HOUDINI FOUNDATIONS

MODEL, RENDER, ANIMATE

Welcome to Houdini. In this lesson you will start from scratch to model, render, animate, and simulate a soccer ball (also known as a football in many parts of the world). You will create a classic bouncing ball animation using the principles of squash and stretch, apply textures and materials, add lights and cameras, and explore the use of dynamics to simulate a group of soccer balls.

These tasks will introduce you to many different parts of Houdini as you create your first Houdini scene, explore the interface and discover some of its most important tools. You will learn how to work interactively in the **Scene View** and how to use the **Network View** to manage your nodes as you refine your model and build your animation rig. You will also set up materials and textures on the **Solaris Stage** then you will render using Houdini's built-in renderer **Karma,** and finally create a **Rigid Body Simulation**.

LESSON GOAL

Model, Render and Animate a soccer ball using Houdini's procedural node-based workflow

WHAT YOU WILL LEARN

- How to work with the **View Tools**
- How to use **Shelves**, **Radial Menus** and the **Tab** key
- How to create **Geometry**
- How to work with **Nodes and Networks**
- How to set up **Custom Attributes** and a **For-Each Loop**
- How to set up **Materials** and **Texture UVs**
- How to **Layout** a shot and render with **Karma**
- How to **Set Keyframes** and add **Motion FX**
- How to use **Rigid Body Dynamics**

LESSON COMPATIBILITY

Written for the features in Houdini 18.0.378+

The steps in this lesson can be completed using the following Houdini Products:

Houdini Core	✔
Houdini FX	✔
Houdini Indie	✔
Houdini Apprentice	✔
Houdini Education	✔

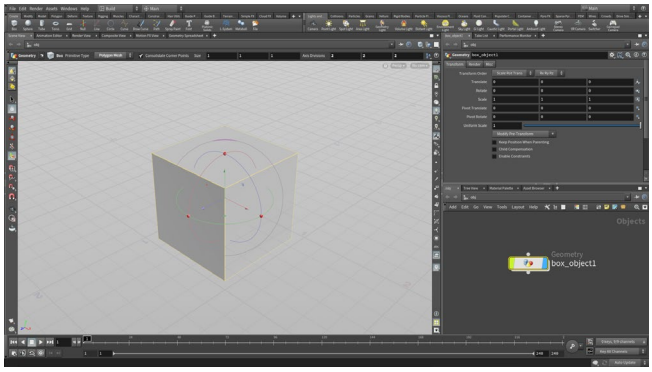
Document Version 2.01 | April 2020

PART ONE: Explore the Houdini UI

To get started, it is important to learn how to work with the Houdini workspace and the three panes you will use the most. The **Viewport** lets you create objects interactively, the **Parameter Pane** lets you edit node properties and the **Network Editor** lets you work directly with the node networks.

PROJECT FILES

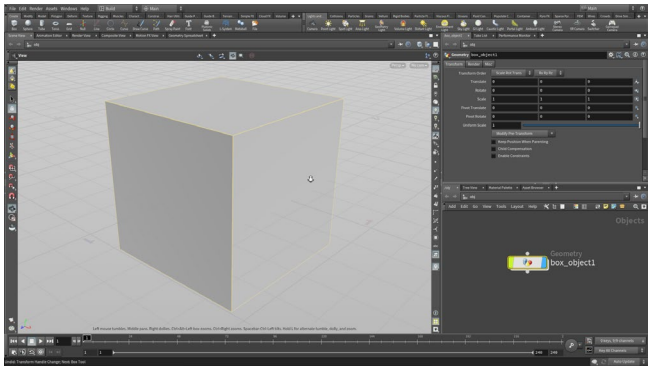
Go to the **soccerball tutorial** page on **SideFX.com**, where you likely got this document, to download the ***intro_lesson*** directory. Put it into the **Houdini Projects** directory which you can find in either the **home** directory or the **documents** directory.



