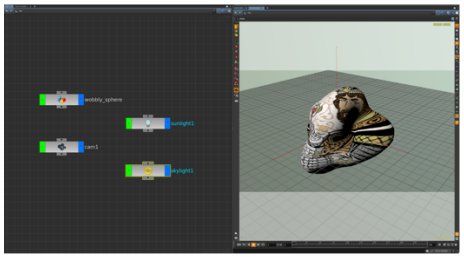
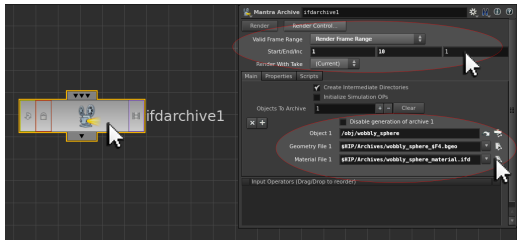
THE DELAYED LOAD RENDERING MECHANISM

Delayed Load Rendering is the process of **creating geometry archives** for all **the scene objects on disk**, which are then loaded for rendering at render time. This **saves Houdini** from **rebuilding geometry networks** for each **frame** being **rendered, making the process of rendering faster**. Before attempting to optimize a larger scene for rendering, it is a good idea to **examine** and **understand** the **Delayed Load Rendering** Mechanism on an extremely **simple scene**. This will help reveal the nuances of creating a Delayed Load Rendering Mechanism in a very clear way. **Open the scene wobbly_sphere_begin.hipnc**



This scene contains a **simple textured** and **animated wobbly sphere**, complete with a **simple lighting** and **camera setup** as well as a **Mantra PBR** rendering node. The **texture map** is **stored** in the **same directory** as the **scene file**, and can therefore be **called** into the **Material** using the **$HIP variable**. **Setting external files** relative **to $HIP** rather than using the full path, makes **the scene file directory portable** rather than tied to a fixed location or computer.



At the **OUT Level** of Houdini, create a **Mantra Archive ROP**. This will create an **archive** of the **wobbly sphere** on **disk** that can be used to **invoke** the **Delayed Load Rendering** mechanism. In the parameters for the Mantra Archive ROP specify:

Valid Frame Range	**Render Frame Range**		
Start/End/Inc	**1**	**10**	**1**
Main >			
Object 1	**/obj/wobbly_sphere**		
Geometry File 1	**$HIP/Archives/wobbly_sphere_$F4.bgeo**		
Material File 1	**$HIP/Archives/wobbly_sphere_material.ifd**		

When the **Render Button** is **pressed** for the Mantra Archive ROP, **a series of .bgeo files** will be created (baking the geometry animation to disk on a per frame basis), as well as a **.ifd text file** that will store **the configuration** of the wobbly_sphere's **material**.

NOTE: As a Mantra Archive ROP **does not yet** have the **ability** to **create** any **intermediate directories**, these should be **created by hand** before rendering of this node commences.

INSIDE A .IFD FILE

The **IFD format** creates a **text file on disk** that can be **examined** using a **Text Editor** such as **gedit**. It is **important** to **examine** the **.ifd files generated**, as their **contents** reveal that **$HIP** has been **hard converted** into the **full directory path**. This can have **implications** for the **portability** of a scene.

For example if the **wobbly_sphere_material.ifd file** is **examined** in a **text editor**, at the **start** of the **.ifd file commands** are **absolute path references** to the what the **$HIP variable was returning when** the **Render button** for the **archive mantra was pressed**.

```
# IFD created by Houdini Version: 14.0.498
# Generation Time: Nov 04, 2015 at 09:02:50
#   Render Target: mantra12.0
#         HIP File:
/Users/Phil/HERCULES_DOCUMENTS/NCCA_LECTURES/1516/HOUDINI/S1/DETOOLS/8.
H14_Render_Optimisation/8.H14_Render_Optimisation_PUBLISH/WOBBLY_SPHERE/
wobbly_sphere_begin, $T=0.36, $FPS=25
#   Output driver: /out/ifdarchive1
ray_version VEX14.0.498
ray_declare global float global:fps 25

setenv HIP =
"/Users/Phil/HERCULES_DOCUMENTS/NCCA_LECTURES/1516/HOUDINI/S1/DETOOLS/8.
H14_Render_Optimisation/8.H14_Render_Optimisation_PUBLISH/WOBBLY_SPHERE"
# Save materials for /obj/wobbly_sphere/material1 at time 0.36
# Corresponding geometry file:
/Users/Phil/HERCULES_DOCUMENTS/NCCA_LECTURES/1516/HOUDINI/S1/DETOOLS/8.
H14_Render_Optimisation/8.H14_Render_Optimisation_PUBLISH/WOBBLY_SPHERE/Archives/
wobbly_sphere_0010.bgeo
ray_start material  # {
    ray_property object name "/shop/mantrasurface"
    ray_property object displacebound 0.0500000000000000028
    ray_property object id 36171
ray_startshader vfl surface op:/shop/mantrasurface
#ifndef VOP_SHADING
#define VOP_SHADING
#endif
#ifndef VOP_SURFACE
#define VOP_SURFACE
#endif
```

This has **implications** for the **portability** of these **.ifd files**; because if the **project directory changes** location (**for example** it is **moved** to **/transfer/Aquascape/Archives**), then these **associated archive .ifd files** would **return path errors** and in turn **affect** the **main scene renders**.

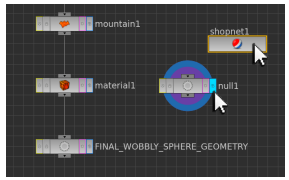
This means **rendering .ifd archives** should **only take place** when the **final $HIP location** has been **established**.

When **rendering** to the **NCCA Render Farm** for example, **project files** will need to be **transferred over to /render**, at which point the **$HIP location** would **change**, and **any .ifd archives** would **need to be regenerated** before any main rendering takes place.

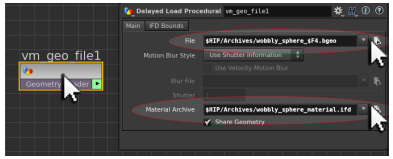
ACTIVATING DELAYED LOAD RENDERING FOR THE WOBBLY SPHERE

At **Geometry Level** for the **wobbly_sphere object**, create a **Null SOP** and a **SHOP Network**. **Activate** the **Display/Render Flag** on the **Null SOP**.

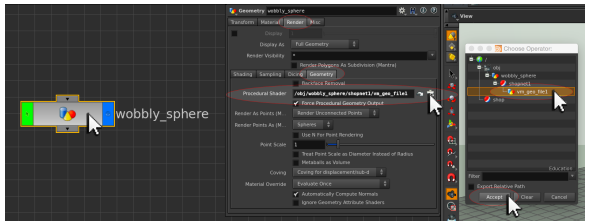
A **SHOP Network** is a custom **SHOP Level** that can be created anywhere inside the Houdini scene. In the context of a Delayed Load setup, a special shader will be placed inside this SHOP Network and used to call the Archive files at render time.



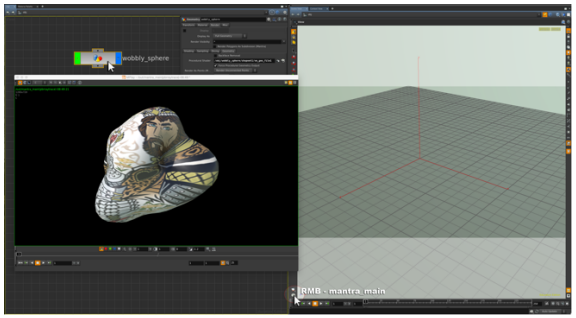
Inside the **SHOP Network** create a **Delayed Load Procedural Shader**, and in its **parameters load** in **both** the **wobbly_sphere_$F4.bgeo** sequence and the **wobbly_sphere_material.ifd** files, using $HIP.



Return to **Object Level**, and in the **Render > Geometry parameters** for the **wobbly_sphere object**, **load** in the **Delayed Load Procedural shader** into the **Procedural Shader parameter**.



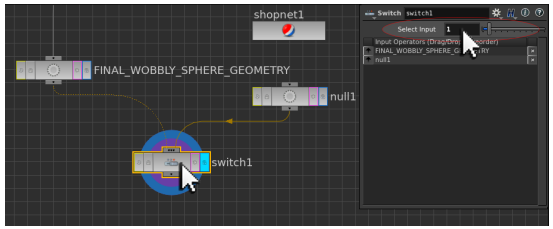
Although the **scene** now **appears** to be **empty**, when **a formal render** is **activated**, the **wobbly_sphere** is **rendered** as normal due to **Mantra reading** the associated **archives** to **generate** the **image**.



