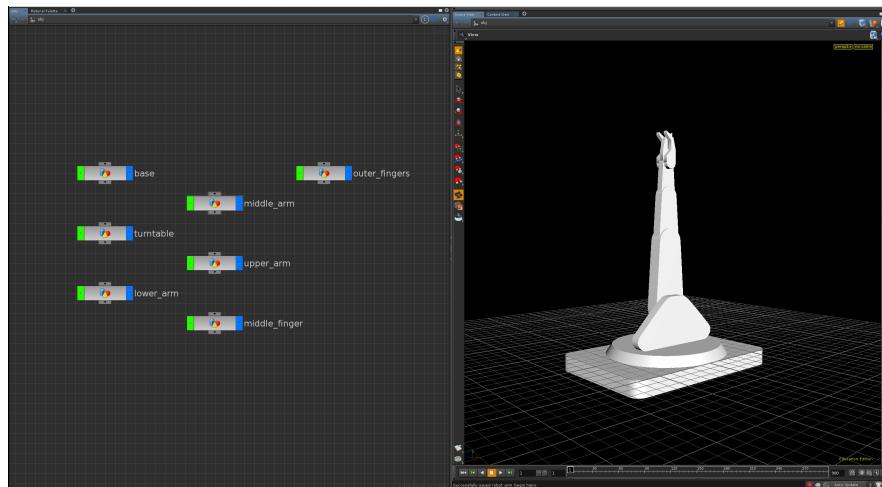


PARENTING OBJECTS

The transformation of one object can be inherited by another object. This is a hierarchical process known as parenting. A parent object passes its transformations to a child object. A child object can also pass down its transformations to other child objects if so required. A child object however cannot pass its transformations back up to its parent object.



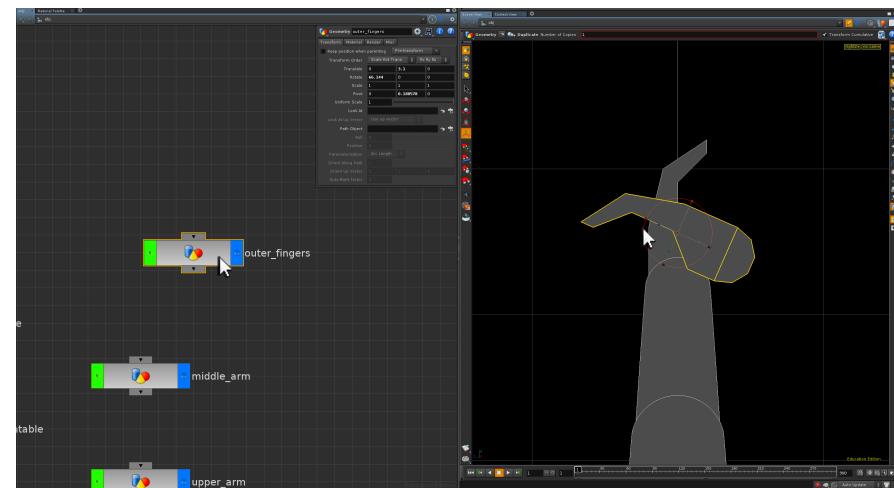
Open the scene **robot_arm_begin.hipnc**. This scene contains a series of independent geometry objects that together form a mechanical arm.

Parenting one object to another takes place at **Object Level** and can be done either interactively in the **Viewer** using the **Parent** tool (**TAB – parent**) or by manually wiring nodes together in the **Network Editor**. If the output of one object (the parent), is wired into the input of another object (the child), transforming the first object (the parent) will result in the same transformations being applied to the second object (the child).

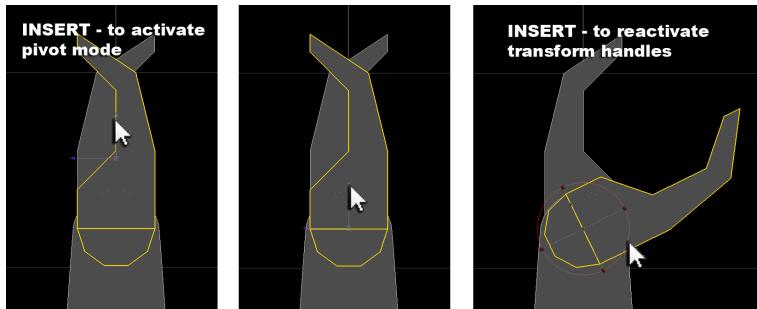
ADJUSTING THE PIVOTS

Each object has its own pivot point, from which any transformation begins. A door for example, opens using hinges on one of its edges. The pivot point of a door is therefore determined by the position of its hinges. For CG objects, a pivot point can be set anywhere in space and affect an object accordingly. For the robot arm, each pivot for each object needs assigning to the most appropriate location for animation purposes.

With the mouse over the **View Pane** press **SPACEBAR + 4** to switch to the Right Orthographic View, and select the **outer_fingers** object. At present, its pivot is set to the centre of the object rather than at its base. If the object is rotated, the object does not move correctly given the context of the object.



Return the **outer_fingers** object to its original position, and press **INSERT** on the keyboard. This will activate the **Pivot Mode** for the object. The pivot for the object can then be moved into the correct position.



With the pivot in the correct position, **INSERT** can be pressed on the keyboard once more to **exit** out of **Pivot Mode**. With the restoration of the Transformation handles, the new pivot position can be tested.

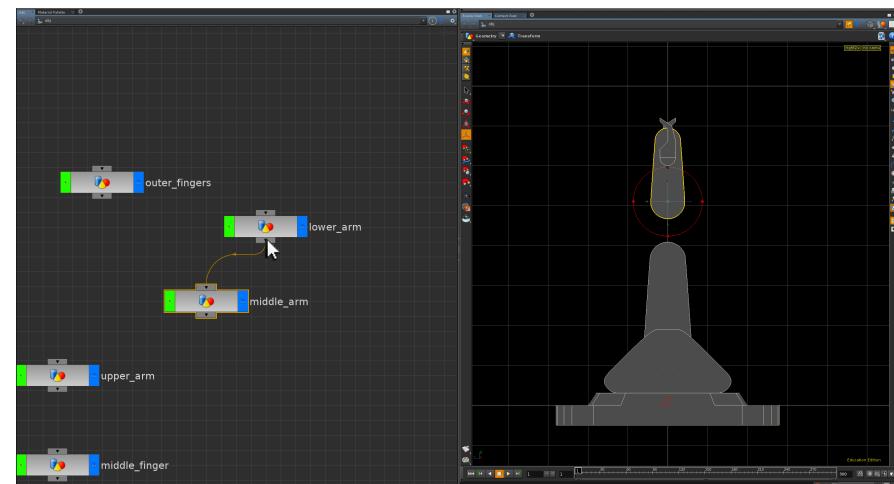


If the Pivot position is correct but the Transform handle still does not align with it, **RMB** on the **Transformation Handle** and from the resulting menu choose **Snap To > Pivot**.

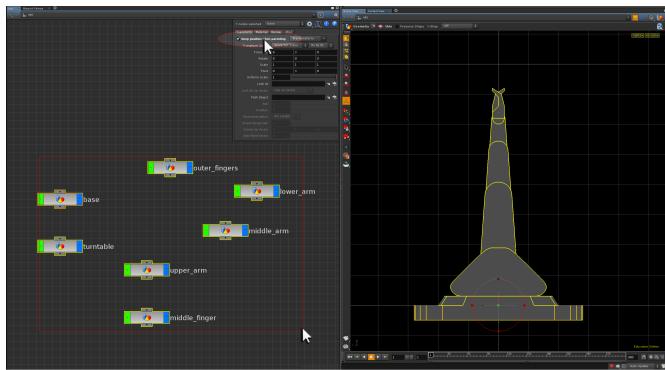
Examine and adjust the Pivots for all objects in this scene, to ensure they are correct for animating the robot arm.