01

In the viewport, **press c** to bring up a radial menu. From this menu, choose **Create > Geometry > Box**. Your cursor now shows the outline of a box waiting to be placed in the scene. **Press Enter** to place it at the origin.

This creates a box in the Scene view, adds a node in the Network editor and shows the object parameters in the Parameter pane. As you work through this project, you will touch on all of these interface elements.



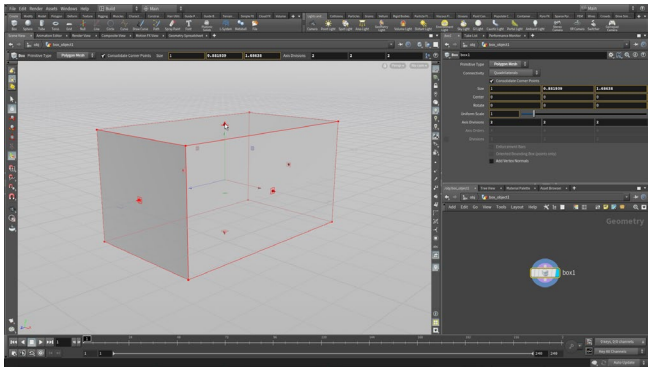
02

You can now explore the **View** tool in Houdini. Press the following hotkeys:

- **Tumble** — Spacebar or Alt[Opt] - LMB click-drag
- **Pan** — Spacebar or Alt[Opt] - MMB click-drag
- **Dolly** — Spacebar or Alt[Opt] - RMB click-drag

In some cases, you will want to home in to get your bearings. There are some hotkeys for that as well:

- **Home Grid** — Spacebar -H
- **Home All** — Spacebar - A
- **Home Selected** — Spacebar - G



03

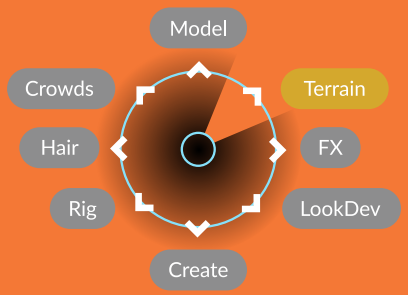
With the object selected, **press i** to go to its geometry level. Use the **Shift** key to drag on handles to make it longer along z axis around the origin.

When an object is created in Houdini, there is an **Object level** which is where you manage the transformations of the object and a **Geometry level** where you define its shape. Pressing **i** brought you down into the geometry level of this object. You can also get there by double-clicking on the object node in the Network editor. Later, to get back to the Object level, you can **press U**.

RADIAL MENUS

One way to access tools in Houdini is radial menus which you can access using the **X**, **C** and **V** hotkeys. Each of these brings up a radial menu with lots of options for you to choose from. The main focus of each menu is as follows:

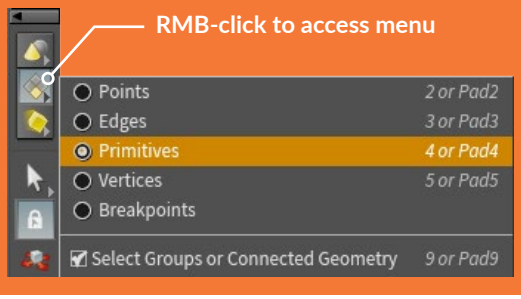
Snapping	**X**
Main (or Custom)	**C**
View	**V**

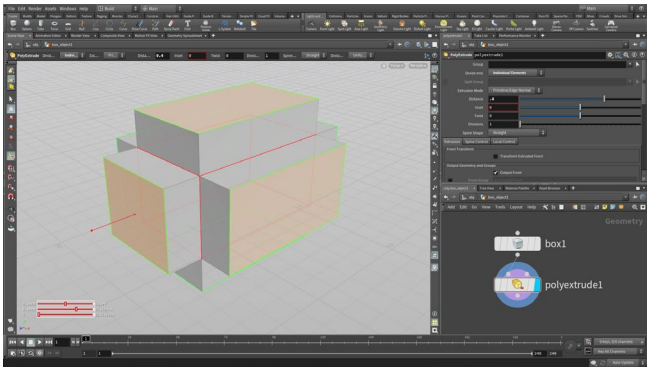


SELECTION HOTKEYS

If you are using the **Select, Move, Rotate, Scale** or **Handles** tools, the following hotkeys will determine your selection mode as well as which level you will be working at.