See file wobbly_sphere_stage1.hipnc

MAKING THE DELAYED LOAD MECHANISM PORTABLE

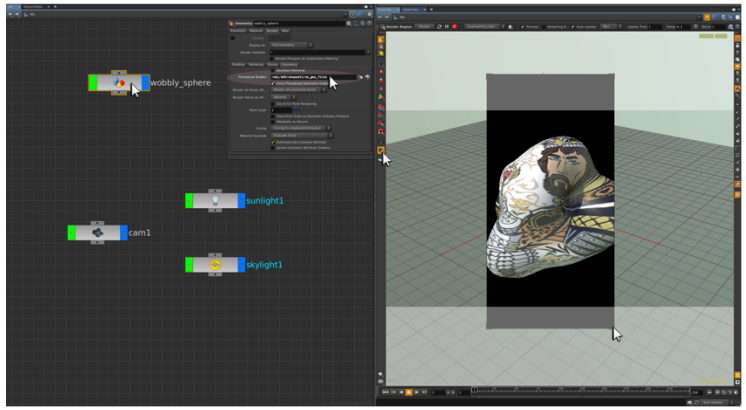
At **Geometry Level** for the **wobbly_sphere**, create a **Switch SOP** and wire the **FINAL_WOBBLY_SPHERE_GEOMETRY Null SOP** as the **first input**, and the **empty Null SOP** as its **second input**.



In the **parameters** for the **Switch SOP** specify a **Select Input** value of **1** to **activate** the **empty Null SOP** as the network being evaluated.

In the **Render > Geometry parameters** for the **wobbly_sphere object**, **modify** the **Procedural Shader parameter** to read:

Procedural Shader **/obj/$OS/shopnet1/vm_geo_file1**

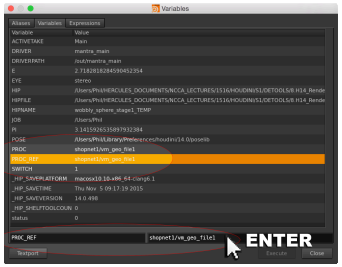


A **Render Region Preview** reveals that the archives are **still rendering correctly**. This is due to the **$OS variable** automatically **returning** the **name** of the **current object**.

CREATING CUSTOM VARIABLES

Custom variables can also be created to control both the Switch SOP and the remainder of the Procedural Shader path. From the **main Edit Menu**, choose **Aliases and Variables**, and under the **Variables section** of the resulting **dialog window**, **activate three custom variables**:

<variable name>	<value>
SWITCH	**1**
PROC	**shopnet1/vm_geo_file1**
PROC_REF	**shopnet1/vm_geo_file1**

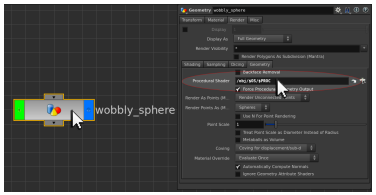


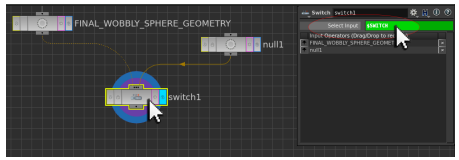
These **custom variables** can then be **called** around the Houdini environment by **using $SWITCH**, **$PROC**, and $PROC_REF.

NOTE: $PROC_REF will be used as a **reference variable**, allowing **$PROC** to **be set to <empty>** to **turn off** the Delayed Load Mechanism. The **value** of **$PROC_REF** can then be **re-copied back into** $**PROC** in order to **turn back on** the **Delayed Load Mechanism**.

Modify the **Render > Geometry Procedural Shader parameter** of the wobbly_sphere object to read:

Procedural Shader **/obj/$OS/$PROC**

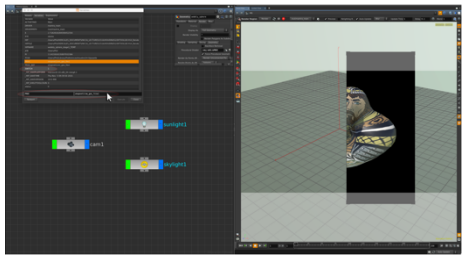




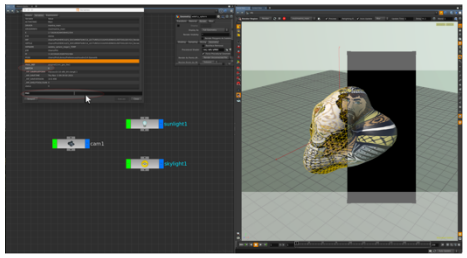
Inside the wobbly_sphere object, **modify** the **Select Input parameter** of the **Switch SOP** to read:

Select Input **$SWITCH**

Now the **Delayed Load Mechanism** can be **fully activated** and **deactivated** using the **Aliases and Variables window**.



When **$PROC** is set to **shopnet1/vm_geo_file1** and **$SWITCH** is set to **1**, the **Delayed Load Render Mechanism** is **active**, and the **empty scene** will be **optimized for rendering**. When **$PROC** is set to **<empty>** and **$SWITCH** is set to **0**, the **Delayed Load Render Mechanism** is **deactivated**, and the **live scene** can be **modified and tweaked** before final rendering commences.



These **modifications** of the **$PROC** and **$SWITCH variables** can also be **called automatically** at **render time** using simple **HScripts** activated on the **Mantra ROP** and **Mantra Archive ROP**.

See file wobbly_sphere_stage2.hipnc

CREATING HSCRIPTS TO CONTROL VARIABLES

Open up a **Text Editor** (for example **gedit**), and **type** in the following **HScript commands**:

archive_script.hsc

```
#change the $SWITCH VARIABLE TO 0
set -g SWITCH = 0

#change the $PROC VARIABLE TO EMPTY
set -g PROC = " "

#FORCE THE VARIABLES TO UPDATE
varchange
```

Save this **file** in the **$HIP directory** as **archive_script.hsc**.

NOTE: Any **line** of the script **beginning** with **#** will be **ignored** when the script is run, allowing for comments to be added to the script for legibility as to what is occurring.

Create a **new HScript** and **type** in the **following commands**:

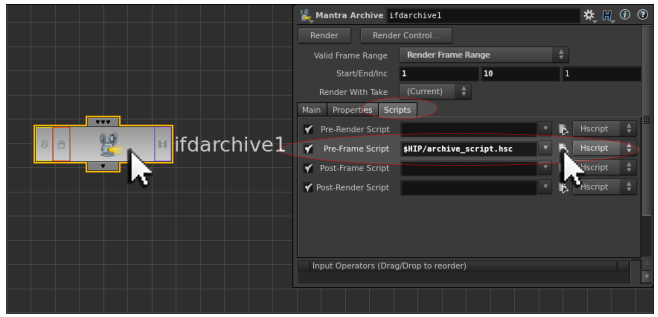
render_script.hsc

```
#change the $SWITCH VARIABLE TO 1
set -g SWITCH = 1

#change the $PROC VARIABLE TO THE SHADER PATH
set -g PROC = "shopnet1/vm_geo_file1"

#FORCE THE VARIABLES TO UPDATE
varchange
```

Save this **file** in the **$HIP directory** as **render_script.hsc**.

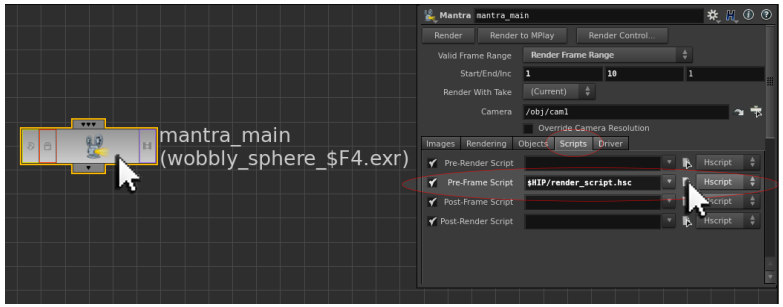
Under the **Scripts section** of the **parameters** for the **Mantra Archive ROP** specify:

☑ **Pre-Frame Script** **$HIP/archive_script.hsc**



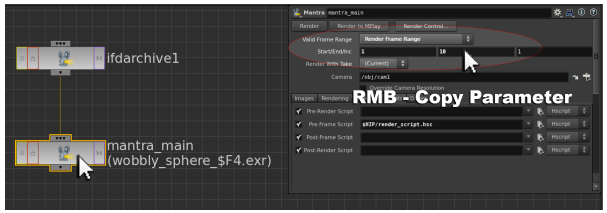
Similarly, under the **Scripts section** of the **parameters** for the **Mantra ROP** specify:

☑ **Pre-Frame Script** **$HIP/render_script.hsc**

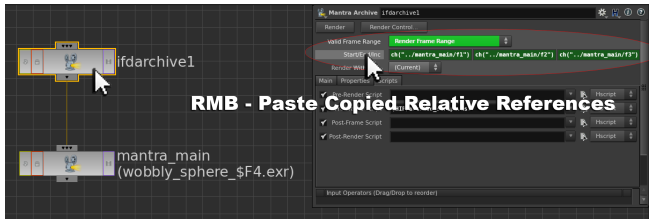


This will ensure that **before each frame** is **rendered** (either the **Archive frame** or the **rendered image frame**), that **$SWITCH** and **$PROC** are **set correctly**.