GEOMETRY POSITIONS AND PARENTING

By default, a child object will 'leap' to the transform position of its parent whenever a parent is assigned. For example, if the output of the **lower_arm** object (**the parent**) is wired into the input of the **middle_arm** object (**the child**), the middle_arm object will be translated by its own Translate Y value plus the Translate Y value of the lower_arm object. If the two nodes are disconnected once again, the geometry returns to its original transform position.

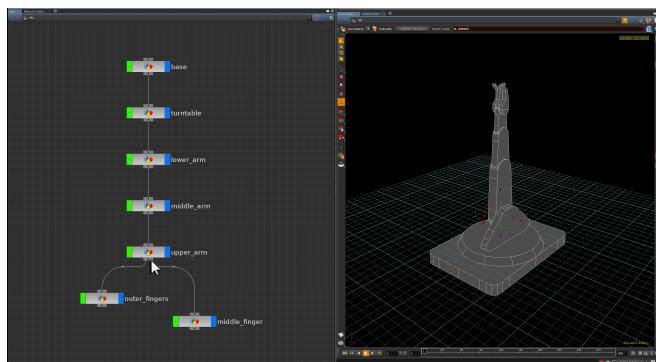


To prevent this behaviour, the **Keep position when parenting** option can be activated simultaneously for every object in a scene by selecting them all, and activating the **Keep position when parenting** option located in the **parameters**. This should be done before any parenting takes place.



The objects in the scene can now be parented to each other without the objects leaping to their parent's position.

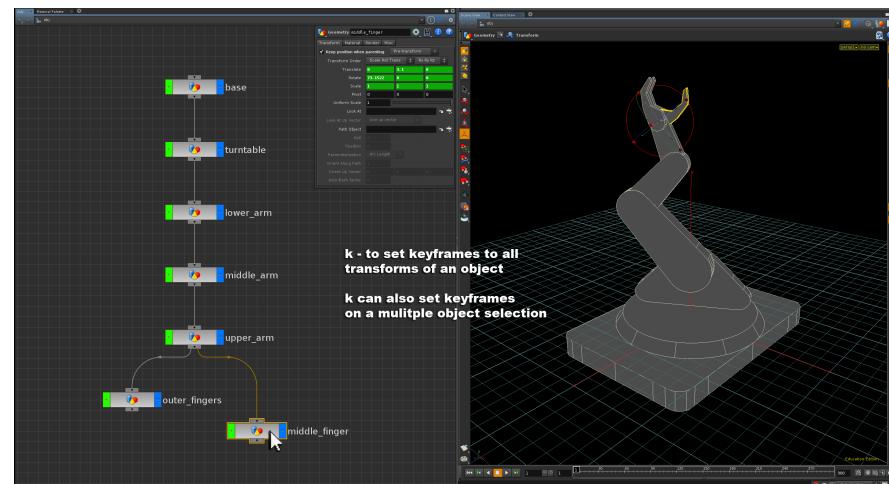
To use the Viewer Parent Tool, press **TAB** and type **parent** with the mouse over the **Viewer**. The **Blue Help Bar** at the base of the interface will ask for the **Child object** to be selected and then **ENTER** to be pressed; it will then ask for the **Parent object** to be selected. It is however just as simple to **parent objects** to each other **directly** in the **Network Editor**. This is simply a case of wiring the appropriate nodes together in the correct order.



Both the **outer_fingers** object and the **middle_finger** object should be children of the **upper_arm** object. L can be pressed on the keyboard with the mouse over the Network Editor to tidy the nodes.

QUICK KEY-FRAMING

With the parenting complete, the robot arm can be animated.

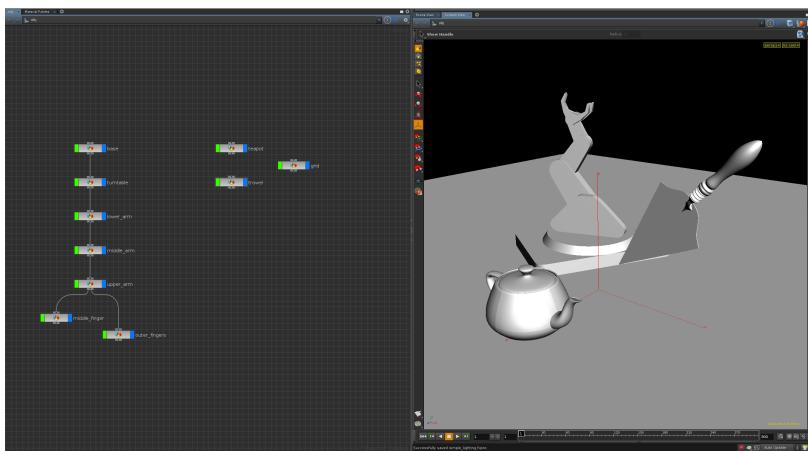


If an object is selected, all of its **Transformation Parameters** can be **simultaneously key-framed** by pressing **k** on the keyboard. This keyboard shortcut allows for quick blocking of animation.

Make a short 250 frame animation for the robot arm using the **k** keyframe hotkey. See file **robot_arm_end.hipnc**

SIMPLE LIGHTING

Open the scene **simple_lighting.hip**. This scene can be utilised for setting up a simple 3 point lighting rig.



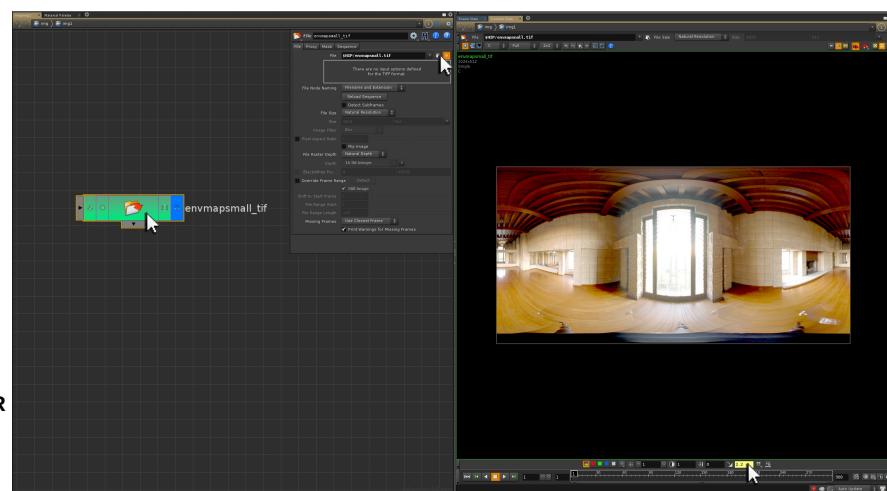
PREPPING BACKGROUND PLATES, TEXTURES AND ENVIRONMENT MAPS FOR A LINEAR WORKFLOW

The NCCA version of Houdini 12.5 has been configured for a **linear workflow**. This means that images being read into Houdini that have gamma natively embedded in them will appear lighter than they do in a normal file browser or image preview application. To make images appear correctly in Houdini, sometimes the embedded gamma needs to be stripped off.

Generally **all 8-bit images** (for example images downloaded from the Internet) have embedded gamma. If you read an image into Houdini and it looks washed out, it means its gamma should be removed.

