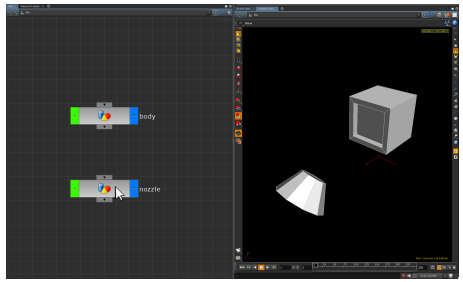
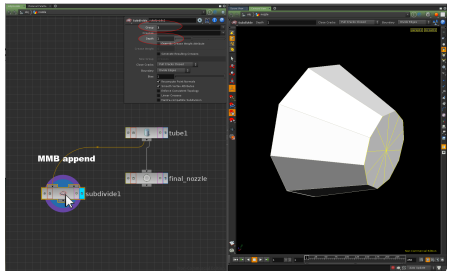
SIMPLE HOSES

The **Spring SOP** is a **low-level tool** for creating **simple cloth**, **hair**, and **wire effects**.



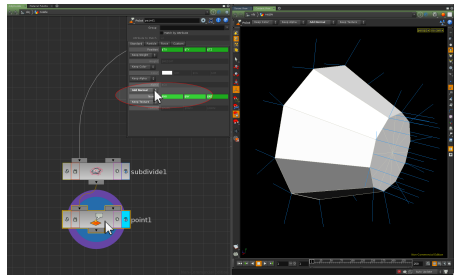
Open the scene hose_attach_begin.hipnc. This scene contains two animated objects that can be joined together with a simple dynamic hose. **Go inside** the **nozzle object**, and **MMB** append to the **Tube SOP** a **Subdivide SOP**. In the **parameters** for the **Subdivide SOP** specify:

Group	**1**
Depth	**1**



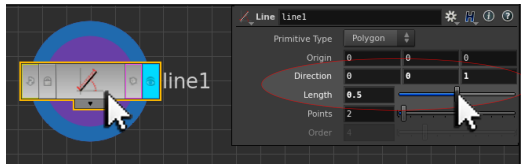
This will subdivide only primitive face number 1, creating a centre point for the attachment of a hose.

To the Subdivide SOP append a **Point SOP**. In its **parameters** set **Keep Normals** to **Add Normals**. This will orient any copied geometry onto its surface. **Point Normals Display** can be temporarily activated in the Viewer to see the effect of this operation.



In a new part of the **Network Editor**, create a **Line SOP**, and in its **parameters** specify:

Direction	**0**	**0**	**1**
Length	**0.5**		

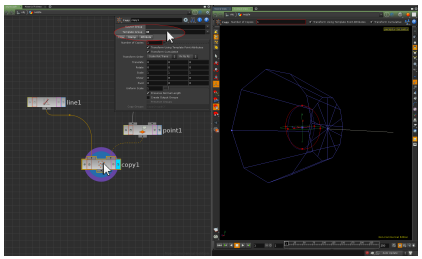


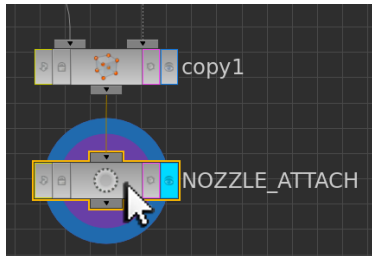
Orientating the line in the **Z-Axis** will ensure it will **copy correctly** onto the **nozzle geometry**.

Append a **Copy SOP** to the **Line SOP**, wiring the **Point SOP** as its second input.
In the **parameters** for the **Copy SOP**, specify:

Template Group	**40**

This will copy the line onto point 40; the centre point created by the Subdivide SOP.

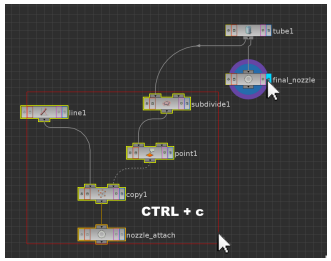


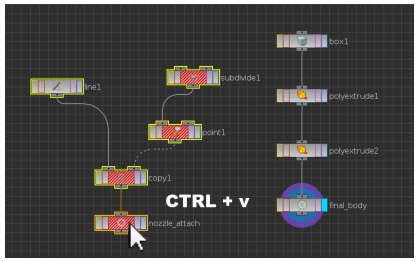


As a final step, append a **Null SOP** to the **Copy SOP**; renaming it to **NOZZLE_ATTACH**.

See file hose_attach_stage1.hipnc

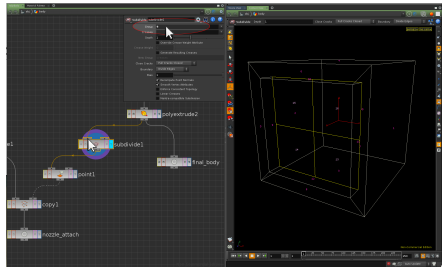
Reset the **Display and Render Flag** back to the **final_nozzle Null SOP**. **Select all** of the **added nodes**, and **Copy** (**CTRL + c**) them into memory.





Head into the **Geometry Level** for the **body object**, and **Paste** (**CTRL + v**) the pasted nodes into its network. Wire the output of the **second PolyExtrude SOP** as the input to the **Subdivide SOP**. Modify the **parameters** of the **Subdivide SOP** to:

Group **9**

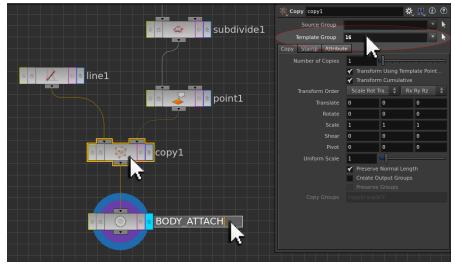


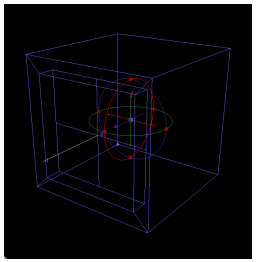
This will subdivide primitive face number 9 to create a centre point on the body geometry for attaching the other end of the hose.

In the **parameters** of the **Copy SOP**, modify the **Template Group parameter** to:

Template Group **16**

This will copy the line geometry only onto point 16 of the body geometry. **Rename** the **Null SOP** at the end of the **Copy SOP** to **BODY_ATTACH**.



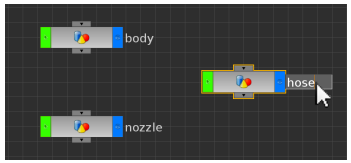


As a final step, **reset** the **Display and Render Flag** back to the **final_body Null SOP**.

See file hose_attach_stage2.hipnc

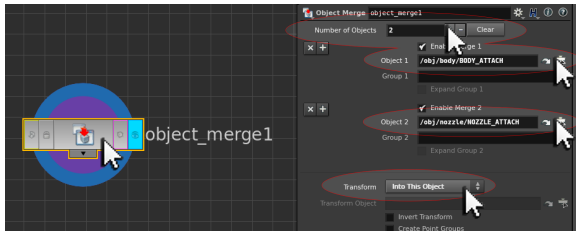
CREATING THE HOSE BASE CURVE

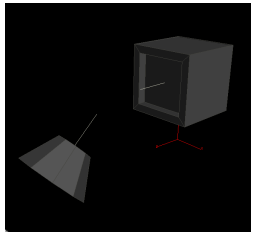
At **Object Level**, create a **new** piece of **Geometry** and **rename** it to **hose**.