The **Mantra Archive ROP** and the **Mantra ROP** can now be **daisy-chained together**.



Channel Referencing can then be used to **synchronize** the **Mantra ROP Frame Range** with the **Mantra Archive Frame Range**.



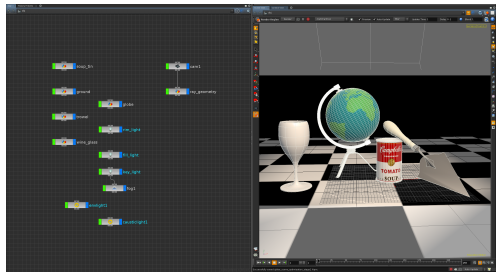
Now when the **mantra_main node** is **rendered**, the **Mantra Archive ROP** will be **rendered first**, with the **archive_script.hsc ensuring** that **$SWITCH** and **$PROC** are **set correctly** to **archive** the **live scene**. **Similarly**, **after** the **first frame of the Mantra Archive ROP** has been **rendered**, the **$SWITCH** and **$PROC variables** will **be reset** to the **archived scene** in order to **render** the **first frame** of the **image sequence**.

This means that the $HIP directory and its contents are now fully portable to any rendering location desired.

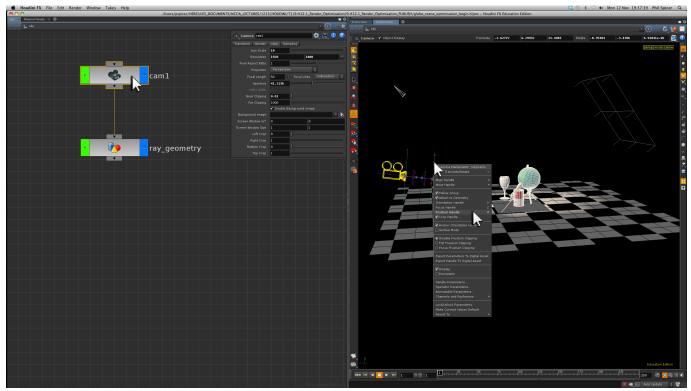
See file wobbly_sphere_complete.hipnc

NEAR AND FAR CLIPPING PLANES FOR LIGHTS AND CAMERAS

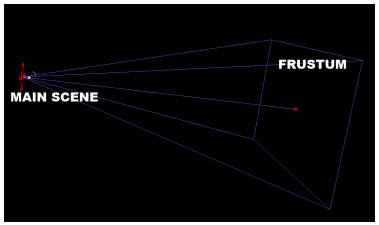
In order to ensure that only the area around a scene is factored at render times, the **Near** and **Far Clipping planes** for both **Lights** and **Cameras** can be **adjusted** so that they only **encompass** the **main render area**. This can reduce render times, as areas outside the Near and Far Clipping Planes will not be calculated. **Open the scene globe_scene_optimisation_begin.hipnc**.



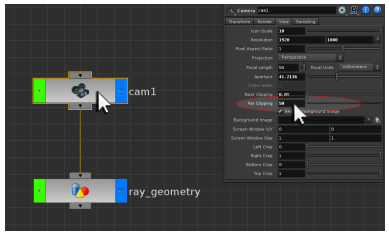
This is a modified version of the Globe Scene with the **globe rotating on its central spindle**. In the **Network Editor**, select the **camera object**, and with the mouse over the **Viewer**, press **ENTER** to go into **Camera Tool Mode**.

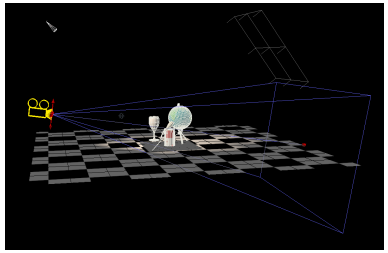


RMB on the Orientation Handle of the camera in the Viewer, and from the resulting menu choose **Frustum Handle**.



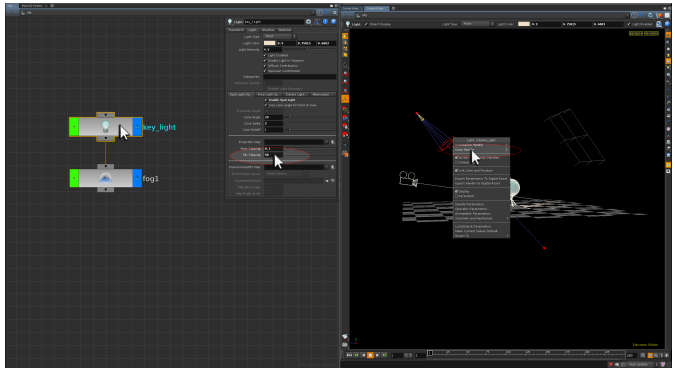
Currently, the Frustum Handle goes far beyond the Main Scene object area**.** Modifying the Far Clipping Plane parameter for the camera can reduce the area that the frustum area covers.





In the **View parameters** for the camera, reduce the **Far Clipping parameter** to **60**. This will create a frustum area for the camera that just encompasses the main scene also allowing for the lit fog. **NOTE: If clipping planes are set too low, big black triangles will appear in the render.**

The **Far Clipping parameter** for the [spotlight] **key_light** can also be adjusted to this value, so that it just encompasses the main scene. **Activating the Cone Handle option for any direct light (no matter what type)** will also reveal its far clipping distance as a line projected out from the cone light to help visualize the far clipping region.



Repeat this process for each **direct scene light** in the scene, setting its **far clipping** parameter to **60**.

NOTE: Indirect Lights such as the Environment Light and Caustics Light do not have Far or Near Clipping parameters.

DELAYED LOAD RENDERING

As **Takes** and **Image Plane Render Passes** have been created for this scene, this will need to be **factored** into the **Delayed Load Rendering Setup**. When creating Delayed Load Rendering, geometry and material archive files will be stored on disk. These files can be stored relative to the **$HIP file location** (ie in a subdirectory) so that Houdini can locate these files at render time.

CONFIGURING THE STATIC OBJECTS

The **static scene objects** can all be archived simultaneously for **Delayed Load Rendering**. As the **globe object** is animated, it will need to be considered separately to the other static scene objects. Switch to the **OUT Level** of Houdini and here create a **Mantra Archive ROP**. Rename this node to **static_objects_main_take_archive**.

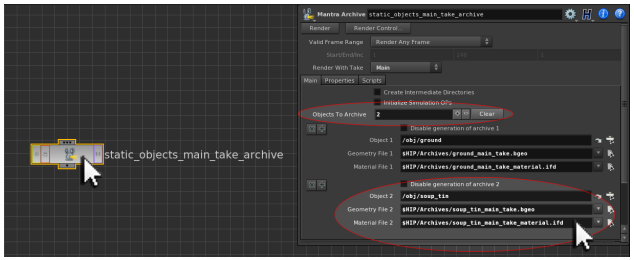
In the **Parameters** for this **Archive ROP** specify:

Render With Take	**Main**
Object 1	**/obj/ground**
Geometry File 1	**$HIP/Archives/ground_main_take.bgeo**
	<create the Archives subdirectory if necessary>
Material File 1	**$HIP/Archives/ground_main_take_material.ifd**

The **.bgeo archive file** will contain all the **geometry information** for the **ground object**, whilst the **.ifd archive file** will contain all of its **material configuration** for the **main take**.

Increase the **Objects to Archive** number to **2**, and in the **Object 2 parameters** specify:

Object 2	**/obj/soup_tin**
Geometry File 2	**$HIP/Archives/soup_tin_main_take.bgeo**
Material File 2	**$HIP/Archives/soup_tin_main_take_material.ifd**



Increase the **Objects to Archive** number to **3**, and in the **Object 3 parameters** specify:

Object 3	**/obj/trowel**
Geometry File 3	**$HIP/Archives/trowel_main_take.bgeo**
Material File 3	**$HIP/Archives/trowel_main_take_material.ifd**

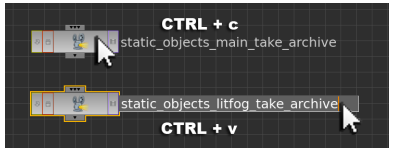
Increase the **Objects to Archive** number to **4**, and in the **Object 4 parameters** specify:

Object 4	**/obj/wine_glass**
Geometry File 4	**$HIP/Archives/ wine_glass_main_take.bgeo**
Material File 4	**$HIP/Archives/wine_glass_main_take_material.ifd**

See file globe_scene_optimisation_stage1.hipnc

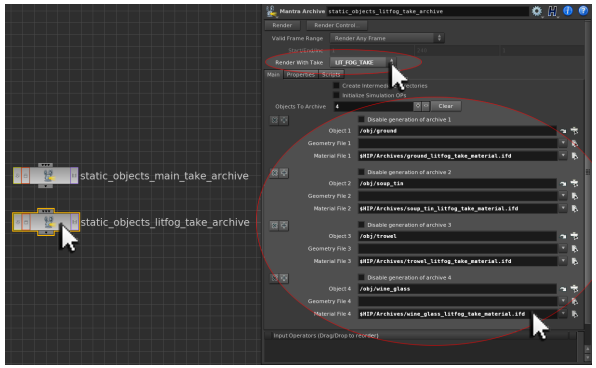
CONFIGURING THE STATIC OBJECTS LIT FOG PASS ARCHIVE

Copy (**CTRL + c**) and **Paste** (**CTRL + v**) the **static_objects_main_take_archive node**, and **rename this copy** to **static_objects_litfog_take_archive**.



