RMB** on the **Scale** parameter and choosing **Delete Channels**. A value of **0.01** will produce a thin web when the scene is rendered. **See file spider_web_end.hipnc**

GEOMETRY DISSOLVES

Deformers can also be used as a discreet way to create geometry dissolve effects. **Open the scene bunny_dissolve_begin.hipnc**



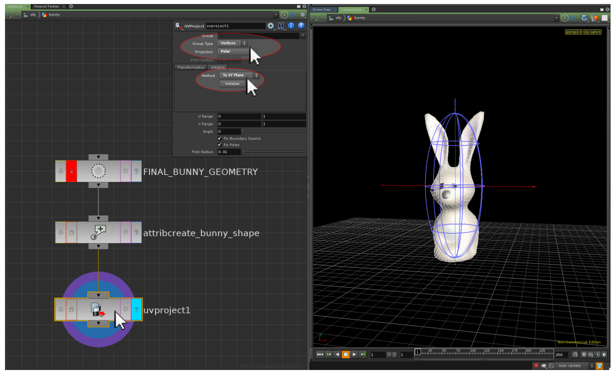
At the **Geometry Level** for the **bunny**, append an **Attribute Create SOP** to the locked Null SOP. In the **parameters** for the **Attribute Create SOP** specify:

Name	**bunny_shape**		
Local Variable	**BUNNY_SHAPE**		
Type	**Vector**		
Value	**$TX**	**$TY**	**$TZ**

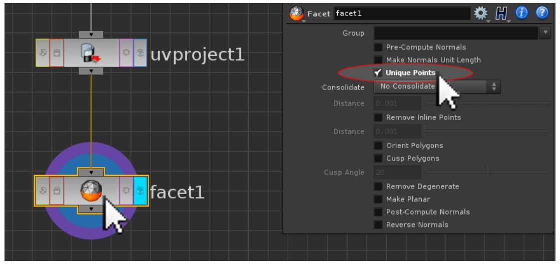
This will store the point position of the bunny geometry as a custom attribute. **Append** to the Attribute Create SOP a **UV Project SOP**. In the **parameters** for the UV Project specify:

Group Type	**Vertices**
Project	**Polar**
Initialize >	
Method	**To XY Plane**

Press the **Initialize button** to complete the UV Projection.



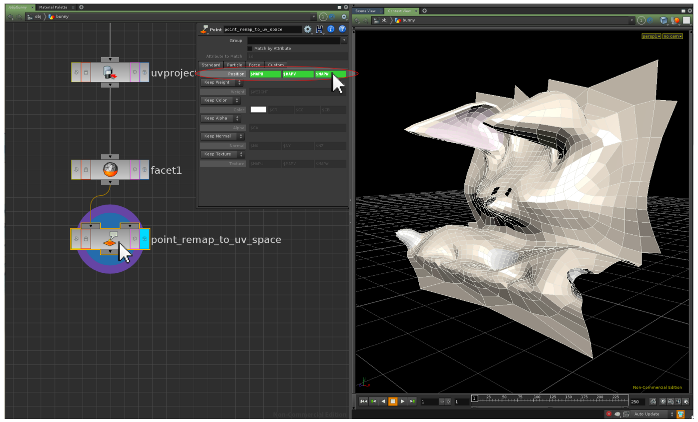
This UV information can be used to initiate a Dissolve effect. To the **output** of the **UV Project SOP append** a **Facet SOP** specifying **Unique Points** in its **parameters**.



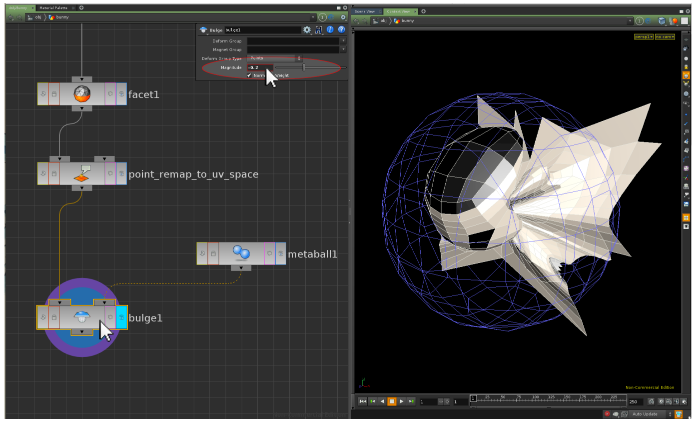
Append to the Facet SOP a **Point SOP**. In the **parameters** for the **Point SOP** specify:

Position **$MAPU** **$MAPV** **$MAPW**

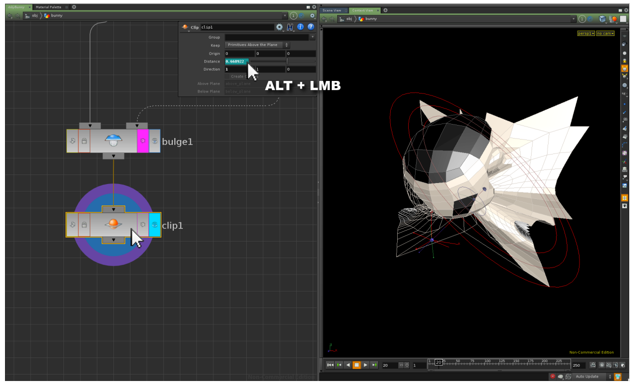
This will project the geometry points of the bunny into UV Space creating a static mesh that can be dissolved. This technique is very helpful for creating consistent dissolve effects on animating geometry.



This UV Space geometry can be deformed even further by using a **Bulge SOP**, with a **Metaball SOP** set as its second input.



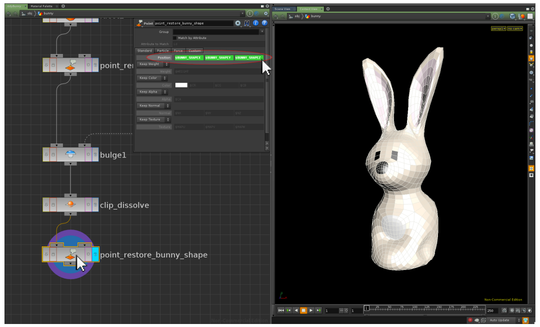
Increasing or **decreasing** the **Magnitude parameter** of the **Bulge SOP** will adjust the amount of bulge taking place.



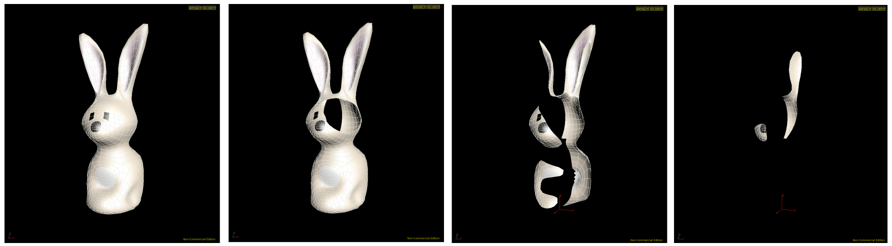
A **Clip SOP** can then be added to the bulged geometry in order to cut it away. **Keyframes** can be set over time on the **Distance parameter** in order to dissolve the geometry away.

A **Point SOP** can then be used to restore the original bunny geometry shape. In the **parameters** of the **Point SOP** specify:

Position $BUNNY_SHAPEX $BUNNY_SHAPEY $BUNNY_SHAPEZ



A **Fuse SOP c**an also be appended to the network to restore smooth shading of the bunny geometry. When **PLAY** is pressed, the bunny dissolves in an interesting way over its surface.



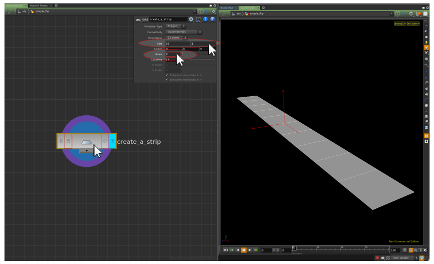
See file bunny_dissolve_end.hipnc

PRIMITIVE SOP UNFOLDING

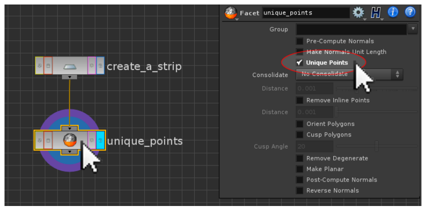
In a new **Houdini scene**, create a **Grid Object**, and at **Geometry Level** specify in the **Grid SOP parameters**:

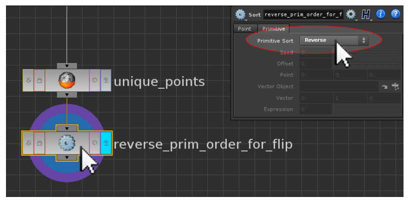
Size	**10**	**1**
Rows	**2**	

This will create a strip of Polygons on which functionality of the Primitive SOP can be demonstrated.



Append to the **Grid SOP** a **Facet SOP**, activating **Unique Points**. This will allow for each primitive face to be animated individually.

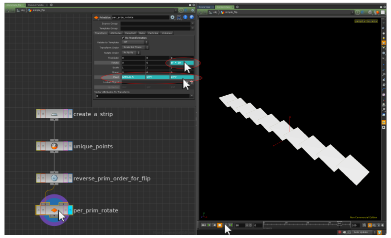




To the **Facet SOP** append a **Sort SOP**. In its **parameters** specify a **Primitive Sort** of **Reverse**.

To the Sort SOP append a **Primitve SOP**. In its **parameters** specify:

☑	**Do Transformation**		
Rotate	**0**	**0**	**$F * 20**
Pivot	**$CEX-0.5**	**$CEY**	**$CEZ**



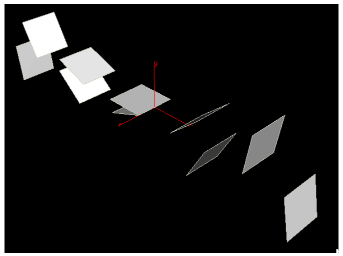
When **PLAY** is pressed, each face of the strip rotates simultaneously.

CREATING AN OFFSET

Modify the expression in the **Rotate Z** parameter of the Primitive SOP to:

Rotate	**0**	**0**	**($F-$PR)*20**

When **PLAY** is pressed, each face now rotates independently offset by its primitive number.

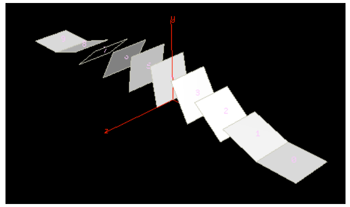


THE CLAMP FUNCTION

In order to create a folding effect, the expression can be wrapped inside a Clamp() Function. This can be used to declare limits of rotation for each face. Modify the **Rotate Z expression** to:

Rotate	**0**	**0**	**clamp(($F-$PR)*20,0,180)**

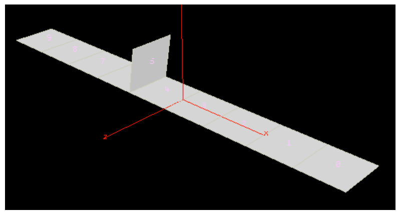
This will limit each primitive face rotation between 0 and 180 degrees. Now when **PLAY** is pressed, the primitive faces flip rotate over similar to a deck of cards being flip rolled over.



An illusion of unfolding can be created by modifying the Rotate Z expression to:

Rotate	**0**	**0**	**clamp(($F-($PR*10))*20,0,180)**

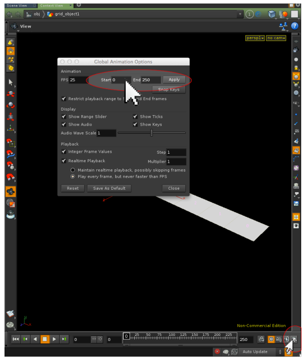
Now when **PLAY** is pressed, a single face appears to rotate walk down the strip of faces.



The illusion of a primitive face unfolding itself can be completed by strategic use of a Delete SOP to remove any faces that have yet to unfold. This will create the effect that the strip is growing as it unfolds.

To begin with, activate the **Global Animation Options**, and increase the Frame Range to:

Start	**0**	**End**	**250**



Rewind the timeline so it starts at **Frame 0**, with the grid strip completely flat. When the **timeline** is **examined**, each of the **primitive face unfolds** takes **10 frames**.