Go **inside** the **hose object** and create an **Object Merge SOP**. This will read in both copied lines as a single piece of geometry. In the **parameters** for the **Object Merge SOP** specify:

Number of Objects	**2**
Object 1	**/obj/body/body_attach**
Object 2	**/obj/nozzle/nozzle_attach**
Transform	**Into this Object**





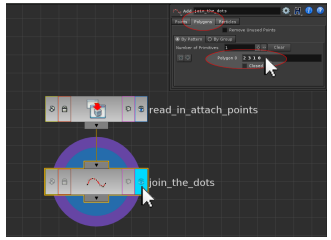
Append an **Add SOP** to the Object Merge SOP. In its **parameters** specify:

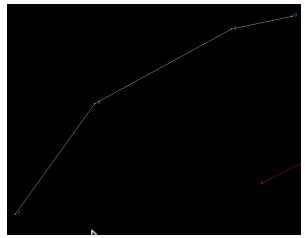
Points >

☑ **Delete Geometry But Keep the Points**

Polygons >

Polygon 0 **2 3 1 0**



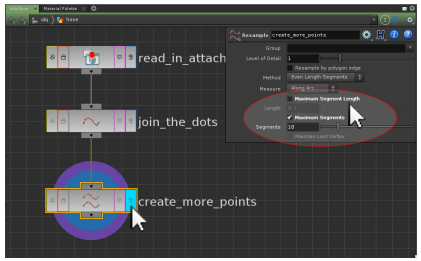


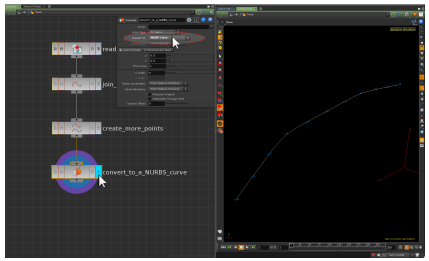
This will remove the existing geometry and draw a new line through the specified points.

Append a **Resample SOP** to the **Add SOP**. In its **parameters** specify:

☐	**Maximum Segment Length**
☑	**Maximum Segments**
Segments	**10**

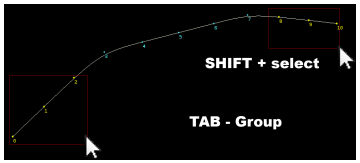
This will create a fixed amount of points on the animated line.





Append a **Convert SOP** to the Resample SOP. Set the **Convert to** parameter to **NURBS Curve**.

In the **Viewer**, **select** the **first three points** of the line and **SHIFT + select** the **last three points**. Press **TAB** and type **Group**. This will create some fixed points that won't respond to any dynamics assigned to the line.



Rename the **Group SOP node** in the **Network Editor** to **declare_fixed_points**.

NOTE: This node renaming will only work if `**opname(".")**` has been assigned as the Group SOP > **Group Name parameter**.

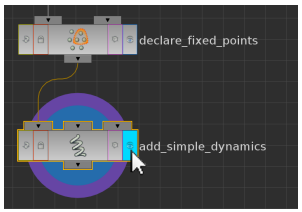
Append a **Spring SOP** to the **Group Geometry SOP**. This node will add simple dynamics to the line. In the **parameters** for the **Spring SOP** specify:

Forces >

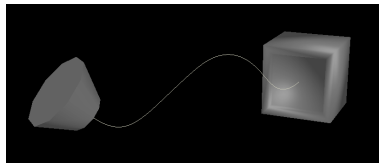
External Force	**0**	**-1**	**0**
Turbulence	**3**	**3**	**3**

Nodes >

Fixed Points	**declare_fixed_points**
Mass	**0.1**



Activating the Fixed Points parameter will ensure the first three and last three points of the curve do not respond to the Spring SOP dynamics.



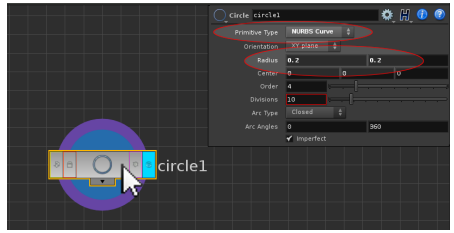
When **PLAY** is pressed, the effect of the Spring SOP can be seen.

See file hose_attach_stage3.hipnc

CREATING THE HOSE GEOMETRY

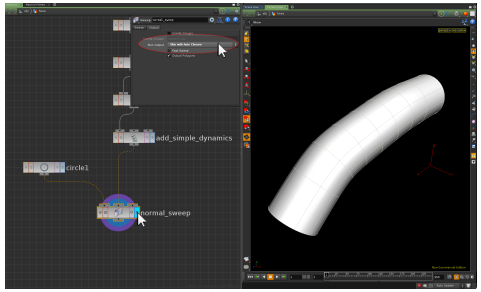
As a new network chain, create a **Circle SOP**. In its **parameters** specify:

Primitive Type	**NURBS Curve**	
Radius	**0.2**	**0.2**



Append a **Sweep SOP**, wiring the output of the **Spring SOP** as its **second input**. In the **parameters** for the **Sweep SOP** specify:

Skin Output	**Skin with Auto-Closure**

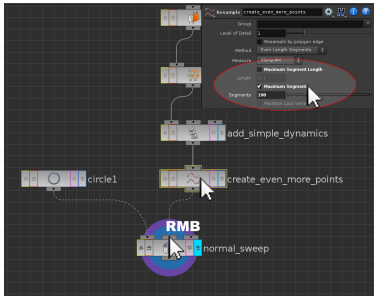


This will create a tube from the swept circles along the springy line.

REFINING THE HOSE SHAPE

RMB on the **second input** of the **Sweep SOP** to **insert** another **Resample SOP** into the springy line network. In its **parameters** specify:

☐	**Maximum Segment Length**
☑	**Maximum Segments**
Segments	**100**

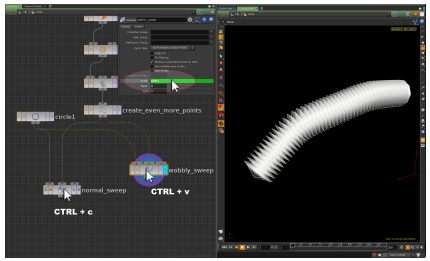


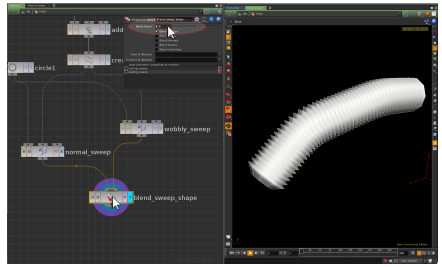
This will create much more detail to the swept tube, allowing for its shape to be refined.

Copy (**CTRL + c**) the **Sweep SOP**, and **Paste** (**CTRL + v**) a copy of it into the network. In the **parameters** for the **second Sweep SOP** specify:

Scale	**$PT%2**

This simple expression will **scale** every **alternate** swept **circle** to **0**, creating a **concertina** effect to the resulting tube geometry. While this shape is perhaps a little too extreme for a connecting hose, it can be **morphed** with the original swept tube to create a more naturalized aesthetic.



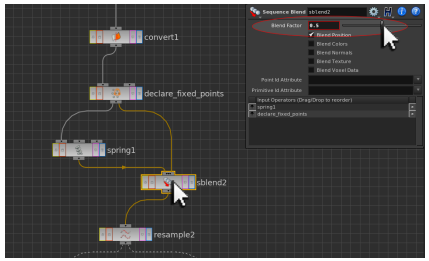


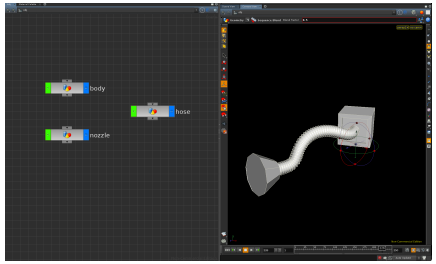
Wire both Sweep SOPs into a **Sequence Blend SOP**. In the **parameters** for the **Sequence Blend SOP** specify:

Blend Factor	**0.5**

This will morph or blend the two tube shapes together to form a more natural hose shape.

An **additional Sequence Blend SOP** can also be **inserted** after the Spring SOP to **blend between** the **springy curve** and the **non-springy curve** to control the stiffness of the connecting hose as it animates.



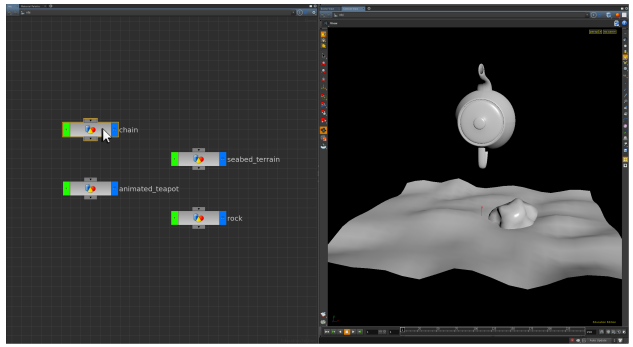


Back at **Object Level**, the effect of the hose can be seen when **PLAY** is pressed. As the body and nozzle animate, the hose automatically stretches between these two objects.