In its **Parameters** specify:

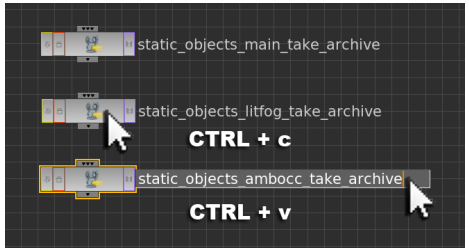
Render With Take	**LIT_FOG_TAKE**
<Main>	
Object 1	**/obj/ground**
Geometry File 1	<empty>
Material File 1	**$HIP/Archives/ground_litfog_take_material.ifd**
Object 2	**/obj/soup_tin**
Geometry File 2	<empty>
Material File 2	**$HIP/Archives/soup_tin_litfog_take_material.ifd**
Object 3	**/obj/trowel**
Geometry File 3	<empty>
Material File 3	**$HIP/Archives/trowel_litfog_take_material.ifd**
Object 4	**/obj/wine_glass**
Geometry File 4	<empty>
Material File 4	**$HIP/Archives/wine_glass_litfog_take_material.ifd**



NOTE: As the **geometry** for the **static scene objects** is the **same**, there is **no need** to **render specific geometry archives** for the **LIT_FOG_TAKE**. Similarly, the only **material differences** for the **LIT_FOG_TAKE** are for the **wine_glass object** (all the others simply have the Matte Shading option specified for this Take); however, as the **AMB_OCC_TAKE** has significant material differences, configuring the **static_objects_litfog_take_archive** in this way will make configuration of the **AMB_OCC_TAKE** archive much simpler.

CONFIGURING THE STATIC OBJECTS AMB OCC PASS ARCHIVE

Copy (**CTRL + c**) and **Paste** (**CTRL + v**) the **static_objects_litfog_take_archive node**, and **rename this copy** to **static_objects_ambocc_take_archive**.



In its **Parameters** specify:

Render With Take	**AMB_OCC_TAKE**
Object 1	**/obj/ground**
Geometry File 1	<empty>
Material File 1	**$HIP/Archives/ground_ambocc_take_material.ifd**
Object 2	**/obj/soup_tin**
Geometry File 2	<empty>
Material File 2	**$HIP/Archives/soup_tin_ambocc_take_material.ifd**

Object 3	**/obj/trowel**
Geometry File 3	<empty>
Material File 3	**$HIP/Archives/trowel_ambocc_take_material.ifd**
Object 4	**/obj/wine_glass**
Geometry File 4	<empty>
Material File 4	**$HIP/Archives/wine_glass_ambocc_take_material.ifd**

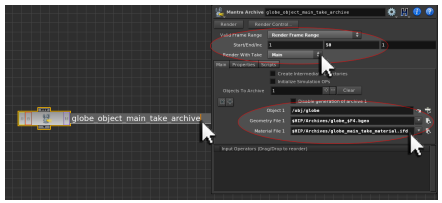
Mantra Archives nodes have now been created for all of the static scene objects, and each associated Take. The only remainder archives to generate are for the animated globe object.

See file globe_scene_optimisation_stage2.hipnc

CREATING ARCHIVES FOR THE GLOBE

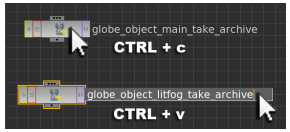
At **OUT Level** create a new **Mantra Archive ROP** for the **Main Take** of the **globe**. Rename this Mantra Archive to **globe_object_main_take_archive**. In its **parameters** specify:

Valid Frame Range	**Render Frame Range**		
Start/End/Inc	**1**	**10**	**1**
Render with Take	**Main**		
Object	**/obj/globe**		
Geometry File	**$HIP/Archives/globe_main_take_$F4.bgeo**		
Material File	**$HIP/Archives/globe_main_take_material.ifd**		



NOTE: As the **globe** has **Geometry Level animation** assigned to it (the spinning globe on its spindle), **a series of .bgeo files** must be generated (**using $F4**).

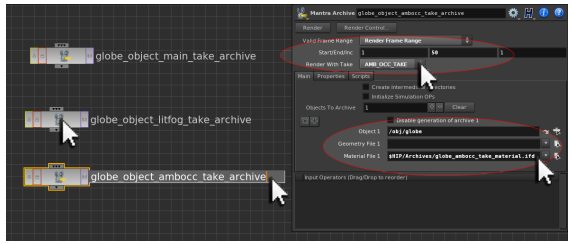
Copy (**CTRL + c**) and **Paste** (**CTRL + v**) this node, and **rename** the copy to **globe_object_litfog_archive**.



In its **parameters** specify:

Valid Frame Range	**Render Frame Range**		
Start/End/Inc	**1**	**10**	**1**
Render with Take	**LIT_FOG_TAKE**		
Object	**/obj/globe**		
Geometry File	<empty>		
Material File	**$HIP/Archives/globe_litfog_take_material.ifd**		

Copy (**CTRL + c**) and **Paste** (**CTRL + v**) the **globe_object_litfog_archive** node, and **rename** the copy to **globe_object_ambocc_archive**.

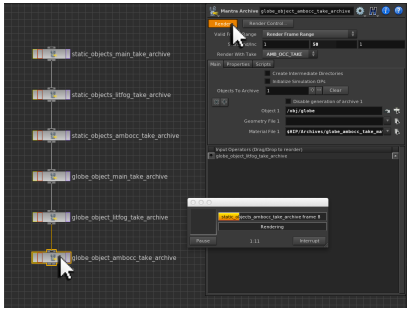


In its **parameters** specify:

Valid Frame Range	**Render Frame Range**
Start/End/Inc	**1 10 1**
Render with Take	**LIT_FOG_TAKE**
Object	**/obj/globe**
Geometry File	<empty>
Material File	**$HIP/Archives/globe_ambocc_take_material.ifd**

With all the **archive nodes configured**, they can be **daisy-chained together** and **rendered simultaneously** from the **base node** of the chain.

NOTE: As the **globe_object archives have an animation frame assigned**, they should be assigned as the **base nodes of the chain**. This will **ensure the full frame range** is **rendered**, rather than a single frame.



When the **Archives** have been **rendered out to disk**, **save** a version of **the scene** in the **$HIP directory** called **globe_scene_optimisation_<yourname>.hipnc**

See file globe_scene_optimisation_stage3.hipnc

GENERAL ARCHIVE GUIDELINES

For **static objects** that are only **animated** at **Object Level**, a **single geometry file** or **material archives** need to be generated. If **geometry** has a **deforming** or **animated surface**; or the **surface** has a **material** which **changes over time**, the **$F4 frame number** command would **need to be set** in the **archive output name** and a **frame range rendered**.

If an **object** has a **fixed shape** for its **geometry** but a **changing surface**, **only** a **$F4 series of material archives** would need to be **generated alongside** a **single .bgeo geometry** archive.

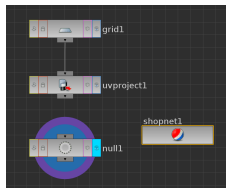
If an **object** has a **changing shape** at **Geometry Level**, but a **fixed surface material**; a **$F4 series of geometry .bgeo archives** would need to be **generated alongside** a **single .ifd material archive**.

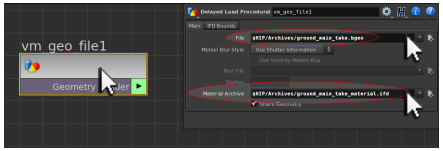
CREATING A DELAYED LOAD RENDERING SETUP

Once archive files have been created, a **Delayed Load Rendering Setup** can be **initiated** on a **per object basis**. As each object is activated as a Delayed Load Render, it will need to be checked in the final Mantra Renders to ensure everything is working as expected.

ACTIVATING DELAYED LOAD FOR THE GROUND OBJECT

At **Geometry Level** for the **ground object**, create a **Null SOP** and a **SHOP Network** as standalone nodes. **Activate** the **Display & Render Flags** for the **Null SOP**.