Objects	Object Level	1
Points	Geometry Level	2
Edges	Geometry Level	3
Primitives/Faces	Geometry Level	4
Vertices	Geometry Level	5

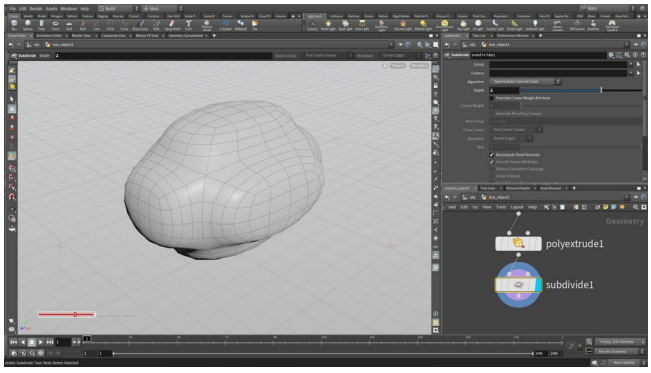




04 **Press S** to go to the **Select tool** then **4** to access Primitive selection. **Press n** to select all then **press c** to bring up the radial menu and choose **Model > Polygons > Poly Extrude**.

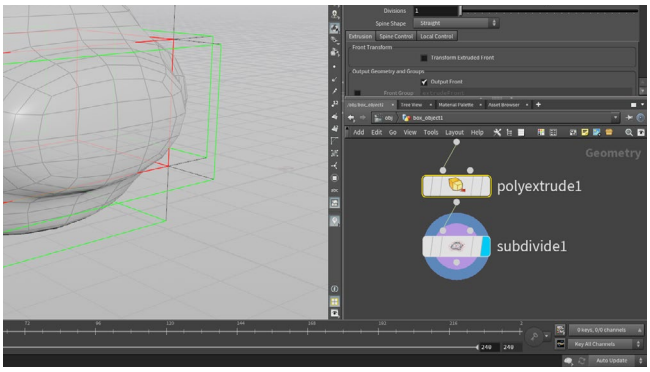
In the Parameter pane, set **Divide Into** to **Individual Elements** and use the handle to set the **Distance** to around **0.4**. This extrudes all the faces of the box along the normals of each primitive.

You can see that there are now two nodes in the Network view. Each step you take in Houdini creates a node that you can work with to refine your scene.



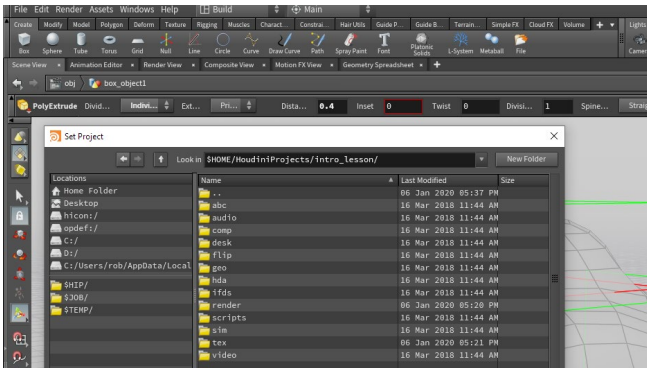
05 Press **n** to select all of the new faces and press **Tab** and begin typing **sub...** then select **Subdivide** from the list. The Tab key is another way to access tools in Houdini. Typing the tool name lets you focus the list making it easier to find what you want without navigating the submenus.