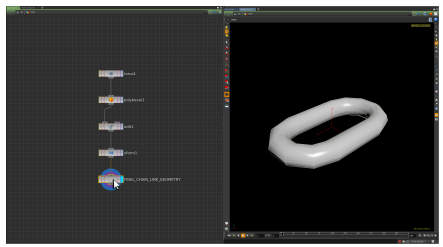
See file hose_attach_complete.hipnc

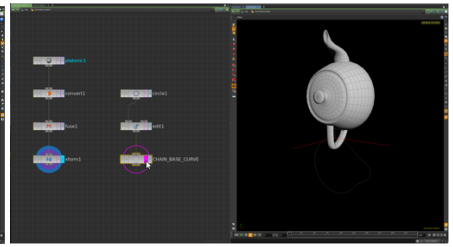
CHAIN EFFECTS

Houdini's Dynamic Engine DOPs can be used to create chain effects that interact with other objects. **Open the scene chain_teapot_begin.hipnc**.



This scene contains a **polygon teapot** animated over a **seabead_terrain** and **rock** objects. **Examination** of the **chain object** reveals a **geometry network** creating **a single chain link**.



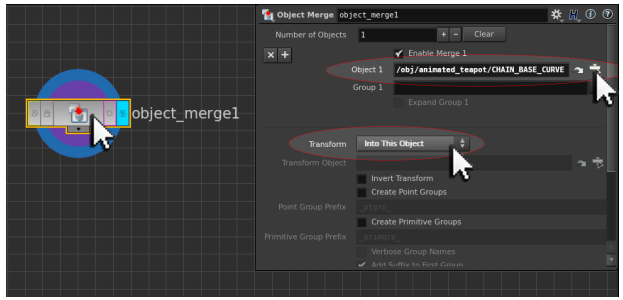


The animated **teapot** also has a **second network** inside it creating a **base curve for the chain effect** that is aligned through the teapot handle. Placement of this chain base curve inside the animated object ensures that no matter where the teapot is moved in the scene, the base curve for the chain will always be correctly positioned. The chain base curve is a **NURBS curve**, and has also been **edited** to create **a starting shape** for the chain.

CREATING THE CHAIN

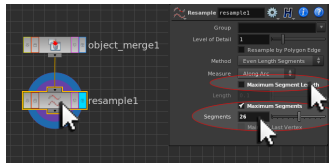
Inside the **chain object**, create an **Object Merge SOP**. In its **parameters** specify:

Object 1	**/obj/animated_teapot/CHAIN_BASE_CURVE**
Transform	**Into this Object**



This will read in the chain base curve, so that the final chain geometry can be built from it.

Append to the Object Merge SOP a **Resample SOP**.



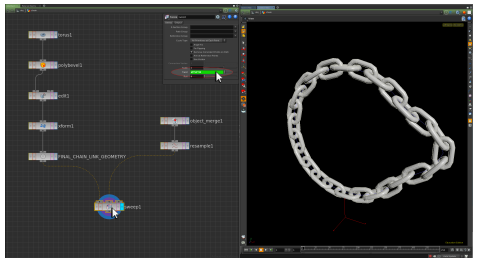
In its **parameters** specify:

☐	**Maximum Segment Length**
☑	**Maximum Segments**
Segments	**26**

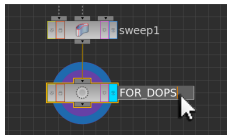
This will **convert** the NURBS curve **to a polygon curve**, and give enough points along it to create interlinking chain geometry.

Create a **Sweep SOP**, and wire the **output** from the **FINAL_CHAIN_LINK_GEOMETRY Null SOP** as the **first input**, and the **Resample SOP** as its second input. In the parameters for the Sweep SOP specify:

Twist	**$PT%2 * 90**



This expression will take each point of the base curve in turn (**$PT**) and assign alternating values of **0** or **1** to them (%2). Multiply this value by 90 will rotate each alternating chain link by 90 degrees.

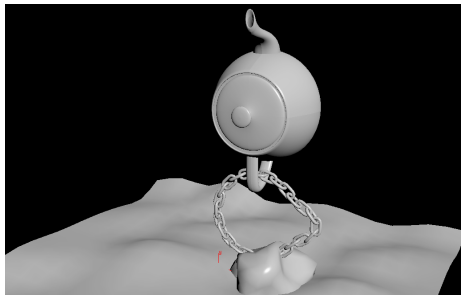


As a final step, append a **NULL SOP** to the Sweep SOP, and rename it to FOR_DOPS. This will help differentiate the chain construction network from the automatic nodes that the dynamics setup will create.

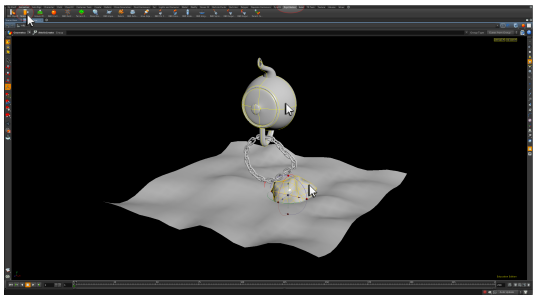
See file chain_teapot_stage1.hipnc

ADDING THE DYNAMICS

Currently when **PLAY** is pressed, the **chain geometry moves** relative to the animated teapot; but it is **static** and **intersects** with the other scene objects as it moves.

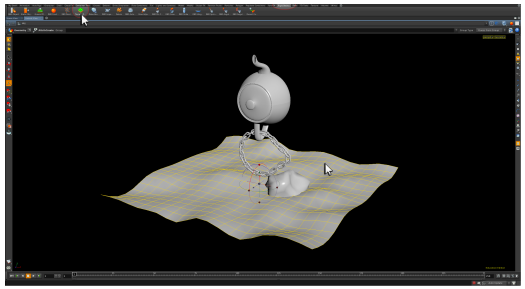


Dynamics (DOPs) can be **activated** for the **scene objects** to rectify this, so that a **naturalistic chain movement** is **created** allowing it to flow over the other objects.



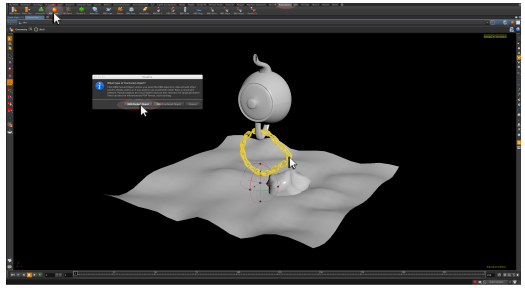
At **Object Level**, rewind the scene and **maximize** the **Viewer**. **Select both** the **teapot** and the **rock** objects, and from the **Rigid Bodies Shelf** activate the **Static Object** button. This will convert these objects into static dynamic objects that the chain can interact with.

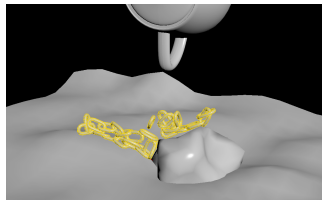
Next, select the **seabed_terrain object**, and from the **Rigid Bodies Shelf** activate the **Terrain Object** button.



NOTE: As the seabed is a grid, the Terrain Object button must be activated rather than the Static Object button. Static Objects have to be geometry with thickness (such as the teapot or the rock) to work correctly. Flat objects with no thickness (such as the seabed) must be assigned as Terrain Objects to work correctly (at which point DOPs will internally extrude flat geometry to create a collision volume).

Finally, **select** the **chain object**, and from the **Rigid Bodies Shelf**, activate the **RBD Fractured Object** button. When the **prompt** appears asking for a **Type of Fractured Object**, **select** the default **RBD Packed Object** option. A **RBD Fractured Object** will consider each internal chain link of the chain as a unique dynamic object in its own right, allowing each chain link to properly interact with each other.





When **PLAY** is pressed, the chain responds as a dynamic object; however currently each chain link falls through the teapot handle and each other onto the seabed.