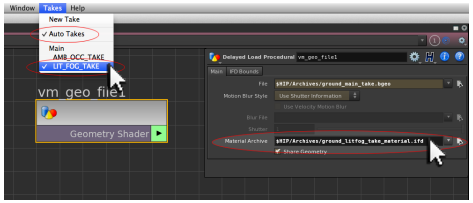




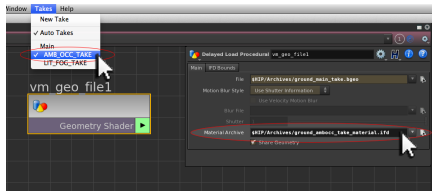
Double LMB on the **SHOP Network** to go inside it. Here create a **Delayed Load Procedural shader**. In its **parameters** specify:

File	**$HIP/Archives/ground_main_take.bgeo**
Material Archive	**$HIP/Archives/ground_main_take_material.ifd**

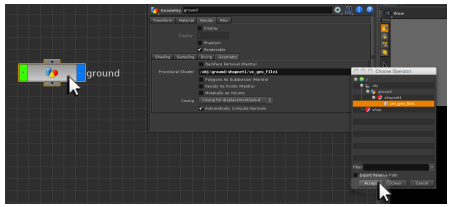
Activate the **LIT_FOG_TAKE** and the **Auto Takes** option. For the **LIT_FOG_TAKE**, replace the **Material Archive** reference with **ground_litfog_take_material.ifd**.



This will ensure that the ground renders with the correct Material Archive relative to the Take. Repeat this for the **AMB_OCC_TAKE**, replacing the **Material Archive** reference with **ground_ambocc_take_material.ifd**

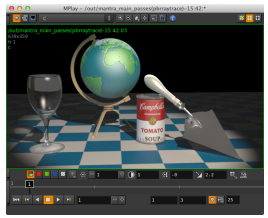


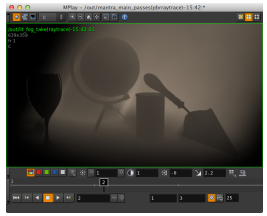
Reset Houdini back to the **Main Take**, and switch to **OBJ Level**. In the **Render > Geometry Parameters** for the **ground object**, load in the **Delayed Load Procedural shader** into the **Procedural Shader parameter**.

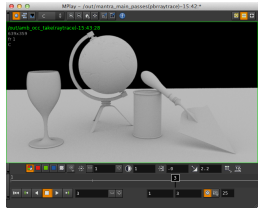


This will **override** the **Null SOP** at **Geometry Level** (and indeed any other nodes), forcing the object to load in the Archive files created instead.

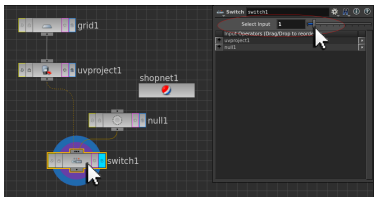
The **success** of the **Delayed Load setup** for the **ground object** can be authenticated by creating **Render Region Previews** for the **Main Take**, **LIT_FOG_TAKE**, and **AMB_OCC TAKE** or by formally rendering out the **output mantras** to **MPlay**.







When the **Delayed Load setup** for the **ground object** has been **verified**; it's a **Switch SOP** can be created, with the **main geometry network** as the **first input**, and the **Null SOP** as the **second input**. In the **parameters** for the **Switch SOP** set a **Select Input** value of **1** to explicitly **call the Null SOP** instead of the main geometry network.

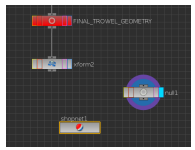


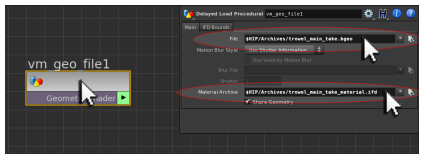
This Switch SOP network will allow for easy switching between the geometry construction network and the Delay Load Rendering setup; as a contingency in case of any issues if Archives need to be regenerated (after for example the $HIP location has been altered).

See file globe_scene_optimisation_stage4.hipnc

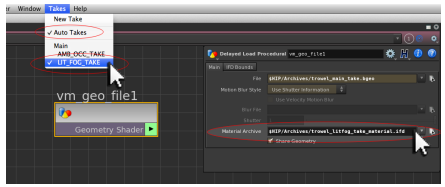
ACTIVATING DELAYED LOAD FOR THE TROWEL OBJECT

At the **Geometry Level** of the **trowel object**, create a **Null SOP** and a **SHOP Network**. **Activate** the **Display and Render Flags** for the **Null SOP**.

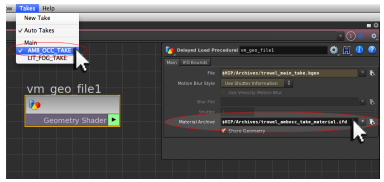




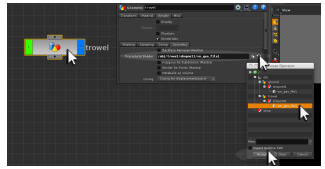
Inside the **SHOP Network**, create a **Delayed Load Procedural Shader**. Ensure that the correct **Archive files** are loaded for the **Main Take**.



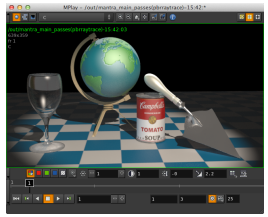
Set the Take to the **LIT_FOG_TAKE** and activate **Auto Takes**. As before trade out the **Material Archive parameter** for the trowel for the **trowel_litfog_take_material.ifd** file.

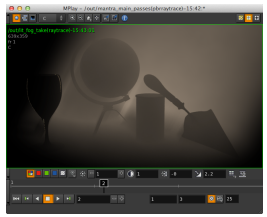


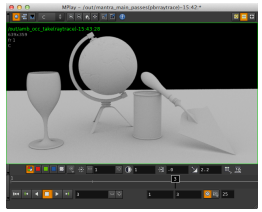
Repeat this process for the **AMB_OCC_TAKE**, replacing the **Material Archive parameter** for the trowel for the **trowel_ambocc_take_material.ifd** file.

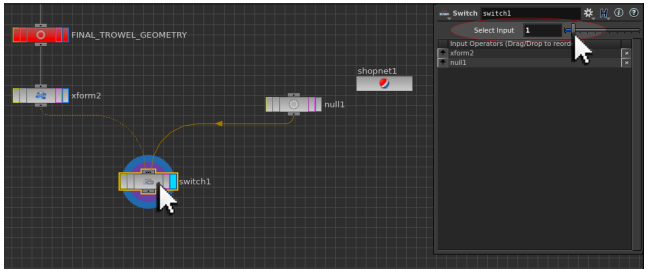


Set Houdini back to the **Main Take**, and at **OBJ Level** go to the **Render > Geometry** section of the **Parameters** for the **trowel object**. Here load in the trowel's Delayed Load Procedural shader into the Procedural Shader parameter. Once again, test renders of the scene can be produced to ensure the Delayed Load mechanism is working as expected.







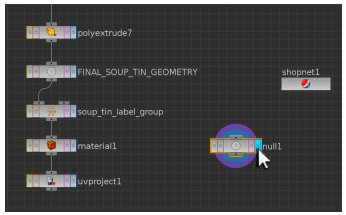


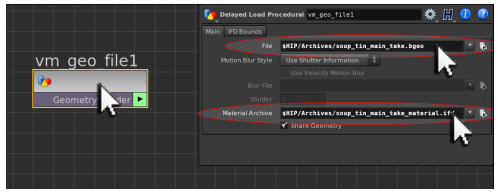
When the **Delayed Load setup** for the **trowel object** has been **verified**; it's a **Switch SOP** can be created, with the **main geometry network** as the **first input**, and the **Null SOP** as the **second input**. In the **parameters** for the **Switch SOP** set a **Select Input** value of **1** to explicitly **call the Null SOP** instead of the main geometry network.

See file globe_scene_optimisation_stage5.hipnc

CREATING A DELAYED LOAD FOR THE SOUP TIN

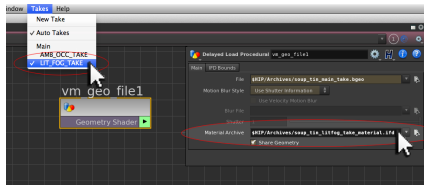
At the **Geometry Level** of the **soup_tin object**, once again create a **Null SOP** and a **SHOP Network**. **Activate** the **Display and Render Flags** for the **Null SOP**.



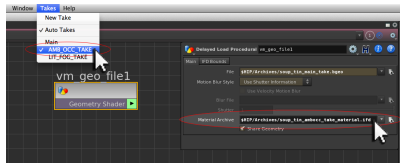


Inside the **SHOP Network**, create a **Delayed Load Procedural Shader**. Ensure that the correct **Archive files** are loaded for the **Main Take**.