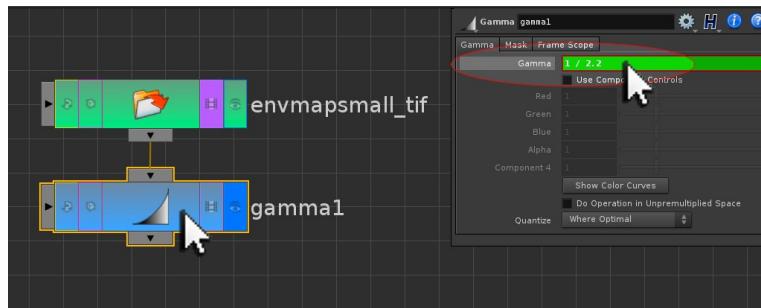
Removing gamma from an image allows for its colour information to be presented as linear for the purposes of correct mathematical processing, with Houdini itself applying a gamma preview to the image to make it look correct.

Head to the **/img Level** of Houdini (Houdini's own compositing level). Inside the default **img1** node, create a **File COP** (Compositing Operator), and read in the **envmapsmall.tif** image. Ensure that the **Gamma preview setting** value at the bottom of the viewer is set to **2.2**.



This setting will assign a gamma of 2.2 to the viewer. As the **envmapsmall.tif** image already has gamma embedded into it, the image appears washed out. To ensure Houdini processes this image correctly; its embedded gamma will need to be removed.

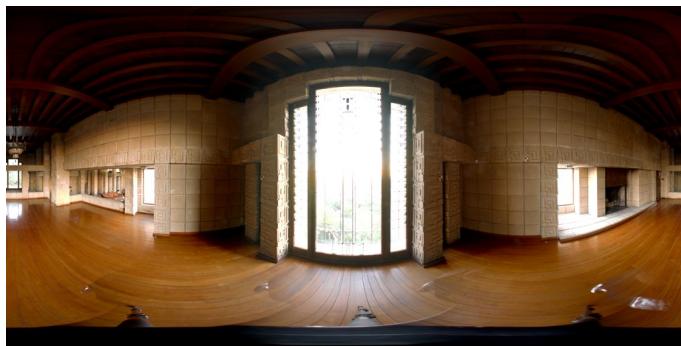
To the output of the **File COP**, append a **Gamma COP**. This node can be used to strip off embedded gamma from an image.



In the **parameters** for the **Gamma COP** specify:

Gamma **1 / 2.2**

When the **Display flag** is activated on the **Gamma COP**, the image presented in the Viewer looks correct relative to how it appears in a normal image preview application.



This is now a linear version of the environment map without any gamma, but with the Viewer adding a gamma preview to it instead.

SAVING LINEAR IMAGES TO DISK

When gamma has been removed from an image; it needs to be resaved out to disk. To the output of the **Gamma COP** append a **ROP File Output COP**. This is the same as a Write node in Nuke, and allows for composited images to be written out to disk.

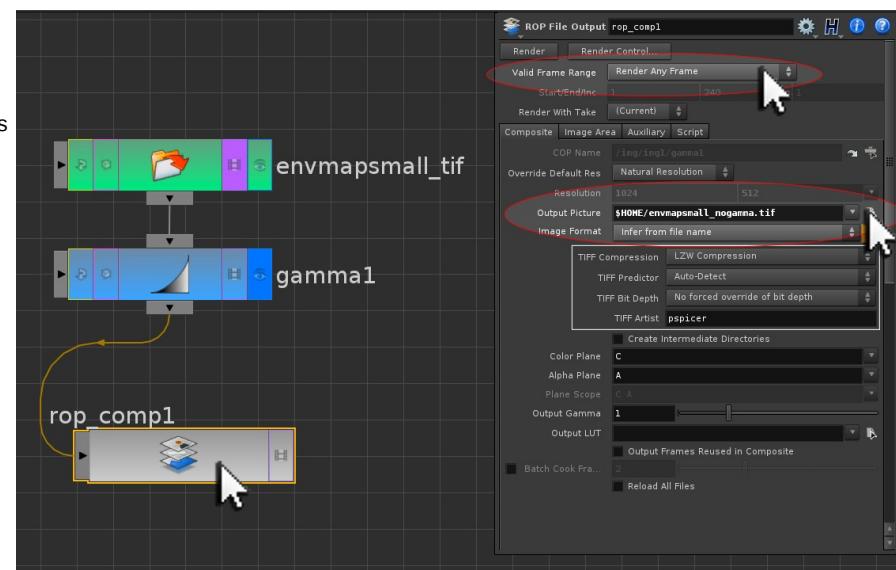
In the **parameters** for the **ROP File Output COP**, specify:

Valid Frame Range

Output Picture

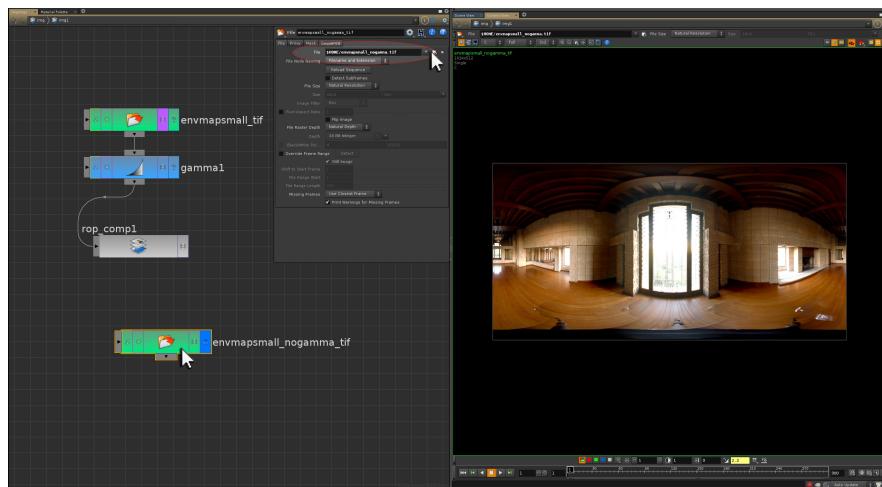
Render Any Frame

\$HOME/envmapssmall_nogamma.tif



The Render button at the top of the parameters can now be pressed and the image will be saved to disk.

A File COP can be used to read back in this image to check that its gamma has been removed successfully.



This image can now be successfully used as an Environment Light map for the simple scene.

IMPORTANT NOTE: This process of stripping off gamma can also be applied to background plates needing to be previewed in the Houdini viewer, and to textures needing to be assigned to geometry. Whenever Houdini has to process a colour image that contributes directly to the render, it is important to make sure it has been converted into a linear image with no embedded gamma.

Lights in Houdini can be created by using either the **Lights and Cameras Shelf Set**, or by utilising the **TAB menu system** (press **TAB** and type **light** with the mouse over either the **View Pane** or the **Network View**). The advantage of the Lights generated by the Shelf is that they are already preconfigured for different tasks. Also, their placement in the scene can be activated by the current view described by the View Pane.

DIFFERENT LIGHT TYPES

Point Light	An omni-directional light emulating for example candlelight
Spot Light	A theatrical cone light for highlighting specific objects
Area Light	A soft light utilised for describing light emitting from small areas (for example a skylight or window).
Distance Light	A light that emulates distant light sources such as the Sun
Environment Light	A light that emulates a special type of shadow called Ambient Occlusion or uses Images (including HDRI) for lighting a scene.
Ambient Light	For adding a uniform brightness to objects in the scene (can be better utilised for texturing tricks than direct lighting effects)

NOTE: all lights (aside from an Ambient Light) are the same Light Object reconfigured.