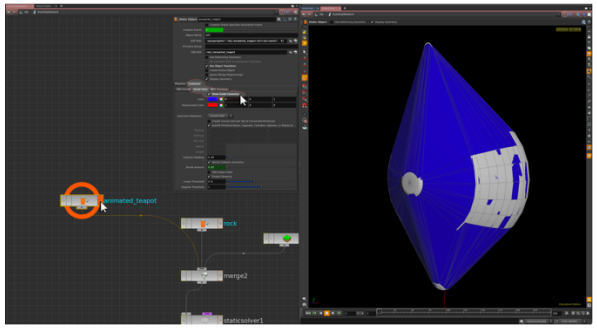
See file teapot_chain_stage2.hipnc

CONFIGURING THE DYNAMICS

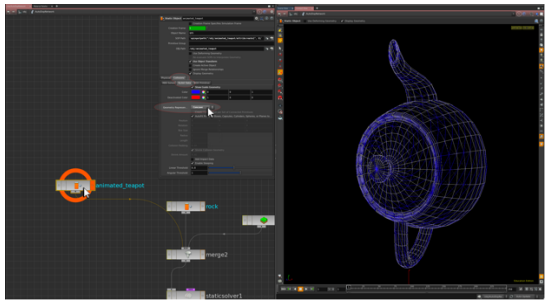
Go inside the **AutoDopNetwork** to access the dynamics setup. Each dynamic object node will need some minor adjustments to get the simulation working as expected.

Activate the **orange Display Flag** for the **animated_teapot Static Object DOP**, and in the **Collisions > Bullet Data** section of the **parameters** activate the **Show Guide Geometry tick box**.

By **default**, the **collision volume** for the teapot only gives a **loose approximation** of the teapot shape.

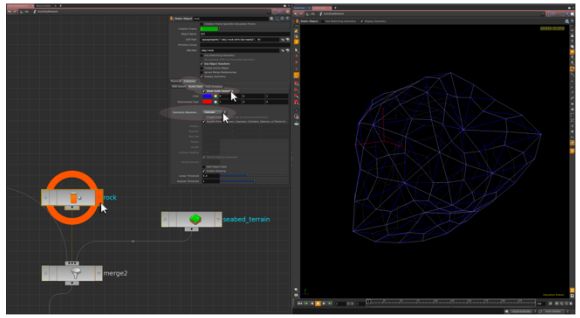


In the **Collisions > Bullet Data** section of the **parameters** set the **Geometry Representation** parameter to **Concave**. A **wireframe examination** of the dynamic teapot reveals a far more refined collision mesh has been generated.

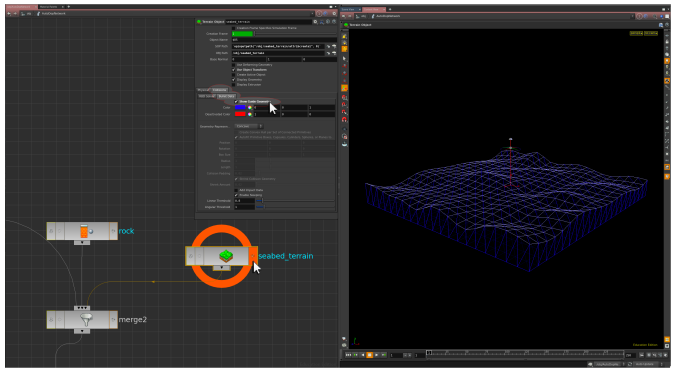


NOTE: All **Bullet Data parameters** respond to the **Bullet Solver** (the **default RBD Engine** of **DOPS**).

Activate the **Display Flag** for the **rock Static Object DOP**, and in the **Collisions > Bullet Data** section of the **parameters** activate the **Show Guide Geometry tick box**. As before, set the **Geometry Representation parameter** to **Concave** to get a more accurate collision mesh to the rock shape.

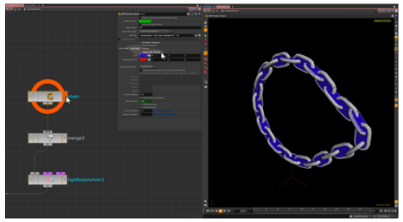


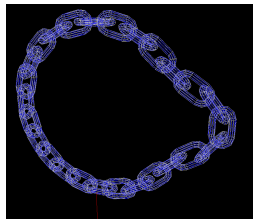
Activate the **Display Flag** for the **seabead_terrain Static Object DOP**, and in the **Collisions > Bullet Data** section of the **parameters** activate the **Show Guide Geometry tick box**.



This will display the **extrusion automatically generated** by the **Terrain Object button**. The **Bullet Data parameters** also reveal that a **Concave Geometry Representation** has been already configured.

Activate the **Display Flag** for the **chain RBD Fractured Object DOP**, and in the **Collisions > Bullet Data** section of the **parameters** activate the **Show Guide Geometry tick box**.





A Shaded View of the collision geometry reveals why the chain links do not yet interact properly. Each chain link is a solid collision mesh rather than a hollow loop. As before activating a **Concave Geometry Representation** can fix this.

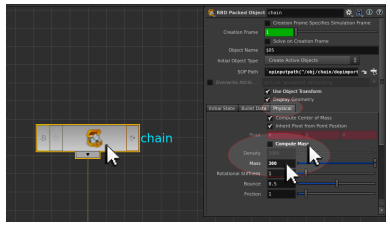
Deactivate the display of all of the **Guide Geometry**, and **reset** the **orange Display Flag** back to the **Output DOP**. When **PLAY** is pressed, the chain links respond correctly to dynamics; however, do not accurately interact.

FINAL FIXES

In the **parameters** of the **Gravity Force DOP** specify:

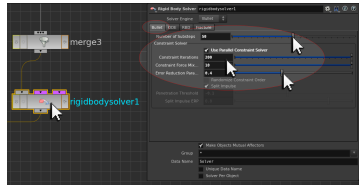
Force	**0**	**-1.5**	**0**

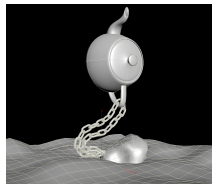
This will create a floating aesthetic to the chain as if underwater. In the **Physical parameters** for the **Chain RBD Packed Object DOP**, **deactivate** the **Compute Mass option**, and specify a **Mass** parameter value of **300**. This will make the chain heavier in the simulation.



In the **Bullet parameters** for the **Rigid Body Solver DOP** specify:

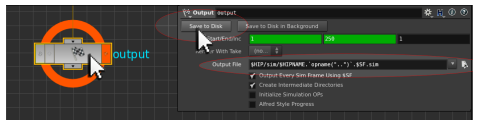
Number of Substeps	**50**
☑ **Use Parallel Constraint Solver**	
Constraint Iterations	**200**
Constraint Force Mix	**10**
Error Reduction Parameter	**0.4**

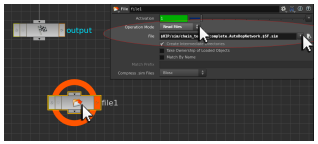




This will increase the accuracy of the chain collisions to ensure that no chain links break from each other, or penetrate through the teapot handle. When **PLAY** is pressed, the dynamic chain responds correctly for an underwater scene.

Before final rendering takes place, the dynamic chain network created by DOPs can be rendered out as a **series of .bgeo files** using the **Output DOP**.



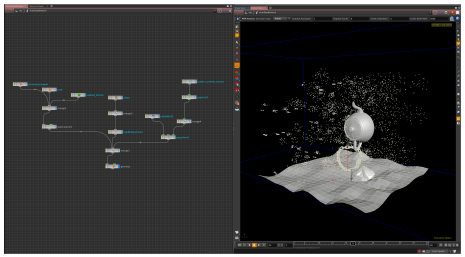


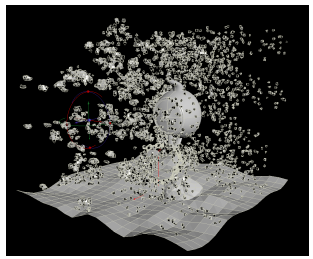
These **.sim files** can then be **read back into DOPs** using a **File DOP**. Set its **Operation Mode** to **Read Files**, and point the **File parameter** to the **rendered .sim files** on disk.

See file chain_teapot_complete.hipnc

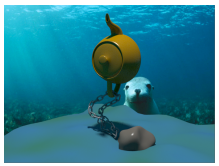
WATER CURRENT EFFECTS

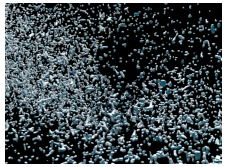
If **water currents** are required as additional underwater effects, they can be achieved using a **variation** on the **wobbly bubbles example** from the **Underwater Effects #1** lecture. **Open** the scene **chain_teapot_water_currents.hipnc**.