In the Parameter pane, set **Depth** to **2**. This subdivides the geometry to create more polygons. Houdini also has a subdivision display option at the Object level which you can use to see subdivisions without actually adding any geometry, but in this case we do want to create more polygons.



06 Select the different nodes in the chain. The handles for each of the nodes appear as you select them but the display remains on the final shape. Set the **Display Flag** on each of the nodes to change which node is the display node. You can also try some of the other flags such as **Bypass** or **Template**. **Wiggle** the *polyextrude* node out of the network then drop it back in.

At the end, return everything to normal and set the **Display flag** on the *subdivide* node. This is very important. The Display flag determines what you will see at the object level. **Always check to make sure you have the right display flag set!**

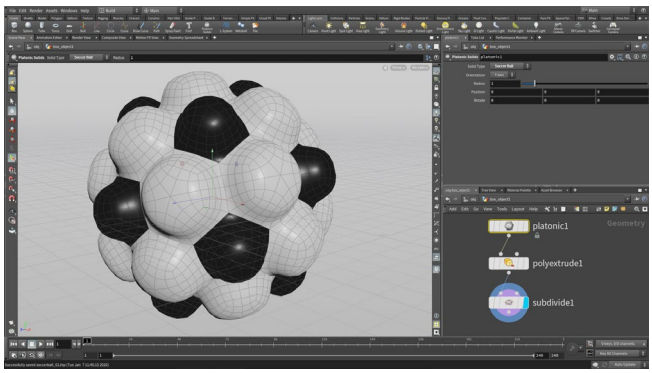


07 Select **File > Set Project.** Find the *intro_lesson* directory that you downloaded earlier and press **Accept**. This makes this project directory and its sub folders the place for all the files associated with this shot.

Select **File > Save As...** You should be looking into the new *intro_lesson* directory. Set the file name to *soccerball_01.hip* (or *football_01.hip* if you would prefer) and click **Accept** to save.

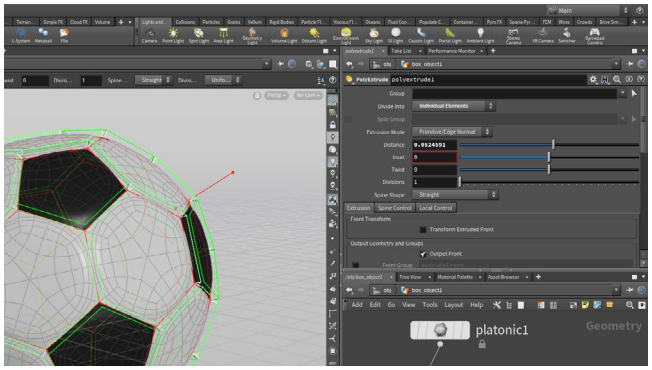
PART TWO:
Adding the Soccerball Geometry

You are now going to replace the box with a soccerball shaped platonic shape. Using Houdini's procedural approach, you can replace the box node with the new geometry. From there you will adjust the Houdini nodes to make it look like a simple soccerball. This ability to swap out input nodes lets you prototype networks with simple geometry for added flexibility.



01 In the **Network editor**, use the **Tab** key to add a **Platonic solids** node to the network. Click to place it down near the top of the chain. Wire the *platonic* node into the *polyextrude* node. In the parameter pane, set **Solid Type** to **Soccer Ball**. Select and delete the *box* node.

Because of Houdini's procedural nature, it is often possible to replace an input node and have the whole network function properly. This gives you flexibility as you work and if you don't like the results after the change then you can always wire back the original shape.



02 Select the *polyextrude* node and use the handle in the viewport to set a smaller **Distance**. You can also set the parameter value in the Parameter pane. This creates a better look for the soccerball. Remember that even though we are viewing the *subdivide* node, selecting the *polyextrude* node gives us access to its handles and parameters.

You might think that with this primitive type we are all set but it is really just a truncated icosahedron with flat faces. We need a round soccerball so we will have to put a little more work into it.