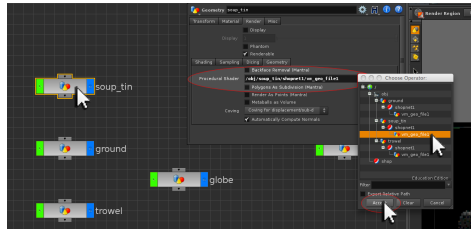
Set the Take to the **LIT_FOG_TAKE** and activate **Auto Takes**. As before trade out the **Material Archive parameter** for the **soup_tin_litfog_take_material.ifd file**.



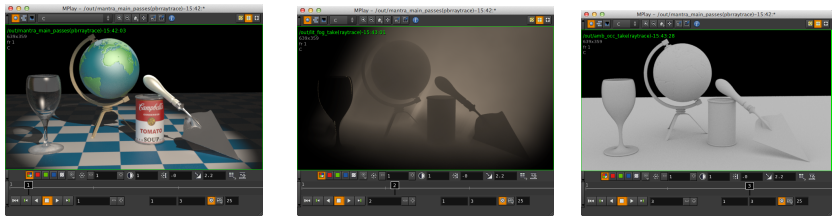
Repeat this process for the **AMB_OCC_TAKE**, replacing the **Material Archive parameter** for the trowel for the **soup_tin_ambocc_take_material.ifd** file.



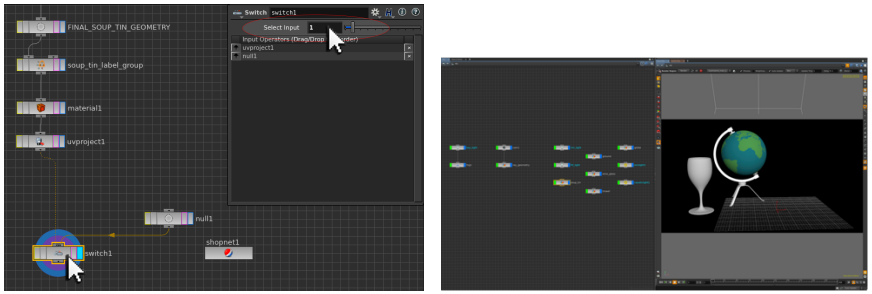
Set Houdini back to the **Main Take**, and at **OBJ Level** go to the **Render > Geometry** section of the **Parameters** for the **soup_tin object**. Here load in the soup_tin Delayed Load Procedural shader into the Procedural Shader parameter.



Test renders of the scene can be produced to ensure the Delayed Load mechanism is working as expected.



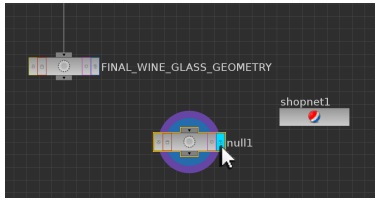
When the **Delayed Load setup** for the **soup_tin object** has been **verified**; it's a **Switch SOP** can be created, with the **main geometry network** as the **first input**, and the **Null SOP** as the **second input**. In the **parameters** for the **Switch SOP** set a **Select Input** value of **1** to explicitly **call the Null SOP** instead of the main geometry network.



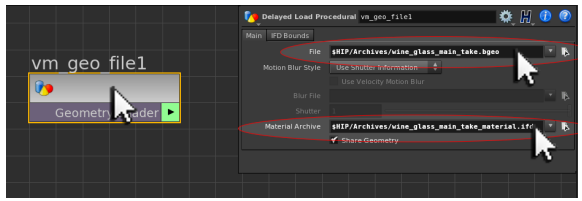
See file globe_scene_optimisation_stage6.hipnc

CREATING A DELAYED LOAD FOR THE WINE GLASS

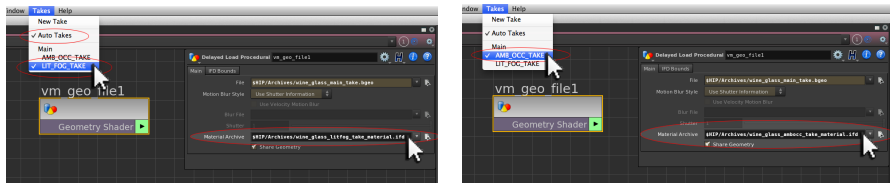
Switch to the **Geometry Level** of the **wine_glass object**. Once again, create a **Null SOP** and a **SHOP Network**. **Activate** the **Display and Render Flags** for the **Null SOP**.



Inside the **SHOP Network**, create a **Delayed Load Procedural Shader**. Ensure that the correct **Archive files** are loaded for the **Main Take**.

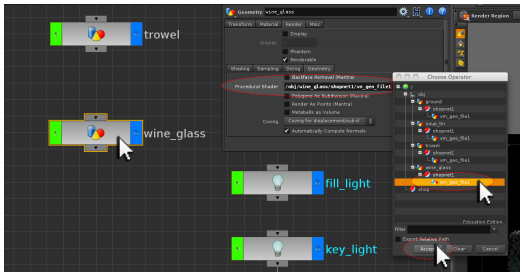


Activate the **LIT_FOG_TAKE** and **Auto Takes**. As before trade out the Material Archive parameter for the **wine_glass_litfog_take_material.ifd file**.

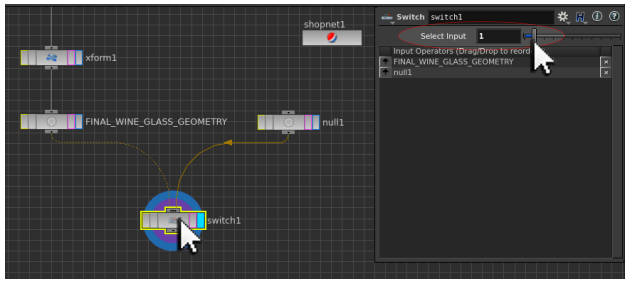


Repeat this process for the **AMB_OCC_TAKE**, replacing the **Material Archive parameter** for the trowel for the **wine_glass_ambocc_take_material.ifd** file.

Set Houdini back to the **Main Take**, and at **OBJ Level** go to the **Render > Geometry** section of the **Parameters** for the **wine_glass object**. Here load in the wine_glass Delayed Load Procedural shader into the Procedural Shader parameter.



Test renders of the scene can be produced to ensure the Delayed Load mechanism is working as expected.

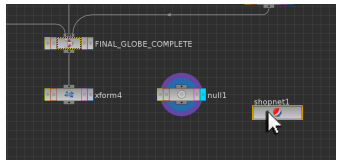


When the **Delayed Load setup** for the **soup_tin object** has been **verified**; it's a **Switch SOP** can be created, with the **main geometry network** as the **first input**, and the **Null SOP** as the **second input**. In the **parameters** for the **Switch SOP** set a **Select Input** value of **1** to explicitly **call the Null SOP** instead of the main geometry network.

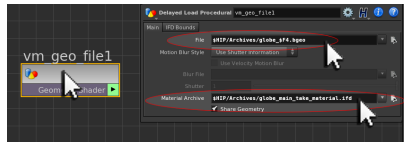
See file globe_scene_optimisation_stage7.hipnc

CREATING A DELAYED LOAD FOR THE GLOBE

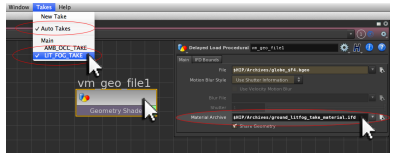
Inside the **Geometry Level** of the **globe object**, create a **Null SOP** and a **SHOP Network**. **Activate** the **Display and Render Flags** for the **Null SOP**.

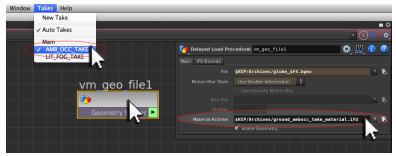


Inside the **SHOP Network**, create a **Delayed Load Procedural Shader**. Ensure that the correct **Archive files** are loaded for the **Main Take**. Remember to load in the **animated .bgeo $F4 sequence** for the globe geometry.



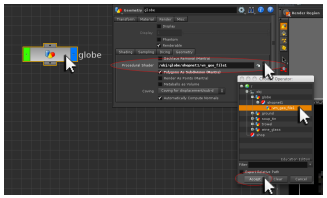
Set the **Take** to the **LIT_FOG_TAKE** and activate **Auto Takes**. As before trade out the **Material Archive parameter** for the **globe_litfog_take_material.ifd file**.

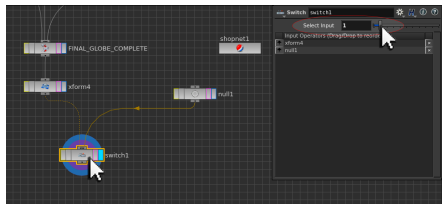




Repeat this process for the **AMB_OCC_TAKE**, replacing the **Material Archive parameter** for the globe object for the **globe_ambocc_take_material.ifd** file.