Sky Light	An Environment Light with an automatic sky reflection map
GI Light	Adds (or increases) the amount of bounce light in a scene.
Caustic Light	Adds caustic light effects to scene objects.
Portal Light	An Environment Light that only casts through doorways or windows
Geometry Light	Turns objects into lights

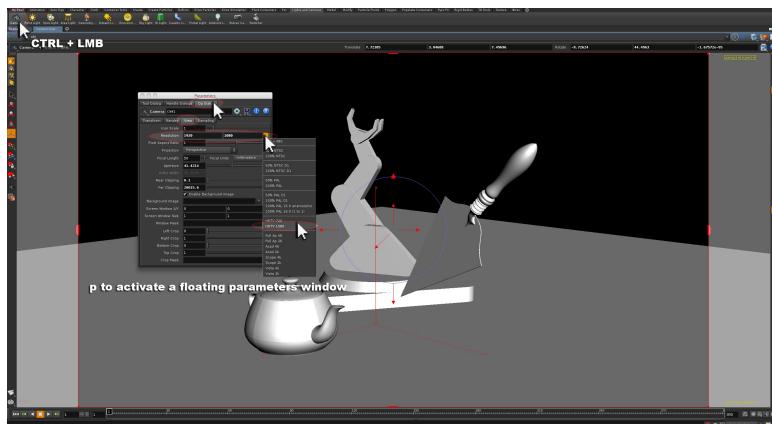
CREATING LIGHTS IN HOUDINI

Head back to **Object Level**, maximise the **Viewer**, and **activate the shelves**.



SIMPLE 3 POINT LIGHTING

Lighting always should be configured relative to a camera view. **Tumble the Viewer** to find an appropriate view of the scene, and from the **Lights and Camera Shelf** **CTRL + LMB** the **Camera button**. This will place a camera at the current view position.



In the **Viewer**, press **p** to activate a **floating Parameters Window**; and from the **Op Dialog > View** section of the parameters, set the **Resolution** parameter to **HDTV 1080**. This will format the camera for HD rendering.

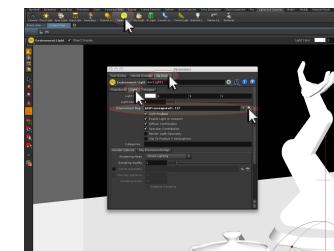


The position of the camera can be modified further by using the central camera controls (when in Camera Tool Mode), or by locking the camera to the Viewer using the **Camera Lock** button.

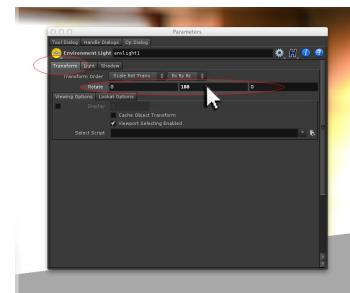
NOTE: The Camera Lock button can also be used on Lights as well as Cameras.

THE ENVIRONMENT LIGHT

From the Lights and Camera Shelf; activate an Environment Light (it will automatically place itself at the origin point of the scene). Activate the **parameters** for the **Environment Light**, and under the **Op Dialog > Light** section, load in **envmapsmall_nogamma.tif** image into the **Environment Map** parameter. This will wrap the environment map onto a giant sphere around the scene.



The **Sampling Quality** parameter of the Environment Light can also be increased to **10** to improve its appearance in the render (this will however increase render times).

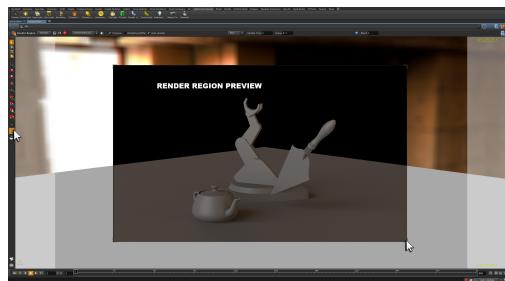


The **Environment Light** cannot be translated however it **can be rotated**. This allows for the end user to orient the environment map correctly to the direction of scene. The **Intensity** parameter of the Environment Light can also be increased or decreased depending upon its desired influence in the final render.

In the **parameters** for the **Environment Light**, specify:

Transform >
 Rotate 0 180 0
 Light >
 Intensity 1.5
 Render Options >
 Sampling Quality 4

A **Render Region Preview** shows the effect on the Environment Light on the scene.



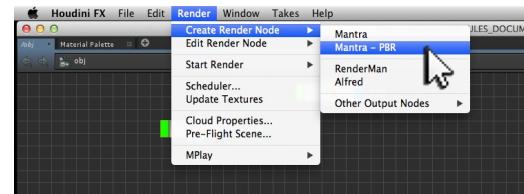
Sometimes a scene can be lit using an Environment Light alone; on other occasions the scene may also need to be augmented with other direct lights. For this example a simple 3 point lighting setup can be added to the simple scene, enhancing the effect of the Environment Light.

CREATING A MANTRA PBR RENDERING NODE

Currently the Render Region Preview of the scene is being done using default render settings.

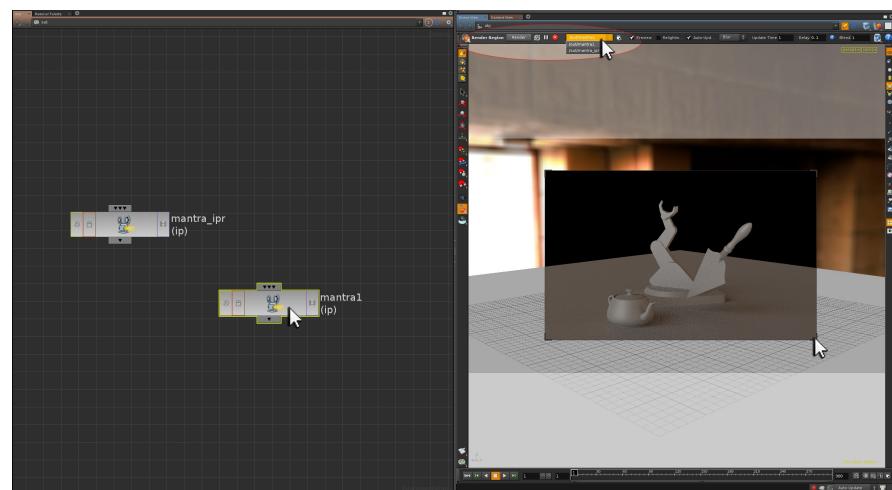
Before progressing with the lighting setup further it is a good idea to configure a Mantra node (Houdini's own renderer) for Physical Based Rendering (PBR).

From the main **Render** menu choose **Create Render Node > Mantra – PBR**.



This will create a **Mantra ROP** (Render Operator) at the **Outputs Level** of Houdini.

Set the **Viewer** to a **Scene View**, and head to the **Outputs Level of Houdini**. There will be two Mantra ROPs; the **mantra_ipr node** (the Interactive Preview node automatically created when Render Region Previews are created), and the **mantra1 node** (this is the newly created Mantra PBR node).



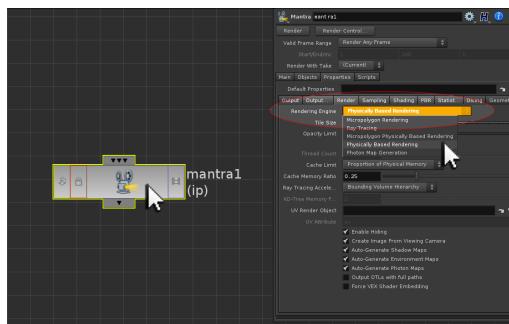
With the Render Region Preview activated over the Viewer, ensure the render node it is using is set to **/out/mantra1** and **re-draw** the **preview window** to re-render the scene. The **mantra_ipr** node can also be deleted as it is no longer required.