03 Press **V** in the viewport and from the **Radial Menu,** select **Shading > Smooth Shaded**. You can also use the menu in the top right of the viewport to change your shading.

This soccerball looks like a cheap plastic ball rather than a proper leathery soccerball. You are now going to branch off and add more nodes to get a better look.

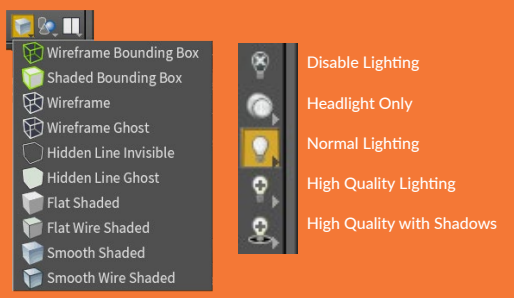
After analyzing it, set the shading back to **Smooth Wire Shaded**.

SHADING OPTIONS

There are a number of **Shading Options** available from either the **View** radial menu or the **Shading** menu in the top right of the Viewport.

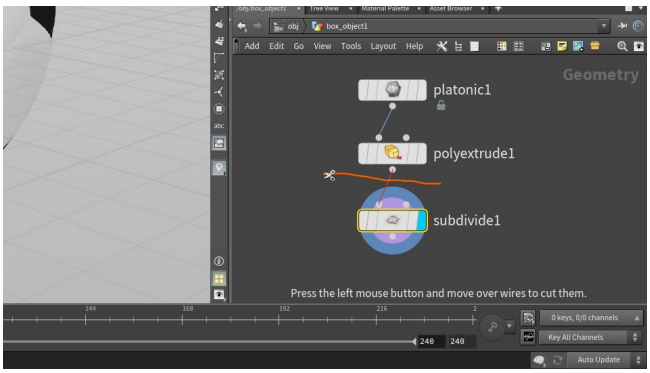
For the shading of your objects, the lighting is determined by the **Display Options** on the right edge of the Viewport. You can choose from a headlight, normal lighting or high quality lighting with shadows.

To quickly toggle from your shaded view to wireframe press the **W** key.

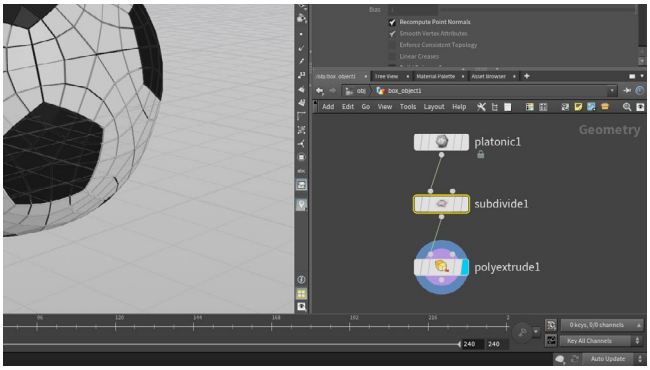


PART THREE:
Create a Realistic Soccerball

There are many ways to create geometry using Houdini's nodes and networks. In some cases, you may want to swap the order of two nodes or rewire a network to try something different. You are now going to add more detail to your soccerball by creating a different network that subdivides before extruding.

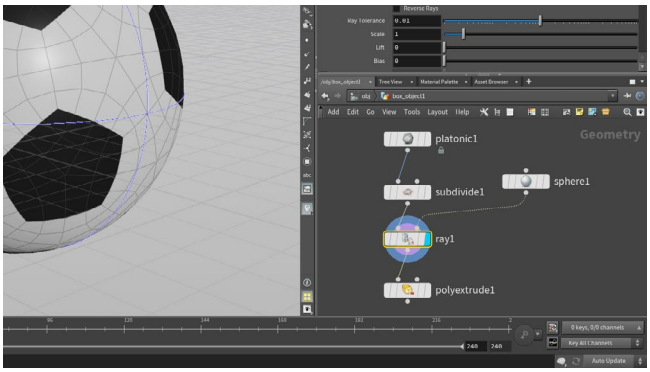


01 In the Network view, **press Y** and drag across the line connecting the *subdivide* node and the *polyextrude* node to break the connection. You are now going to move the subdivide inbetween the other two nodes so that we get a rounder soccerball.