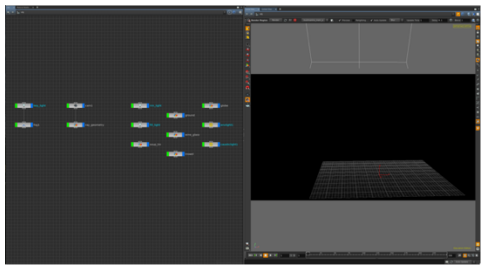
Set Houdini back to the **Main Take**, and at **OBJ Level** go to the **Render > Geometry** section of the **Parameters** for the **globe object**. Here **load** in the globe **Delayed Load Procedural shader** into the Procedural Shader parameter. **Test renders** of the scene can be produced to ensure the Delayed Load mechanism is working as expected.





When the **Delayed Load setup** for the **globe object** has been **verified**; it's a **Switch SOP** can be created, with the **main geometry network** as the **first input**, and the **Null SOP** as the **second input**. In the **parameters** for the **Switch SOP** set a **Select Input** value of **1** to explicitly **call the Null SOP** instead of the main geometry network.

The **final effect** of the **Delayed Load Rendering Setup** is a **completely empty scene file** devoid of any geometry level construction networks, **with only lights and cameras remaining**.

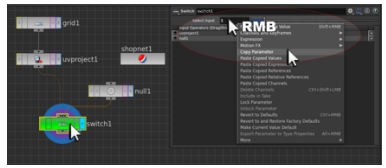


See file globe_scene_optimisation_stage8.hipnc

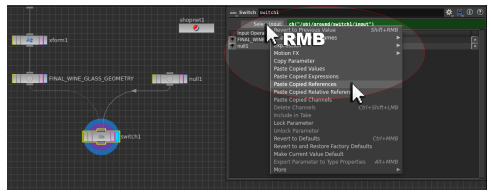
GLOBALLY CONTROLLING THE DELAYED LOAD SWITCHES AND MECHANISM

Now that all scene objects have been activated as Delayed Load Objects, it can be **useful** to **globally control** both the **Delayed Load Mechanism** as well as **the internal Switch SOPs** of each network. Doing this can make the **restoration of a scene to live networks** (Non-Delayed Load) **quick and efficient if Archives** need to be **regenerated** (for example if an **Archive Mantra** is **incorrectly configured** or if the **$HIP location changes**).

To **configure** the **Switch SOPs**, go inside the **ground object** and **RMB** on the **Select Input parameter** of the Switch SOP and choose **Copy Parameter**.



The **Switch SOP node** can also be **coloured** to help **indicate** that it is the **primary switch controller** for the scene.

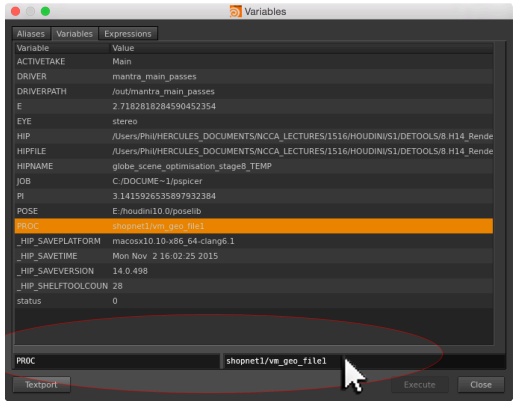


Next go inside **wine_glass object**, and **RMB** on the **Switch SOP's Select Input parameter** and **choose Paste Copied References** to create a **Channel Reference** to the **main switch**.

Continue Pasting Copied References into the **Switch SOPs** for the **globe**, **soup_tin**, and **trowel**.

As the **Delayed Load Mechanism** is **also controlled** at **Object Level** inside the **Render > Geometry** section of the **parameters**, the **node path reference** of the **Procedural Shader parameter** can also be controlled in a similar way.

From the **main Edit Menu**, choose **Aliases and Variables**…



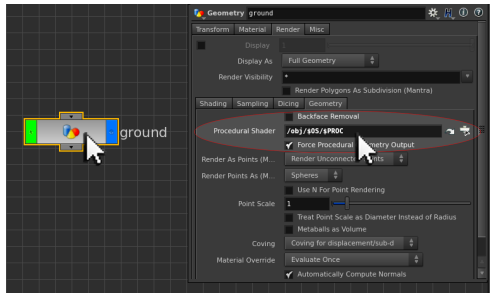
Go to the **Variables** section, and **create** the following **new variable** and **value** in the **text entry fields** at the bottom of the window:

PROC **shopnet1/vm_geo_file1**

When **ENTER** is pressed, this **variable** will be **recorded** in the **Global Variables list**, meaning its value can be **called anywhere** in the Houdini environment **using $PROC**.

Under the **Render > Geometry parameters** for the **ground object**, modify the **Procedural Shader** parameter to read:

Procedural Shader **/obj/$OS/$PROC**

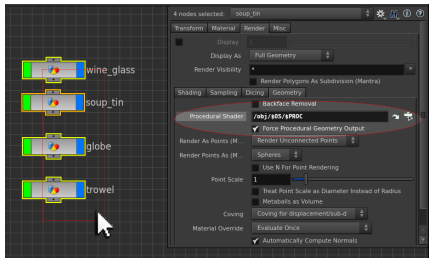


This **variable path** will still **return** the **full path** of the **Delayed Load Procedural shader**, as **$OS** will return the **name** of the **current node** (in this case '**ground**') and **$PROC** will return the **name** and **location** of its internal shader ('**shopnet1/vm_geo_file1**').

This **variable path** can now be **assigned** to **all** the other **scene objects** meaning that they can now be **controlled** by the **$PROC** variable **definition** found in the **Aliases and Variables** window.

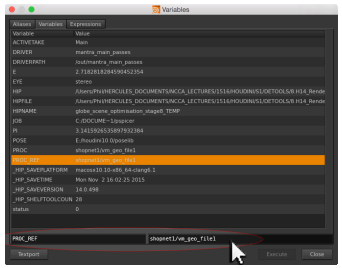
Select all the **remaining scene objects**, and in their **combined parameters** specify in the **Render > Geometry** section:

Procedural Shader **/obj/$OS/$PROC**



Return back to the Edit > **Aliases and Variables window** and create a second variable:

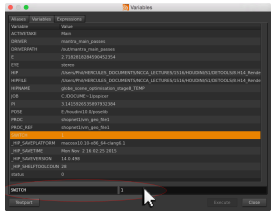
PROC_REF **shopnet1/vm_geo_file1**

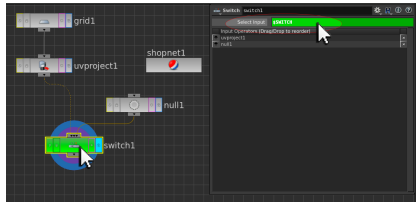


This will store a **copy** of the **$PROC variable** as **a reference**, meaning that the **$PROC variable** can now be **emptied** and **restored easily** as a way to **switch off** and **on** the **Delayed Load Rendering mechanism.**

In the **Aliases and Variables** window, create a **new variable** to control the **ground's Switch SOP**:

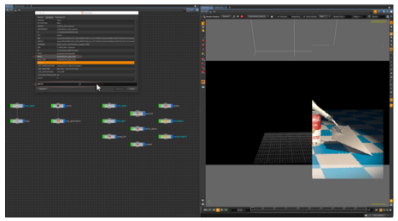
SWITCH **1**

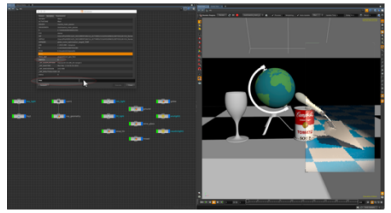




This **$SWITCH variable** can then be **set** in the **Select Input parameter** of the **ground object's Switch SOP**.

Now the **Delayed Load Mechanism** can be **fully activated** and **deactivated** using the **Aliases and Variables window**.





When **$PROC** is set to **shopnet1/vm_geo_file1** and **$SWITCH** is set to **1**, the **Delayed Load Render Mechanism** is **active**, and the **empty scene** will be **optimized for rendering**. When **$PROC** is set to **<empty>** and **$SWITCH** is set to **0**, the **Delayed Load Render Mechanism** is **deactivated**, and the **live scene** can be **modified and tweaked** before final rendering commences.

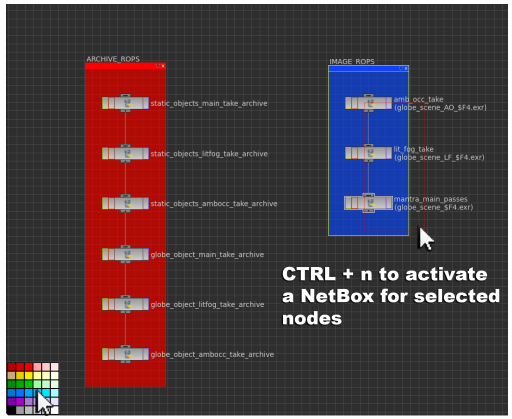
NOTE: Any **further modification to** either **geometry networks** or **materials** will result in the **Archives needing** to be **re-generated** before final rendering commences.