PBR RENDERS

The Render Region Preview of the scene looks slightly different when the scene is rendered using parameter, and switch between **Physically Based Rendering** and the default **Micropolygon** the mantra1 (PBR) node.



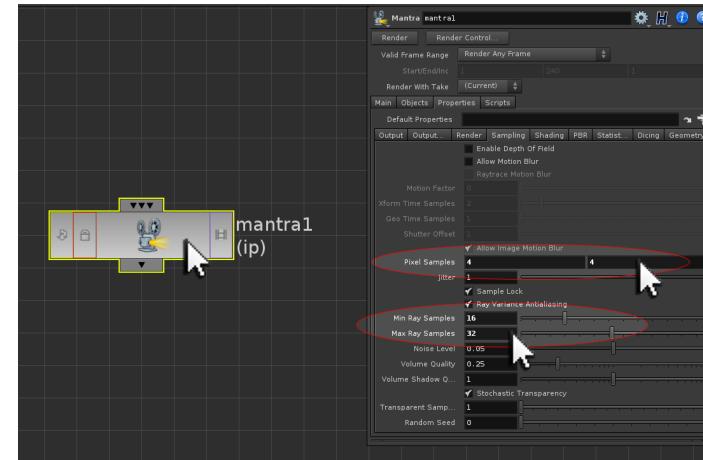
Firstly, the render is slightly brighter than before. This is due to the PBR (Physically Based Render) aspect of the render calculation. Bounce light between the scene objects is now being automatically generated. Secondly, the render has white speckles over the surfaces. This is due to the standard render settings currently applied to the mantra1 node.



In the parameters for the mantra1 node, locate the **Properties > Render > Rendering Engine** engines. This will cause the Render Region Preview to update accordingly. Take a moment to note the difference between the two render aesthetics.

With the **Rendering Engine** parameter set back to **Physically Based Rendering**, head to the **Properties > Sampling** section of the parameters. This parameter section controls the quality of the rendered image. In this section of the parameters specify:

Pixel Samples	4	4
Min Ray Samples	16	
Max Ray Samples	32	



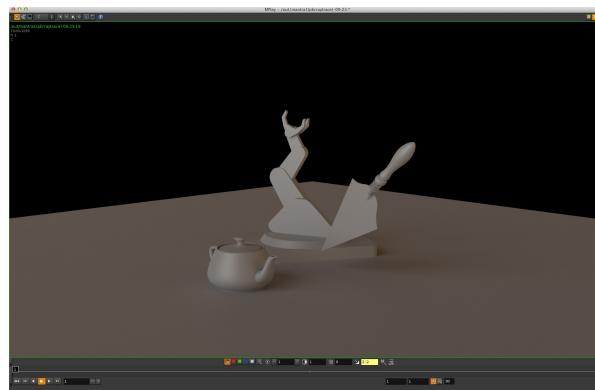
This will increase the render quality of the resulting image. It is worth noting however that when sampling values are increased to obtain better rendering results, render times also increase. It is

always therefore a balance finding the most appropriate render quality settings relative to also obtaining fast efficient optimum rendering times.

A formal render can be generated of the scene, by RMB on the Film Spool button of the Viewer and choosing mantra1, or by LMB on the Render button found at the top of the mantra1 node parameters pane.



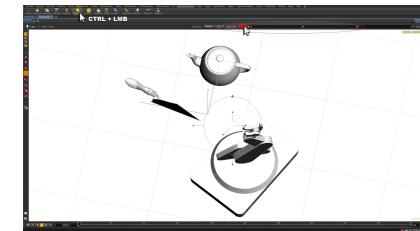
In either case, a full render will appear in MPlay (Houdini's image viewer) using the camera image resolution specified on the camera object.



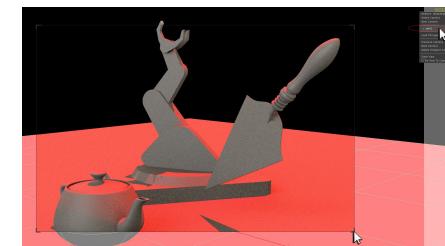
This formal render can be called whenever detailed image checking is required. For the positioning and configuration of lights however, the Render Region Preview offers a convenient alternative.

THE RIM LIGHT

Head back to **Object Level**, **maximise the Viewer**, and **activate the Shelves**. Interactively position the **Viewer**, so that it looks at the back of on the objects from a high vantage point. Using **CTRL + LMB** activate a **Distant Light** using the Shelf. This will position a Distant Light at the View Pane's current position.



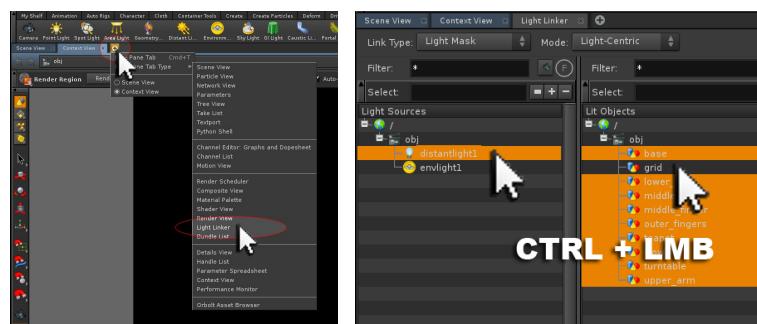
In the **Viewer parameters** for the **Distant Light**, set the **colour** of the light to **red**. This will make seeing its contribution to the scene easier to determine. **Reset the Viewer** back to the **camera view** using the **Camera Menu**, and redraw a **Render Region Preview**.



NOTE: if a Distant Light is created from the current viewing position, and the viewer then tumbled away; the Viewer will return an orthographic view of the scene rather than a perspective view. This can be rectified by pressing SPACEBAR + o with the mouse over the Viewer.

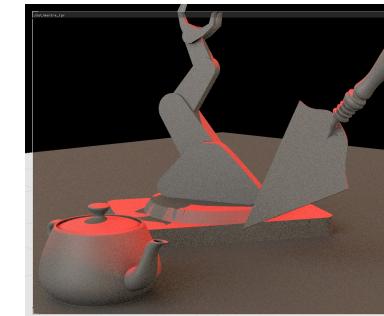
ASSIGNING LIGHTS TO SPECIFIC OBJECTS

Currently the **Rim Light** is affecting every object in the scene. As its function is one of highlighting the edges of objects, it is possible to assign it only to the primary objects in the scene. This is a process known as **Light Linking**. Over the **Viewer panes**, activate a **Light Linker** as a new **Pane Tab Type**.