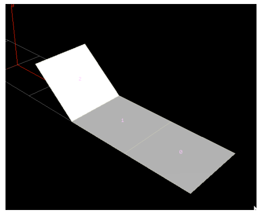
Template the **Primitive SOP** and to it's output append a **Delete SOP**. In the **parameters** for the **Delete SOP** specify:

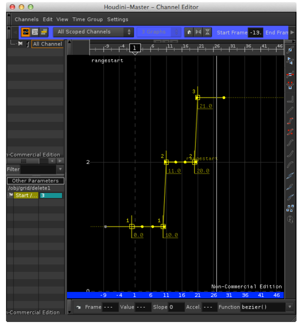
Operation	**Delete by Range**	
Start / End	**0**	**$N**
Select _of_	**1**	**1**

This will initially delete all the primitives; however when the **Start parameter** of the Range is **keyframed**, the deleted faces can be made to reappear. Use the following keyframe values for the Start parameter:

Frame	**0**	**1**
Frame	**10**	**1**
Frame	**11**	**2**
Frame	**20**	**2**
Frame	**21**	**3**

This will create a step based animation curve, giving the appearance that the unfolding primitive face is leaving a trail of primitives behind it.





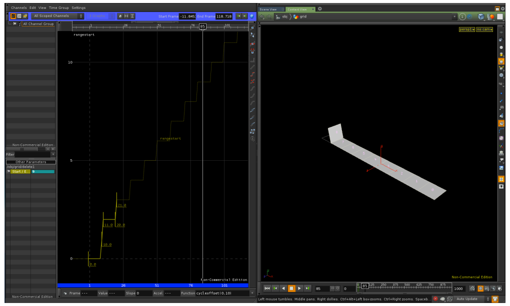
Scope the Parameters for the **Range Start Parameter** in order to see the keyframes set. **Select all** of the **keyframes**, and **modify** the **Function** of the animation curve from **bezier()** to:

Function	**cycleoffset(0,10)**

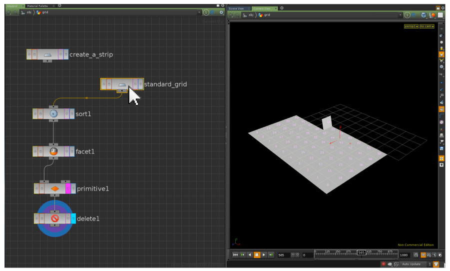
The cycleoffset() animation function is a special animation curve function that will cycle any keyframes set between the two cycleoffset frames specified.

IMPORTANT NOTE: This should be manually typed into the Function parameter rather than selecting cycleoffset() from the animation curve type list.

NOTE: A similar animation curve function called **cycle** also exists that will cycle keyframe curves without offsetting.



When **PLAY** is pressed, all of the unfolding occurs along the strip due to the cycleoffset function. The Channel Editor reveals the cycle values using a dotted line that appears after the set keyframes.



If a standard Grid SOP is then fed into the start of the network and the timeline increased to 0 to 1000, this unfolding primitive effect reveals all of its faces. **See file simple_flip.hipnc**

MAGNET DEFORMATIONS

In a new Houdini scene, create a **Geometry Object** and rename it to **magnet_defomer**. At its **SOP Level**, create a **Grid SOP**. In the **Parameters** for the Grid SOP specify:

Rows	**20**
Columns	**20**

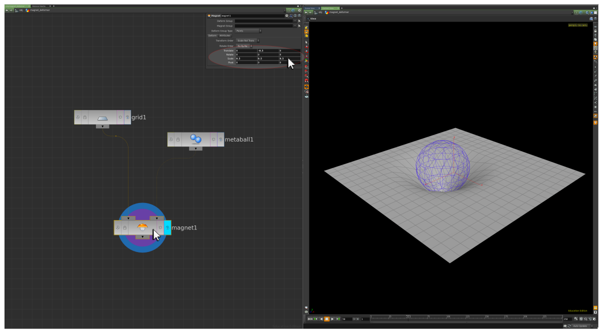
Alongside the grid, create a **Metaball SOP**. In its **Parameters** specify:

Radius	**3**	**3**	**3**
Center	**cos($F * 2) * 5**	**0**	**0**

This will create an animated metaball that passes through the grid geometry.