This also means that the **scene file directory** is **portable**; **allowing** for the **regeneration** of **Archive files** when the **live scene** is **active** in the **new scene file directory location**.

See file globe_scene_optimisation_stage9.hipnc

FINAL STEPS

When all of the **Delayed Load Rendering setup** is complete, the **Mantra ROPs** can be selected and stored in a **NetBox** (**CTRL + n** with nodes selected).



NetBoxes can also be coloured in a similar way to **colouring nodes** (**c** with the mouse over the **Network Editor** to activate the node/netbox colour swatch).

archive_script.hsc

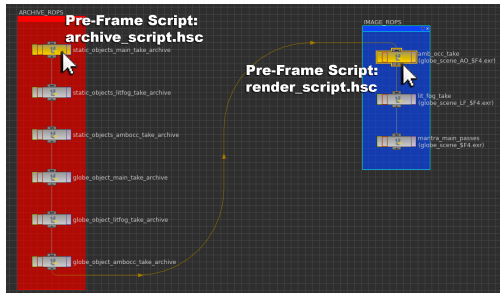
```
#change the $SWITCH VARIABLE TO 0
set -g SWITCH = 0

#change the $PROC VARIABLE TO EMPTY
set -g PROC = " "

#FORCE THE VARIABLES TO UPDATE
varchange
```

render_script.hsc

```
#change the $SWITCH VARIABLE TO 1
set -g SWITCH = 1

#change the $PROC VARIABLE TO THE SHADER PATH
set -g PROC = "shopnet1/vm_geo_file1"

#FORCE THE VARIABLES TO UPDATE
varchange
```

As with the **wobbly_sphere example**, simple **archive_script.hsc** and **render_script.hsc** HScripts can be developed and **saved in $HIP** to **control** the **$SWITCH** and **$PROC variables** relative to whether Archives or Images are being generated.

These should be **assigned** in the **Pre-Frame Script parameters** of the **top node** of **each mantra chain**. These **nodes** can be **coloured** to **indicate** that **scripts** have been **assigned**.



The **end node** of the **Archive ROPs chain** can then also be **wired** into the **top node** of the **Image ROPs chain**. This will ensure that **$SWITCH** and **$PROC** are set correctly at the start of each chain's rendering. The **final node** of the **IMAGE_ROPS chain** can also be **coloured green** to help **indicate** that this is the **one to be rendered**.

Channel Referencing can also be **assigned** from the **mantra_main_passes ROP** to the **Globe Archive ROPs** to **synchronize** the **Frame Range** being **rendered**.

When **rendering** is **activated** on the **mantra_main_passes node**, all the **nodes higher up** will be **rendered first**, **starting** with the (**yellow**) **static_objects_main_take_archive ROP**.

See file globe_scene_optimisation_complete.hipnc

IMPORTANT NOTE: It is a good idea to always render Image Sequences to an Empty Directory rather than rendering over existing files. If re-rendering needs to take place, delete or move old renders to an alternate location to ensure the new renders are properly generated (it has been observed for example that overwriting an existing exr file can cause Mantra to stall).

RENDERING OPTIONS

When rendering from Houdini there are a number of options that can be followed:

1. **STRAIGHT RENDER FROM HOUDINI**

   A scene with correctly configured parameters can be outputted directly by pressing the **Render Button** of a **Mantra ROP**. This will render; however there are memory overheads in terms of Houdini being open and running, and all internal networks being processed one node at a time. If for example the construction nodes of a geometry object are still present within the scene, Houdini will build the object again and again for each frame being rendered. If a **straight render** from Houdini **is invoked**, wherever possible **Geometry Objects** should be rendered out **as .bgeo files** and read back in using a **File SOP** before rendering commences. This can be done by RMB on the final node of the geometry and choosing Save Geometry. An **alternate way** to render geometry as **.bgeo sequences (for example a deforming mesh)** is by appending a **ROP Output Driver** to the node network, and specifying a frame range of geometry to render out.

2. **STRAIGHT RENDER FROM HOUDINI USING DELAYED LOAD RENDERING**

   Delayed Load Rendering is where all geometry and materials are rendered out as .bgeo and .ifd archives. The Object itself can then be told to read in these archive files at render time. Delayed Load Rendering can be invoked with Houdini open; however having Houdini open will also factor into the computer's memory usage.

   STRAIGHT RENDERING FROM HOUDINI IS NOT ADVISED AS IT USES UP HOUDINI LICENSES UNNECESSARILY

3. **HRENDER TO RENDER HIPNC FILES**

   A Houdini utility application called **HRender** can be called using the **command Shell**. This will allow all Houdini scenes to be closed, so that all rendering is invoked using the Shell alone. This will help the computer dedicate all resources to the HRender command. Rendering this way also prevents Houdini licenses being used. **NOTE**: **We have many Batch licenses for rendering but only a few Houdini licenses (used for running the Houdini UI).**

   a. **Open up a Shell, a cd (change directory) into the folder containing the hipnc file to be rendered.**
   b. **Type hrender –h in order to display the help card for the hrender command**
   c. **Determine the hipnc file to be rendered, plus the name of the final Mantra ROP in the daisy-chained Mantra Network**
   d. **Invoke hrender by typing the command:**

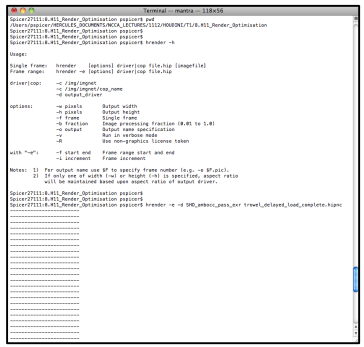
   hrender –e –R –d <name of Mantra ROP> <name of hipnc file>

   for example:
   hrender –e –R –d mantra_main_passes globe_scene_optimisation_complete.hipnc

   e. **Wait for the renders to emerge…**

   By default the **–e flag** will render the **frame range specified** on the Mantra ROPs. This can be overridden if necessary (see hrender –h for more information).

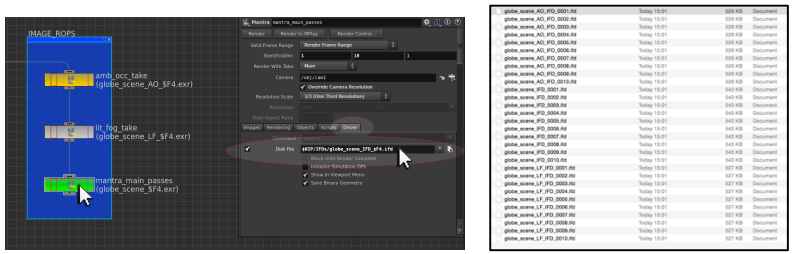
   The **–R** flag will ensure that a **Batch Render license is used** rather than a Houdini license.



NOTE: If a **daisy-chained Mantra node** is **invoked for rendering** using the **Terminal** and **hrender**; this behaviour differs slightly where all the frames for the first node in the chain will be created, before hrender then moves onto the second node in the chain.

4. IFD RENDERING

An alternative to using **hrender** is to **render** in a **Terminal Shell** using the **mantra** command instead. Before invoking a **mantra terminal render**, **all Image Mantra ROPs** must have the **Driver > Disk File option activated**, with a **$HIP relative path** to **store** the resulting **.ifd files**.



When either the Mantra ROPs are **rendered directly** or by using **hrender**; rather than images being generated, **ifd text files** are generated instead. As with **material .ifd files**, **$HIP path listings** are **converted into full paths**, meaning that the **location of the $HIP directory** has to be **correct before** these **.ifd files** are generated.

These .ifd text files contain a per frame script for creating the final rendered image. On the **NCCA Linux Systems** a simple **Shell Script** can be developed to **render a sequence** of **.ifd files** (see **mrender** script):

```
#! /bin/bash
# mantra render script

# define start number of ifd sequence
i="1"

# define end number of ifd sequence
n="10"

# define prepad-number name of ifd sequence to be rendered
name="wobbly_"

while [ $i -le $n ]
do

pad=`exec ~istephen/bin/padnumber $i 4`
mantra $name$pad".ifd"

i=$[$i+1]
done
```

See file globe_scene_optimistation_complete_IFD.hipnc

THE NCCA RENDER FARM

A formal **demonstration** of using **Houdini** (both **hrender** and **mantra ifd rendering**) on the **NCCA Render Farm** will take place shortly.

MEASURING RENDER TIMES

Which ever Rendering Option is chosen, **Rendering Times should be measured**. Key frames from an image sequence (including any render passes) can be rendered out to disk and the time taken can measured accordingly. This is the best way to determine how long a series of renders will take.

With **Houdini open**, **rendering times** can be **measured** using the **OUTPUTS Level Context Viewer**.