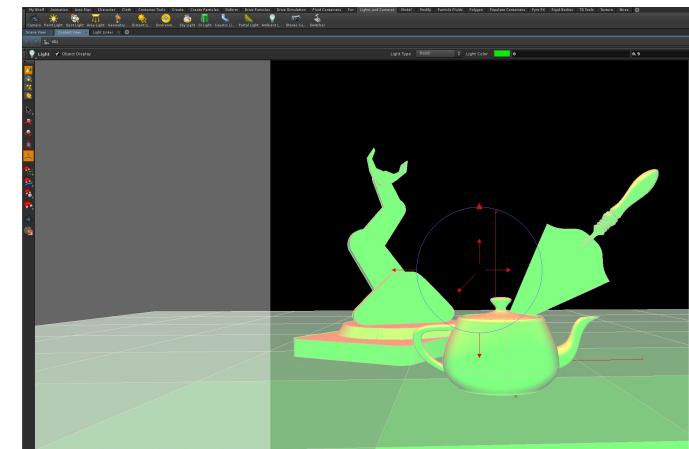
Adjust the **Lit Objects** list so that the **Rim Light** (`distancelight1`) affects everything in the scene **except the ground object**. This can be done by using **CTRL + LMB** to deselect the ground object. A Render Region Preview shows the results of the light linking. See file

`simple_lighting_stage1.hipnc`

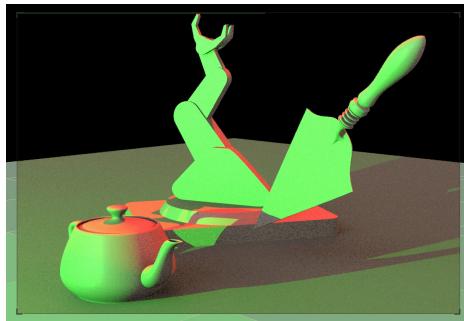


THE FILL LIGHT

Position the **Viewer** so that the scene faces all of the **unlit grey areas** of the scene geometry, and using the Lights and Cameras Shelf; **CTRL + LMB** activate a **Point Light**.

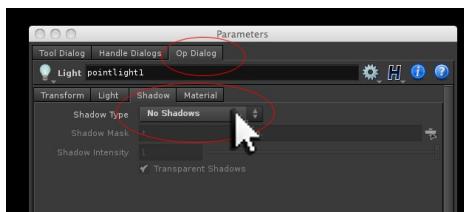


In the **Viewer parameters** for the Point Light, set the **Light Color** parameter to **green**. A **Render Region Preview** reveals the effect of the Fill Light on the scene.

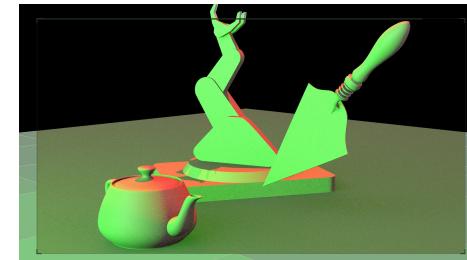


Currently, the Fill Light is casting large shadows over the scene, as well as contributing specular highlights to the geometry. Whilst the Light Linker can also be used to assign shadows to specific objects; the simplest way to remove the shadows and specular contribution of the Fill Light is to deactivate this behaviour from the Point Light parameters.

Press **ESC** to exit from the **Render Region Preview** mode, and press **ENTER** with the mouse over the Viewer to reactivate the **Point Light** as the **current tool**. Press **p** to activate the **floating parameters window**; and from the **Op Dialog > Light** section of the **parameters**, deactivate the **Specular Contribution** tick box.



From the **Op Dialog > Shadow** section of the **parameters**; set the **Shadow Type** parameter to **No Shadows**. A Render Region Preview shows the effects of these modifications on the scene.



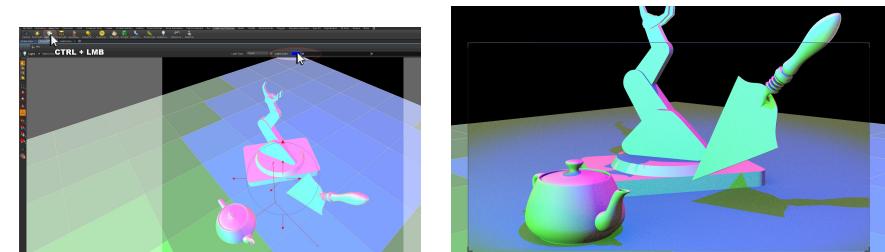
Shadows can also be deactivated on the Distant Light if required. [See file](#)

simple_lighting_stage2.hipnc

NOTE: When you have finished with a Render Region Preview; close the preview window and press **ESC** and **ENTER** with the mouse over the Viewer to activate the last created/selected node.

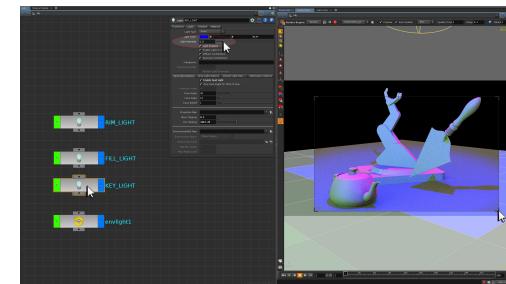
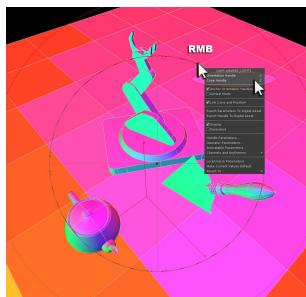
THE KEY LIGHT

Position the Viewer so that the scene faces a high vantage point on the opposite side of the camera to the Fill Light. **CTRL + LMB** activate a **Spotlight** from the Lights and Camera Shelf, and set its colour to blue.



Set the Viewer to the **camera view**, and activate a **Render Region Preview** to see the effect of the key light. The primary colours of the lights can be used to help determine if their position is correct relative to what is seen in the render.

When viewing the scene directly through a selected spotlight, interactive position adjustment handles will appear in the Viewer when in the Light Tool mode (similar to the Camera adjustment controls). The position adjustment controls can be changed into controls for modifying the shape of the spotlight's Cone Angle by **RMB** on the **positioning handles** and choosing **Cone Handle** from the resulting menu.



It is also a good idea to rename the lights to better match their function in the scene. Generally the Rim Light and the Key Light will have the highest intensities, with the Fill Light and Environment Light having a much lower intensity by comparison.

When the intensities of the lights have been approximately set; the colours of all the lights can be adjusted so that they more closely resemble real world lighting. As a general guide, the Rim Light should be set to white; the Key Light set to a warm near white colour; and the Fill Light set to a cold colour

Similarly **RMB** on the **Cone Handle controls** and choosing **Orientation Handle** will reset the controls back to the interactive position adjustment handles.

MODIFYING THE INTENSITY

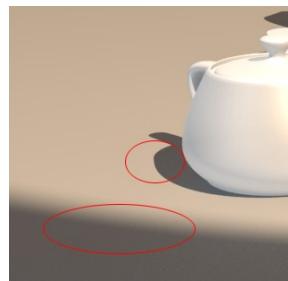
With the lights correctly positioned, the **Light Intensity** can be adjusted so that the lights have a better visual relationship with each other. Currently all lights have the same intensity. In the **Network Editor**, go through each light in turn and adjust their intensities accordingly.



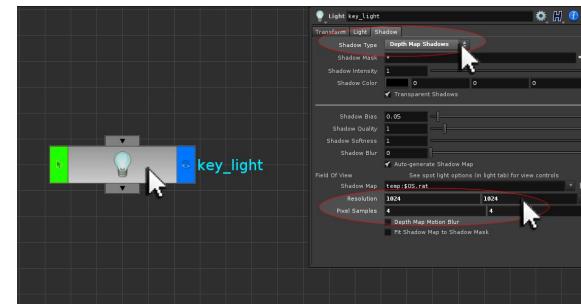
When a formal render of the scene is activated through MPlay, a Detail View can also be activated for checking exact pixel values of the resulting image. Pressing m with the mouse over MPlay will activate the Details View. This is a floating window that will magnify areas of the image, and it uses the position of the mouse to determine which area of the image to magnify. [See file simple_lighting_stage3.hipnc](#)

DEPTH MAP SHADOWS

When the render is examined, a visual difference is occurring between the edge of the spotlight and the shadows being cast. The edge of the spotlight region is soft, whilst the shadows being cast by the spotlight are sharp. By default, shadows from lights are created using Ray Tracing. This results in sharp shadows.