02 **Drag** the *subdivide* node in between the *platonic* solid node and the *polyextrude* node. You can drop it on the connecting wire and it will insert itself in automatically. If not then jiggle it a little until it finds the connection. This will give more detail to the sphere before it is extruded.

If you set the display on the *polyextrude* node you will see that it doesn't look right because we have lots of small polygons being extruded but you will fix that in the following steps.



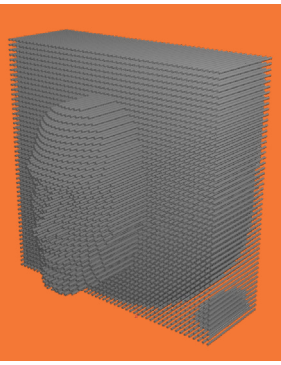
03 Use the **tab key** in the network editor to add a **Ray** node and wire it in after the *subdivide*. Now add a **sphere** node to the network and wire it into the second input on the *ray* node. This will project the subdivided ball onto a perfect sphere.

This is a very powerful node in Houdini that lets you project points from one piece of geometry onto another. It is the perfect solution to our problem of a subdivided soccerball that wasn't truly round.

THE RAY NODE

The ray node is a tool that projects points out to another piece of geometry. This is similar to the pinboard toy you played with as a kid. In fact, this is the node you would use to set up a pinboard in Houdini.

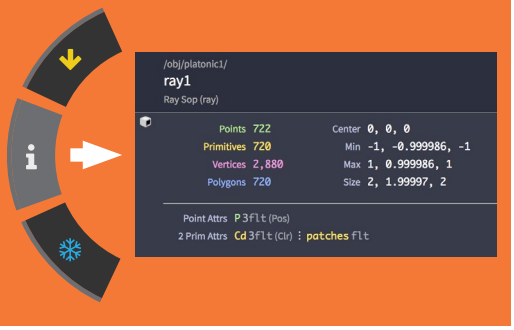
GETTING HELP | To learn more about each node, you can click on the help button in the top right of the Parameter pane to open up the node's online documentation. You can also hover over the tool in the shelf and **press F1**. In many cases, there are sample files that you can open in Houdini to learn what the node can do.

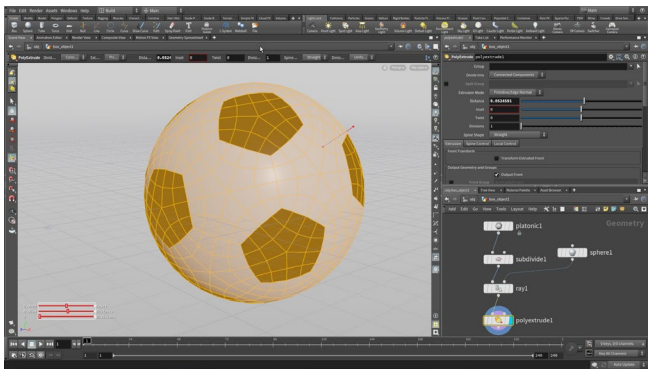


WORKING WITH ATTRIBUTES

Attributes can be assigned to Points, Primitives or Vertices. Some typical types of attributes include **color (Cd)** or **UVs**. You can see the attributes at any point in your chain by mousing over a node and choosing the **i** from the radial menu. You can also review the attribute values in the **Geometry Spreadsheet** panel.