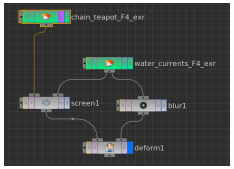


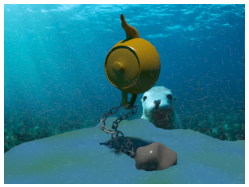


This scene contains the chain teapot example, added to with a simple particle system to help convey water currents. These particles are then rendered separately using a **Particle Fluid Surface SOP** and **Mountain SOP** to give them a blobbly shape.





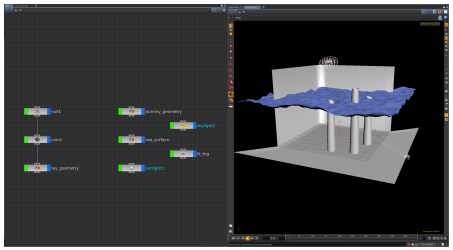


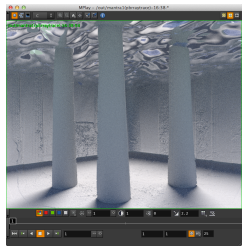


The blobby particle water current render can then be used in the composite to add **animated** water **particulate matter** as well as driving subtle underwater **deformation effects**.

WATER SURFACE CAUSTICS & CATHEDRAL RAY EFFECTS

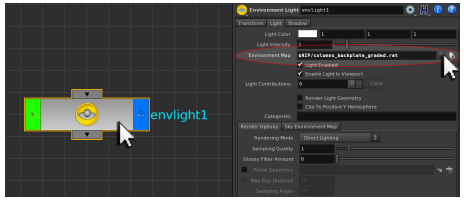
Animated water surfaces can be used to drive both underwater caustic effects and cathedral ray effects. **Open** the scene **water_temple_begin.hipnc**. This is the **temple dummy geometry scene**, added to with **an animated water surface** (with a **Basic Liquid Material** assigned), and **Sky Light** lighting created by the **Lights and Cameras Shelf**. A **lit_fog Atmosphere Object** has also been configured in preparation for cathedral ray effects, but is **currently inactive in the scene**.





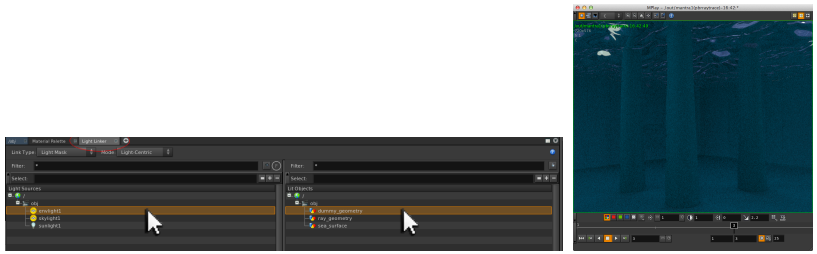
When the scene is rendered, a simple camera projected displacement can also be seen on the dummy geometry. At **Object Level**, create an **Environment Light**. In its **parameters** specify:

Environment Map **$HIP/columns_backplate_graded.rat**



This light will use the underwater graded version of the background plate to add overall environment lighting to the emerging underwater scene.

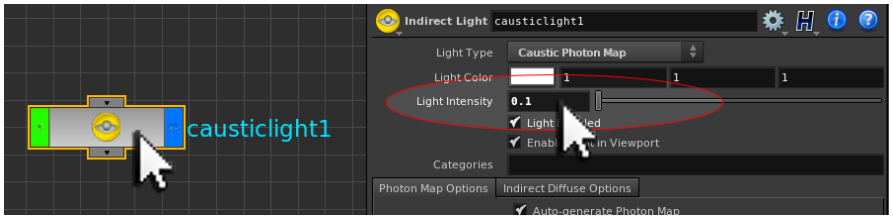
Activate a **Light Linker** as a **New Pane Tab Type** over the **Network Editor**, and set the **Environment Light** to **only illuminate** the **dummy_geometry**. Use the **Light Linker** to also set **skylight1** and **sunlight1** to **only illuminate** the **sea_surface**.



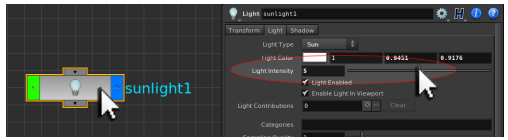
When the scene is rendered, the effect of this light linking can be seen.

Using the **Lights and Cameras Shelf** to add a **Caustics Light** to the scene. In its **parameters** specify:

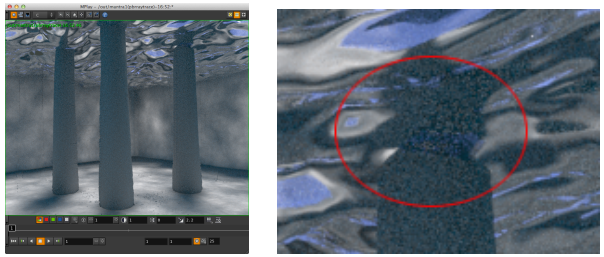
Light Intensity **0.1**



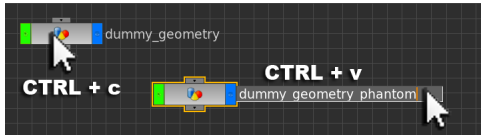
In the **parameters** for **sunlight1**, increase the **Light Intensity** value to **5**.



When the scene is rendered, the effect of the caustics on the scene can be observed.

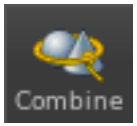


Closer examination of the render reveals that currently the sea surface is reflecting the dummy geometry. As the sea surface render will ultimately be used in a photo-real composite, it needs to reflect the underwater graded background plate instead. Activating this will ensure that any reflections of the scene in the sea surface match the graded background plate detail.

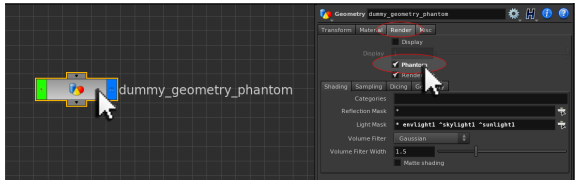


At **Object Level**, **Copy** (**CTRL + c**) and **Paste** (**CTRL + v**) the **dummy_geometry object** to create a second version of it. **Rename** this copy to **dummy_geometry_phantom**.

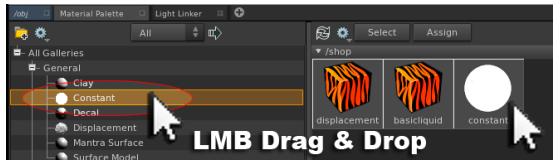
NOTE: If a number of **separate objects** are creating **dummy geometry**, they can be **combined** into a **single object** by using the **Combine tool** found on the **Modify Shelf**.



In the **Render parameters** for the **dummy_geometry_phantom** object, **activate** the **Phantom tick box**. This will ensure that this duplicate of the dummy geometry only appears in scene reflections rather than the main scene render.



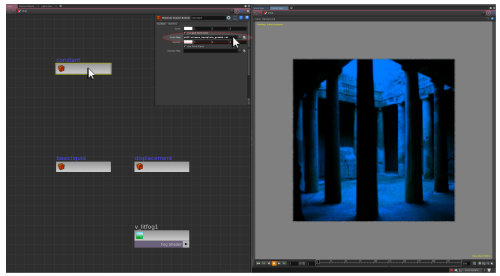
As both the **dummy_geometry object** and the **dummy_geometry_phantom object** both have the **displacement Material** assigned at **Object Level**; a **Constant Material** can be configured with the **graded version of the background plate** for **Geometry Level Material** assignment to **the dummy_geometry_phantom object**.



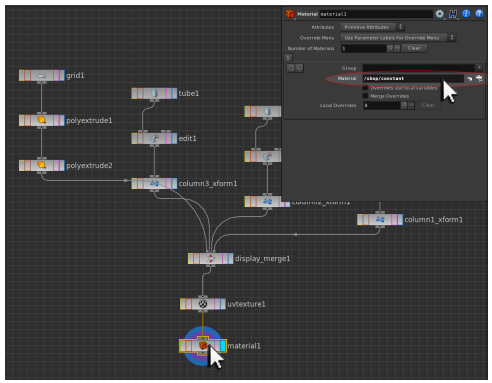
Using the **Material Palette**, **LMB Drag** and **Drop** a **Constant Material** into the **/shop palette area**.