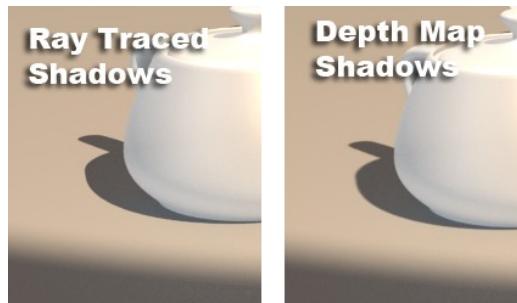
An alternative to Ray Traced Shadows is Depth Map Shadows. This is where shadow information is rendered out to disk as an image, and then called at render time to generate the shadows. One advantage of Depth Map Shadows is that they can be softened.



In the **parameters** for the **Key Light**, specify under the **Shadow** section:

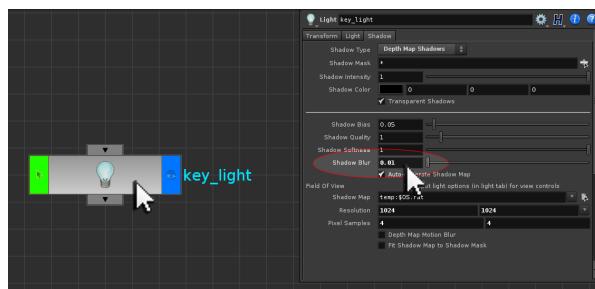
Shadow Type	Depth Map Shadows		
Resolution	1024	1024	
Pixel Samples	4	4	

This will generate an image of the shadow information. Increasing both its resolution and Pixel Samples will help ensure good depth map shadow quality. Currently this depth map shadow image is being rendered out to disk automatically just before the main image rendering takes place. A future lesson will demonstrate how to render out depth map shadows ahead of render time, bypassing this automatic shadow map generation.



When a formal render of the depth map shadows is activated, the visual appearance is initially very similar with only a slight softening of the edge of the depth map shadow.

With Depth Map Shadows configured, the Shadow Blur parameter can be adjusted to further soften the edges of the shadow.



In the **parameters** for the **key_light**, specify under the **Shadow** section:

Shadow Blur **0.01**

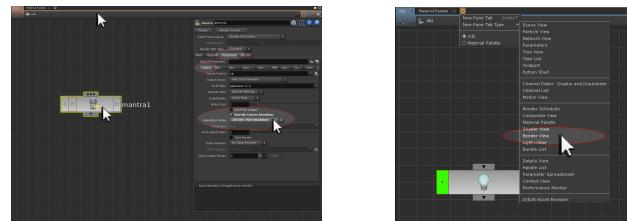
When a formal render of the scene is activated, the effect of the shadow blurring can be seen. See file **simple_lighting_stage4.hipnc**



IMPORTANT NOTE: This blur value is very small as a result of the unit values used by Houdini. By default a Unit Value of 1 is the default size of a Box SOP. A blur value of 1 would therefore result in a very large blur relative to the scene size, causing the shadows to blur to the point of disappearing from the render.

OTHER LIGHTING UTILITIES

Sometimes it can be useful to have a Render View, as well as a Scene View to help position and modify light colours and intensities to aid the refinement of the render. **Switch** to the **Outputs Level of Houdini** and locate the **mantra1** ROP node just created. In the **Properties > Output** section of its **parameters**, activate the **Override Camera Resolution** tick box, setting a **Resolution Scale** to **1/3 (One Third Resolution)**.



Diffuse contribution

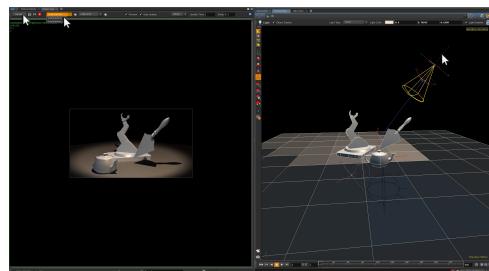
This specifies whether or not a light affects the diffuse (matte) part of an object's surface

Specular Contribution

This specifies whether or not a light affects the specular

Back at **Object Level**, activate a **Render View** as a new **Pane Tab Type** over the **Network Editor**.

In the **Render View**; ensure the driver to render with is set to **/out/mantra1** and hit the **Render Button**. This will generate a **1/3 resolution** render of the scene, that will update as the lighting is further adjusted in the **Viewer**.



The **parameters** for a **selected light** can also be activated in the **Viewer** by pressing **p** with the mouse over the **Viewer** to activate the **floating parameters window**.

TURNING LIGHTS ON AND OFF

Every light also has additional functionality relating to how it interacts and contributes to the aesthetic of the scene. This functionality can be found in the **parameters** for the light.

Light Enabled

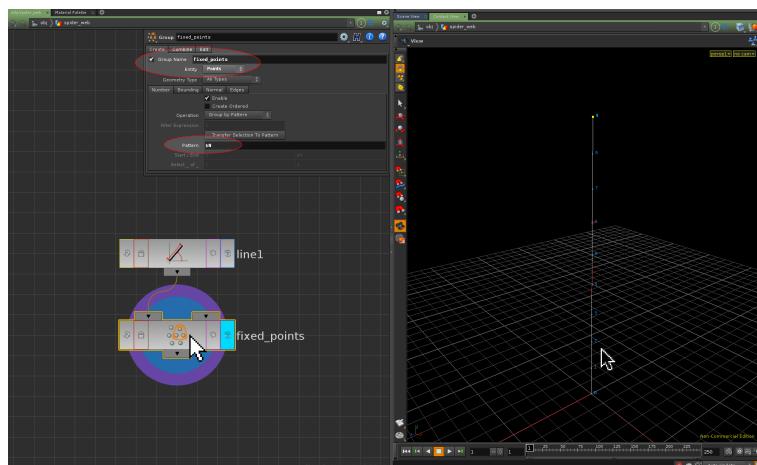
This turns the light on or off

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:

Distance	3
Points	10

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

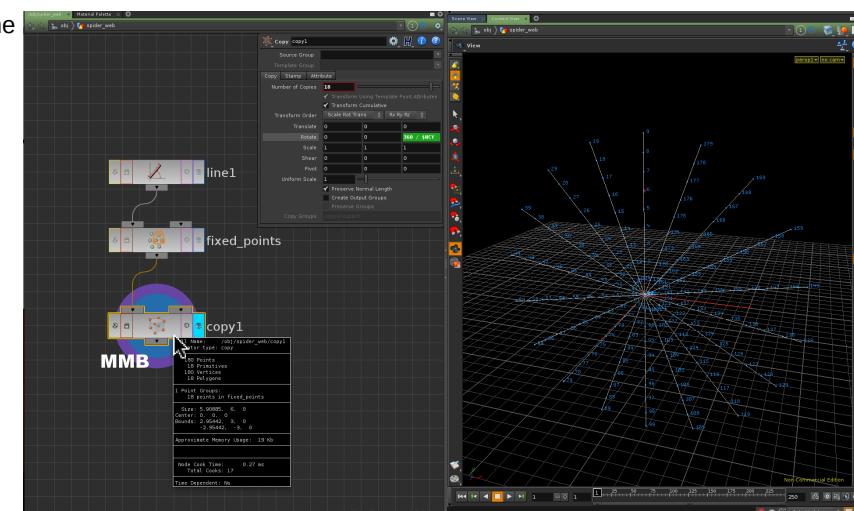


In the **Pattern** parameter enter the expression **\$N**. This will procedurally group the very last point of the line.