Create a **Magnet SOP** and wire the output of the **Grid SOP** into its **first** input and the output of the **Metaball SOP** into its **second** input. In the **parameters** for the **Magnet SOP** specify:

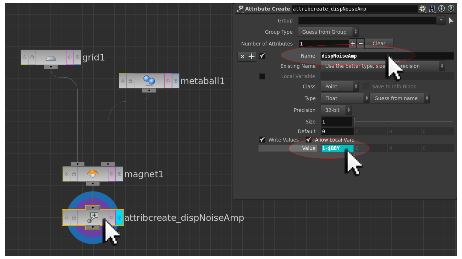
Translate	**0**	**-0.5**	**0**
Scale	**0.5**	**0.5**	**0.5**



This will create a pinched surface for the grid that can be used to drive a displacement effect.

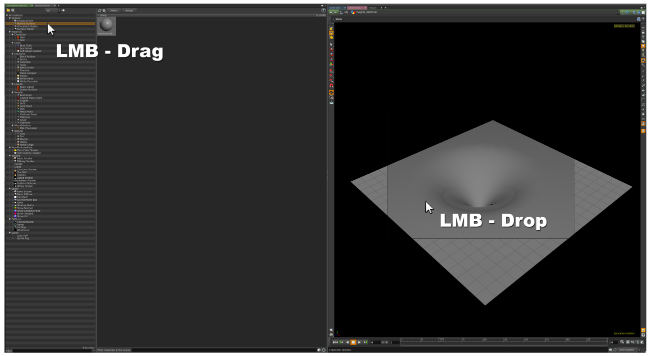
To the **Magnet SOP** append an **Attribute Create SOP**. In its **parameters** specify:

Name	**dispNoiseAmp**
Value	**1 - $BBY**

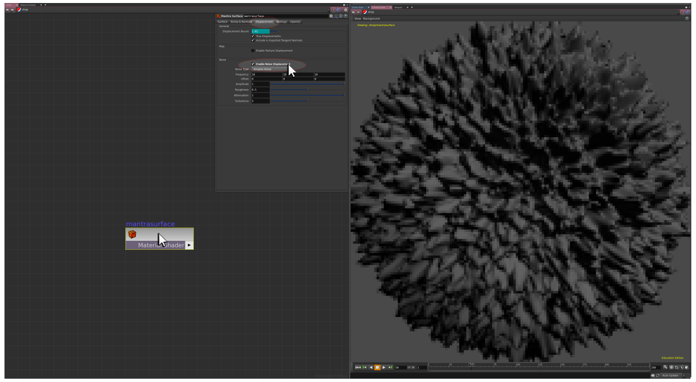


This will create a custom attribute that returns the inverted Bounding Box Y information of the input geometry, where the tip of the pinch will return a value of 1 falling off to a value of 0 towards the un-deformed grid.

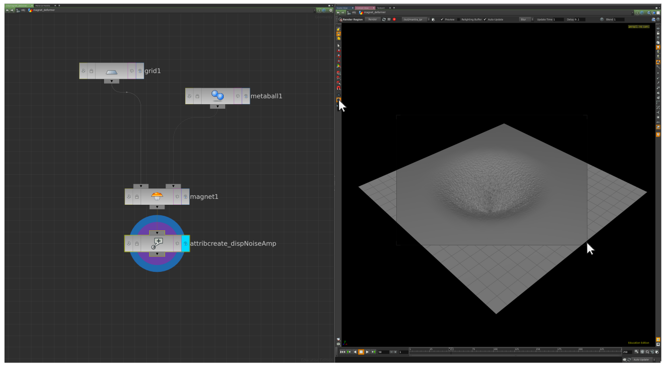
Activate the **Material Palette Tab** over the Network Editor, and **LMB Drag** and **Drop** a **Mantra Surface** material onto the **grid**.



At **SHOP Level**, **activate** the Displacement > **Enable Noise Displacement option** in the **Mantra Surface material parameters**.



When a **Render Region Preview** is activated over the Viewer, the **material displacement effect** takes place **relative** to the **Magnet SOP deformation**.



See file magnet_material_displacement.hipnc.