At **SHOP Level**, locate the **Constant Material** and in its **parameters** specify:

Color Map **$HIP/columns_backplate_graded.rat**

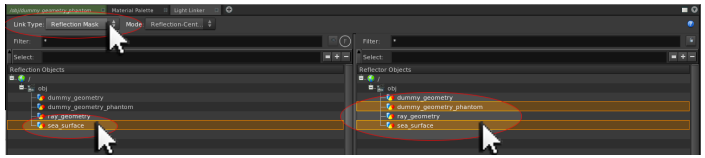


This will read in the graded version of the background plate that will return a constant render.

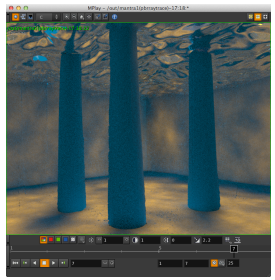


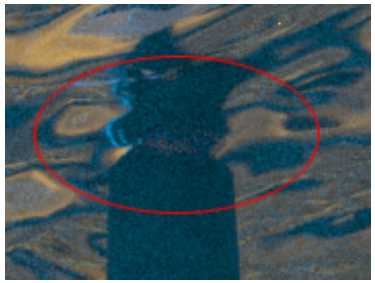
At the **Geometry Level** for the **dummy_geometry_phantom_object**, add a **Material SOP** to the **end** of the **network**, and **load into it** the **Constant Material**. As the displacement effect is assigned at Object Level, this additional material assignment will not interfere with the displacement effect, but simply add a constant render of the graded background plate to it.

In order to **remove** the **influence** of the **original dummy_geometry object** from the **sea surface render**, the **Light Linker** can be used. **Activate** the **Light Linker**, and set the **Link Type** parameter to a **Reflection Mask**.



Select the **sea_surface** from the **Reflection Objects list**, and **activate** only the **dummy_geometry_phantom object** and the **sea_surface object** in the **Reflector Objects** list. This will ensure that the sea surface only reflects itself and the phantom dummy geometry. When the **scene** is **rendered**, the **dummy_geometry object appears** in the **main render**; however the **dummy_geometry_phantom object appears** in the **sea_surface reflections**.

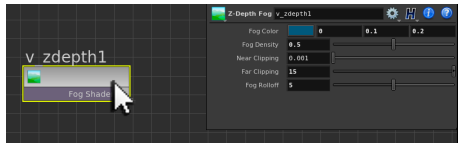


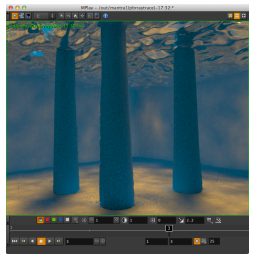


The **colours** of both **sunlight1** and **skylight1** can also be **modified** to prove that the dummy_geometry object is no longer influencing the sea_surface reflections. The colour of the caustics light can also be Channel Referenced to the colour of the sunlight object.

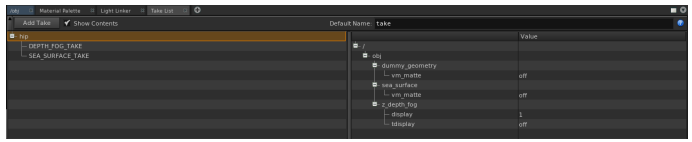
See file water_temple_stage1.hipnc

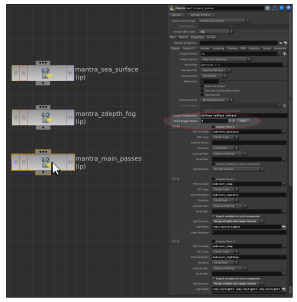
Z-Depth Fog can also be added as an additional effect to the underwater temple scene.



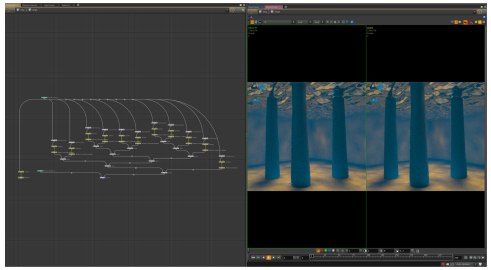


Takes and **Extra Image Planes** can also be activated to isolate the **sea_surface**, **depth_fog** and individual render components creating the main render (such as the caustics). For reference about configuring Takes and Image Planes, see the **Houdini Lectures #6 and #7**.





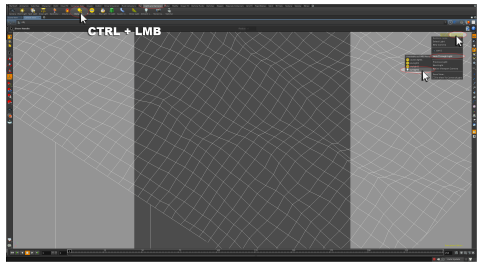
NOTE: The **reflection information** for the **sea_surface** containing the **dummy_geometry_phantom reflections** are part of the **Indirect Emission Image Plane**.



The various render passes can then be rebuilt in the composite to create an exact duplicate of the main beauty render. **See file water_temple_stage2.hipnc**

CREATING CATHEDRAL RAYS

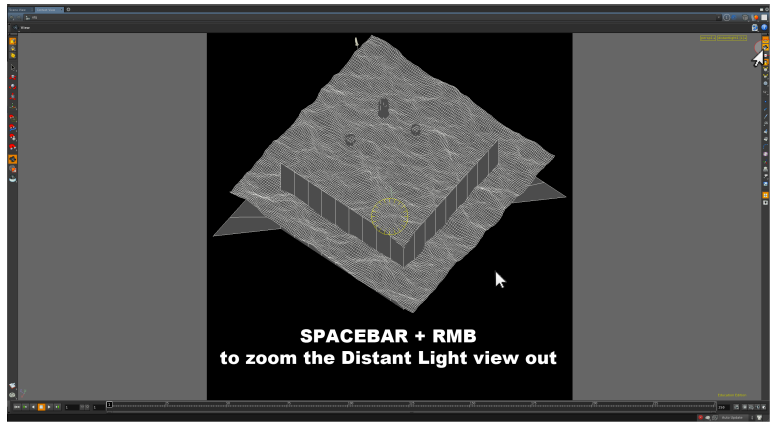
While the sea_surface animation will automatically drive the associated caustics effect on the dummy geometry, it would also be good to have the sea_surface drive cathedral rays as well. Configuring a new scene light specifically to generate cathedral rays can do this. At **Object Level**, **maximize** the **Viewer** and **set the view** to **look through** the **sunlight1** light.



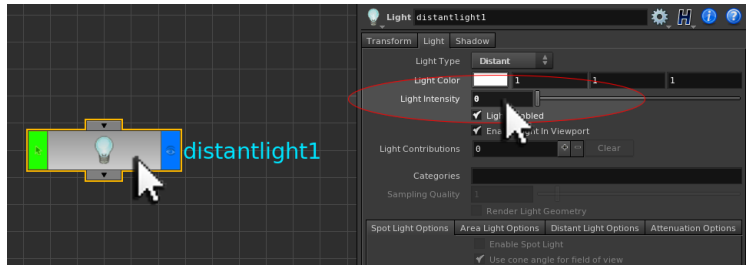
From the **Lights and Cameras Shelf**, **CTRL + LMB activate** a **Distant Light**. This will place the Distant Light at the exact same position of the sunlight1 light.

NOTE: A **spotlight** can be used instead of a Distant Light for a **Snell's Window Effect**.

Use the **Camera Lock button** to **lock the view** of this new **Distant Light** to the **Viewer**, and carefully **SPACEBAR + RMB zoom** the view backwards to reveal more of the sea_surface.



NOTE: The **position** of the new **Distant Light** can also be **offset** from the **sunlight1** position to allow for further bespoke angling of the resulting cathedral rays relative to the **refractive effect** of **water** on **light**. A **Spot Light** could also be used.



In the **parameters** for the **Distant Light** set the **Light Intensity** parameter to **0**. This will keep the light active in the scene, but it won't affect or influence the main render.