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

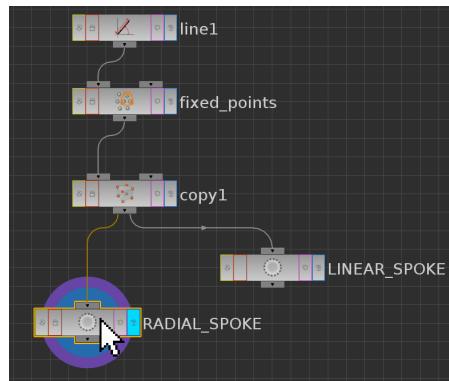
Number of Copies	18
Rotate	0
0	0
360 / \$NCY	

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_spokes**.



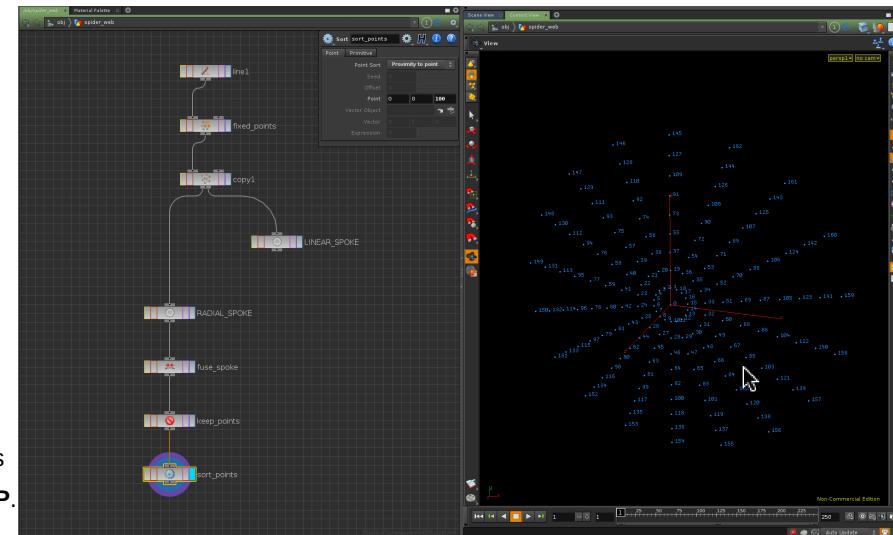
Creating the Radial Spoke To the **radial_spoke** Null SOP append a **Fuse SOP**. This will fuse together the centre points of the web. To the **Fuse SOP** append a **Delete SOP**. In its **parameters** specify:

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

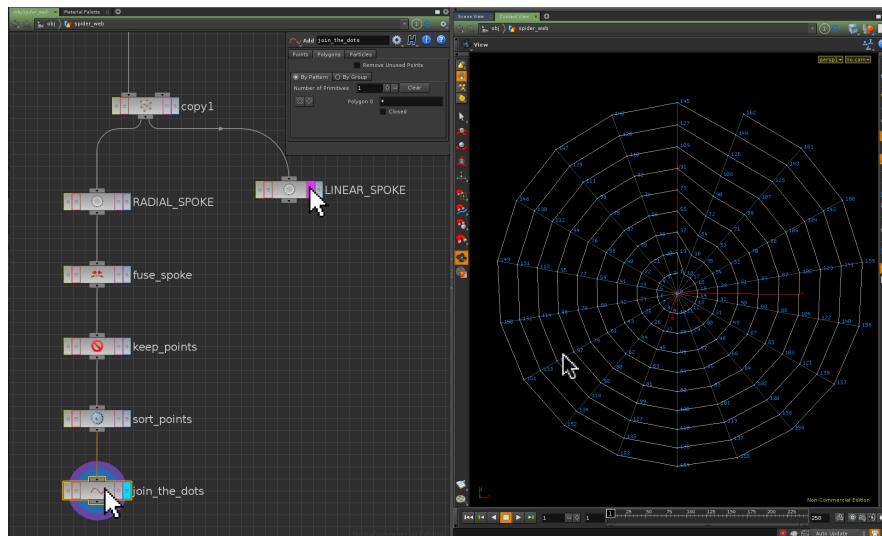


This will perform a radial sort of the point numbers from the centre outwards forming a spiral.

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.



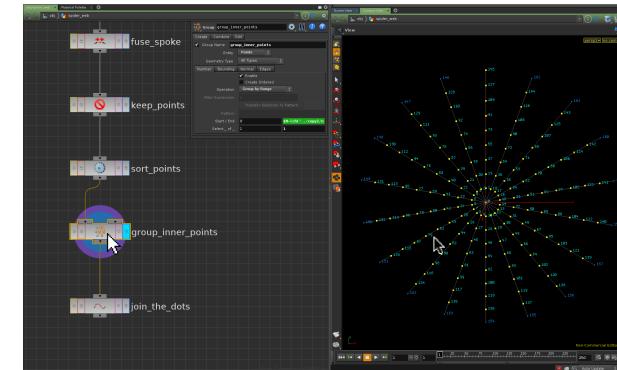
CONTROLLING THE DRAWN POINTS

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.

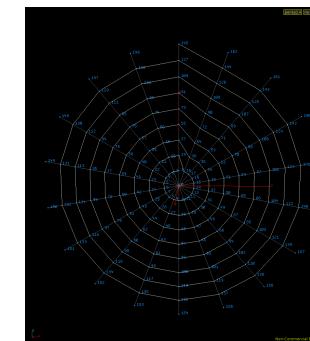
An expression can be created that will automatically group the entire inner web points, no matter how many spokes are assigned to the web. Between the **Sort SOP** and the

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



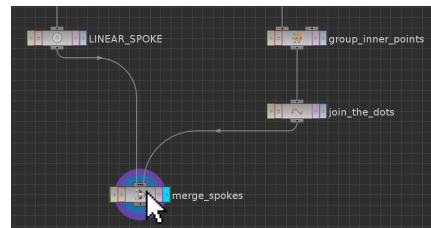
This will procedurally group all of the points that will make up the radial spoke. The **\$N - (ch("../copy1/ncy")-1)** expression returns the total number of points minus the total number of copies specified in the Copy SOP. Visually this equates to all of the points being grouped apart from all but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

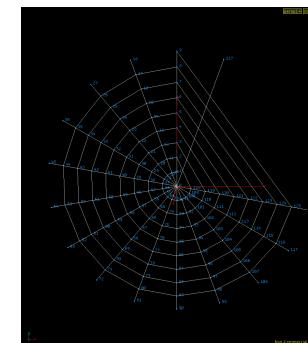
CREATING AN ANIMATED WEB

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.



DYNAMICALLY ANIMATING THE WEB

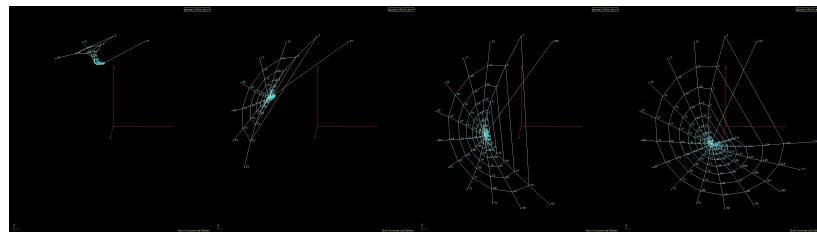
To the **Delete SOP** append a **Spring SOP**. This node can be utilised to add simple dynamics to the web. Under the **Forces** section of its Parameters, specify:

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 to the Spring SOP a **Point SOP**. In the **Particle** section of its **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.

See file **spider_web_end.hipnc**