GENERATING THE CATHEDRAL RAYS

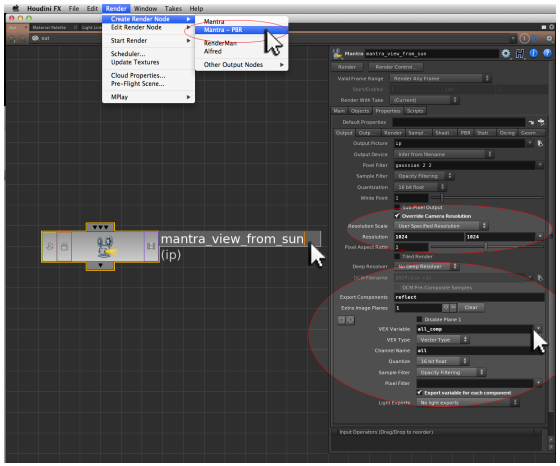
In order to generate a cathedral ray effect, the **view** from this **Distant Light** must be rendered independently. This is so the resulting images can be reassigned as a projection map on the distant light to create the cathedral rays. At **Outputs Level**, use the **main Render Menu** to create a **Mantra – PBR** render node, and **rename** this mantra node to **mantra_view_from_sun**. In its **parameters** specify:

Main >

Camera	**/obj/distantlight1**	

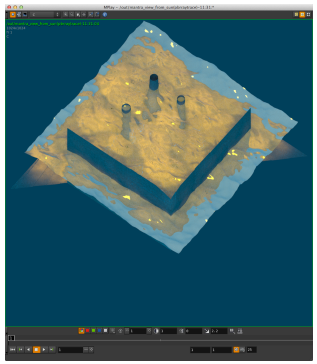
Properties >

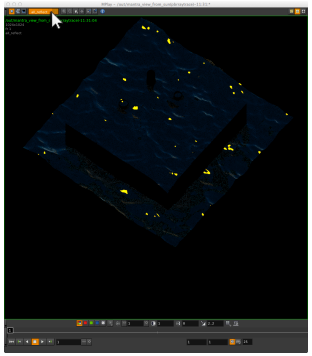
☑	**Override Camera Resolution**	
Resolution	**1024**	**1024**
Export Components	**reflect**	



Activate an **Extra Image Plane** and set the **associated VEX Variable parameter** to **export Combined lighting (per-component).**

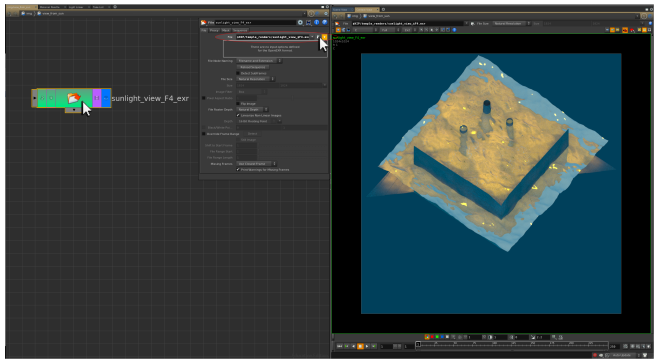
When a **test render** of the **mantra_view_from_sun node** is generated, the **all_reflect render pass** contains the information required to create the cathedral rays.





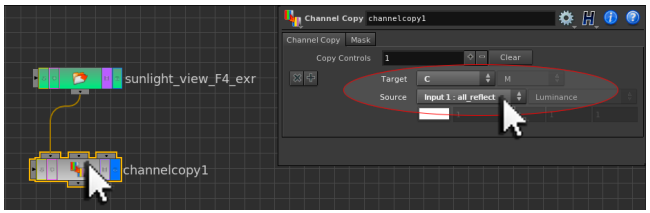
This **test render** can be **saved** out from **MPlay** as a **.exr file**, so it can be processed using the Houdini compositor to extract the **all_reflect** render pass.

At the **IMG Level of Houdini**, create a **new Image Network** and inside it use a **File COP** to read in the **sunlight_view_$F4.exr** file.



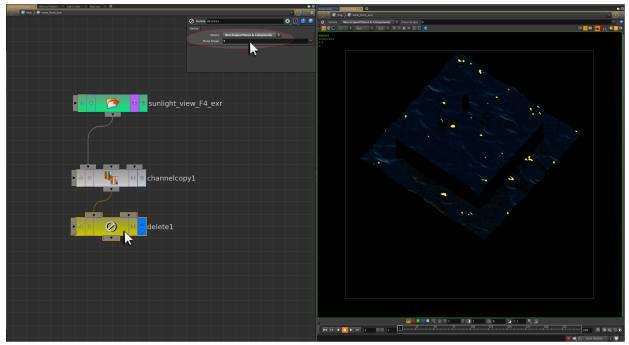
Append to the File COP a **Channel Copy COP**, and in its **parameters** specify:

Target	**C**
Source	**Input 1: all_reflect**



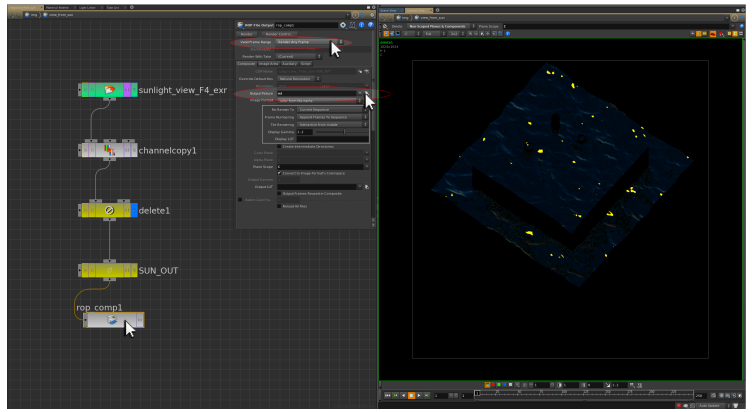
This will move the all_reflect image layer onto the main C-plane image layer. A **Delete SOP** can then be **appended** to the Channel Copy SOP to remove any excess image layers. In the **parameters** for the **Delete SOP** specify:

Delete	**Non-Scoped Planes and Components**
Plane Scope	**C**



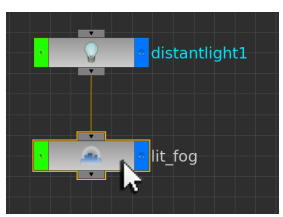
This will reduce the image down to the C-Plane information only.

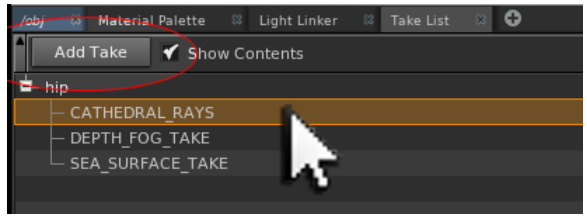
This **modified version** of the view from the sun render can then be **resaved to disk** as a **new .exr file** using a **ROP File Output COP**.



CREATING A CATHEDRAL RAY TAKE

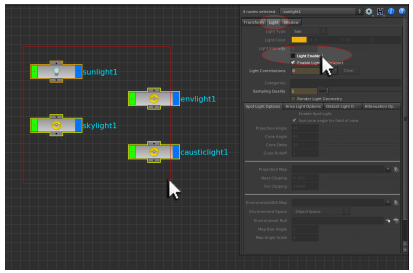
At **Object Level**, **parent** the **lit_fog Atmosphere Object** to the **Distant Light**.



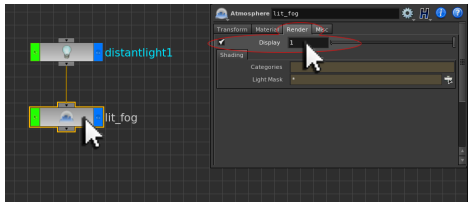


Activate the **Take List**, and from the **main hip take**, use the **Add Take Button** to create a **new Take** called **CATHEDRAL_RAYS**. With the CATHEDRAL_RAYS Take selected, make sure **Auto-Takes** are activated so that the scene can be modified in the context of this new take.

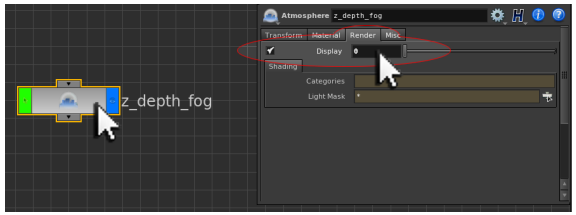
In the **Network Editor**, **select all** of the **scene lights apart from the Distant Light**, and in **their combined parameters deactivate** the **Light Enabled tick box**.



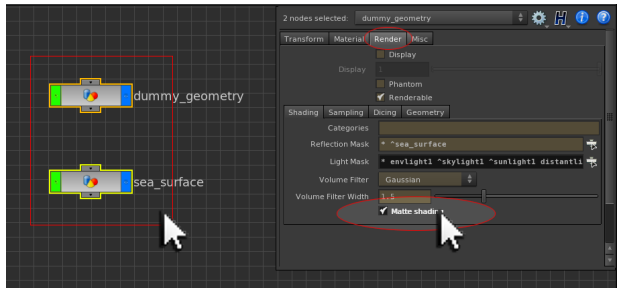
In the **Render parameters** for the **lit_fog Atmosphere Object**, activate the **Display parameter** setting a **value** of **1**. This will turn on the lit fog effect.



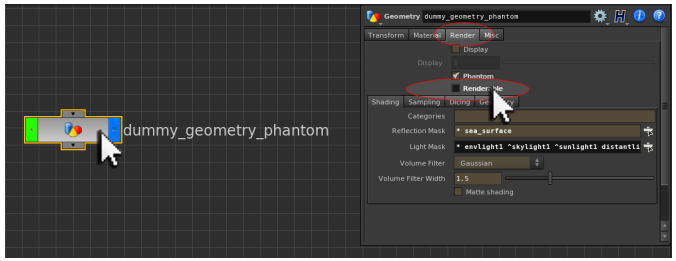
In the **parameters** for the **z_depth_fog Atmosphere Object**, **activate** the **Display parameter** and set a **value** of **0** to **turn off the z-depth fog** from the render.



Select the **dummy_geometry** object and the **sea_surface object**, and in the **Render** section of their **combined parameters activate** the **Matte shading tick box**.



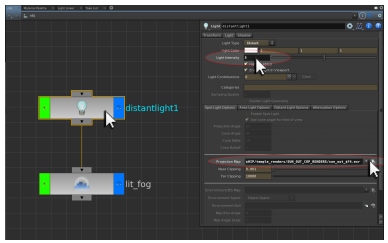
In the **Render parameters** for the **dummy_geometry_phantom object deactivate** the **Renderable** tick box.

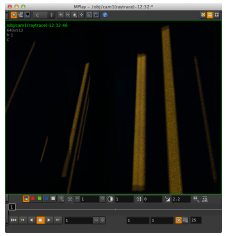


As a final step, activate the **Light section** of the **parameters** for the **Distant Light** and specify:

Light Intensity **5**

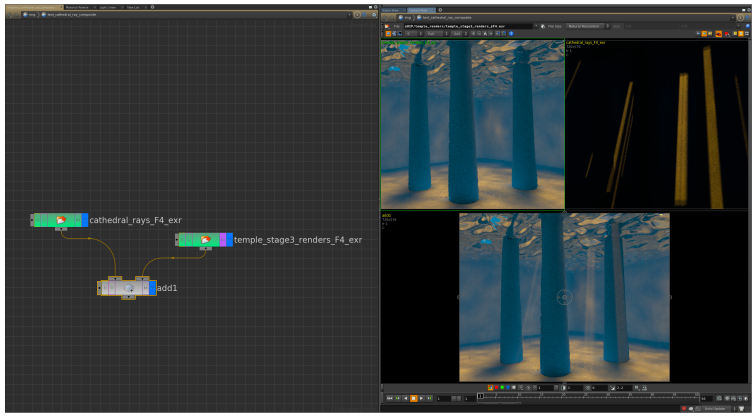
The **saved out .exr file** of the **all_reflect image plane** can then be loaded in as a **Projection Map**.





When the **CATHEDRAL_RAYS Take** is **rendered**, the cathedral rays can be seen. **See file water_temple_stage3.hipnc**

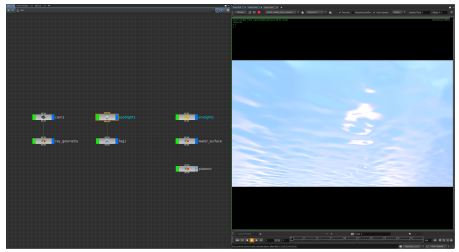
A **Mantra ROP** for rendering out the **CATHEDRAL_RAYS Take** can then be generated, and formal image sequence renders of the scene can then be created. Remember that before the CATHEDRAL_RAYS Take can be rendered, the view from the Distant Light must be rendered as an image sequence before hand.



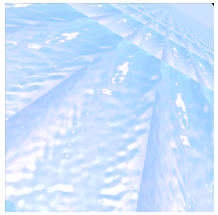
The resulting renders can then be tested in the composite to prove the cathedral ray effect is working correctly. **See file water_temple_complete.hipnc**

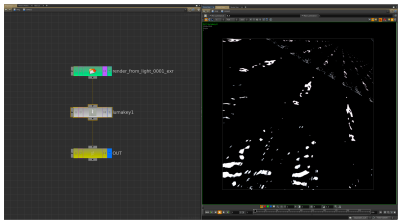
SNELL'S WINDOW

A **variation** on the **previous example** can help to create a Snell's Window effect. **See file snells_window**. In this scene, a **simple water surface** has been **created**, and **lit** using an **Environment Light** assigned with a **sun_sky.jpg Environment Map**. The **Render Light Geometry option** of the Environment Light has been **activated**. When the scene is rendered, the sun is refracted through the water surface.



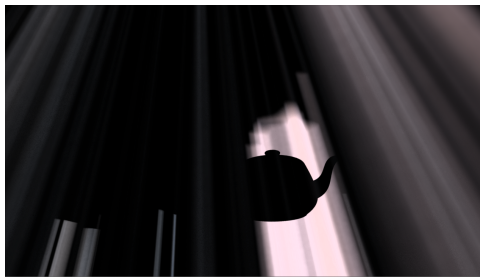
A **CAMERA_PROJECT_TAKE** has then been **created**, where this **initial render** view from the camera is **assigned back** onto the **water surface** as a **UV Texture camera projected constant Material**, with no scene lighting activated. This allows for an inactive **spotlight** to **render the scene from** the approximate **position of the sun**.



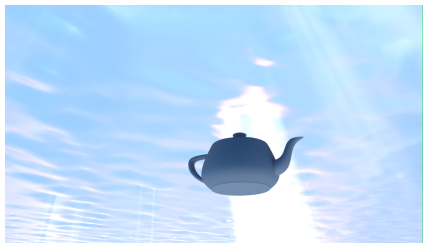


At the **IMG Level** of Houdini, a **simple compositing network** has been created to **extract a Luma Key** from the render_from_light image. This **Luma Key version** of the image is then **saved back to disk** for **use** as a **Projection Map** on the scene's spotlight.

A **CATHEDRAL_RAY TAKE** is then **created**. In this Take, the **Environment Light** is **deactivated**, and the **Spot Light activated**, with the **luma key Projection Map assigned**. **Lit Fog** is also **activated** on the **Spot Light**, and the **water_surace** is set to **render** as a **Matte Shading** object. A **Matte Shading teapot** has also been activated in the scene.



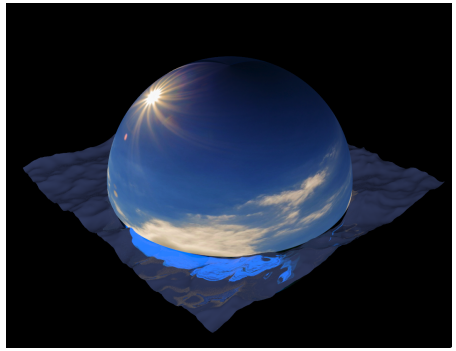
When this **Take** is **rendered**, the **cathedral rays** are **created**; **procedurally driven** from the **water surface highlights** that are **seen by the camera**.



A **final Take is created**, rendering the **Main Take**, but **with** the **teapot** now active in the render. These **two render passes** can then be **composited together** to create a **Snell's Window effect**.

OTHER SNELL'S WINDOW CONSIDERATIONS TO EXPERIMENT WITH…

An **Environment Map** could **consist** of **both** a **sky** and **underwater colour grade**. When **assigned** to an **Environment Light** as a map, these combined images would be textured as **two hemispheres** creating both **above water** and **below water lighting**, **if** the **water surface** is kept on the **origin's X-Z plane**.



The **Index of Refraction parameters** of the **Basic Liquid Material** can also be **modified beyond** the **normal values** for **water** (Inside IOR – 1.33; Outside IOR – 1). This can help bend refracted light further to help emphasize the Snell's Window effect for deeper water scenes.