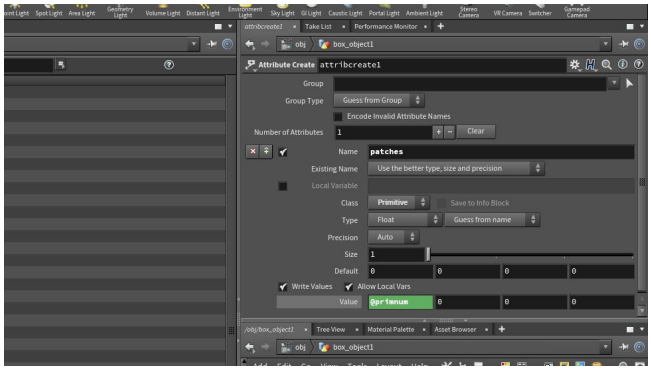
In this lesson, you will be creating a custom attribute called ***patches*** which will help you later on in the network.





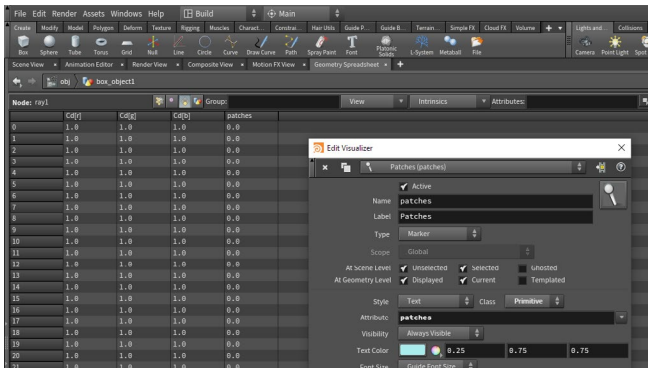
04 Set the **Display flag** on the *polyextrude* node. With **Divide into** set to **Individual elements** all the small polygons are extruded but we don't want that. But if we change it to **Connected Components** then all the polygons are extruded which is also not what we want yet.

Set it to **Connected Components**. We need a way for this network to let us extrude the original patches of the soccerball but after the ball has been subdivided. We can do this using the primitive numbers on the original geometry.



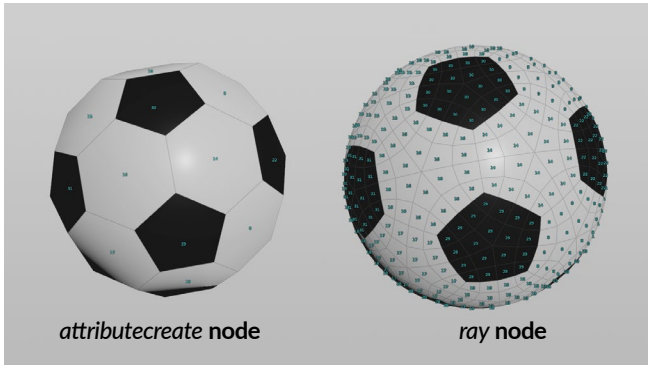
05 Add an **AttributeCreate** node after the *platonic* node. – Set its **Name** to *patches* and its **Class** to **Primitive.** Now set the first of the **Value** fields to **@primnum**. This expression takes the primitive number attribute and turns it into a new attribute called *patches*.

With the *attributecreate* node selected, click on the **Geometry Spreadsheet** tab next to the main viewport. Click on the **Primitive** button and you can see the primitive numbers on the left, three color attributes which show the color of the patches and the *patches* attribute which matches the primitive numbers.



06 Now click on the *ray* node. This attribute will be carried forward when the shape is subdivided. You can now see there are a lot more primitives but the *patches* attribute only goes as high as 31 and then it goes back to 0.

Go back to the **Scene View** tab, **RMB-click** on the **Visualizer display** button on the **Display Options** bar and click on the **+ Plus sign** next to **Scene** and choose **Marker.** In the Edit Visualizer panel, set **Name** and **Label** to *Patch_Numbers*, set **Type** to **Marker**, **Class** to **Primitive** and **Attribute** to *patches*.



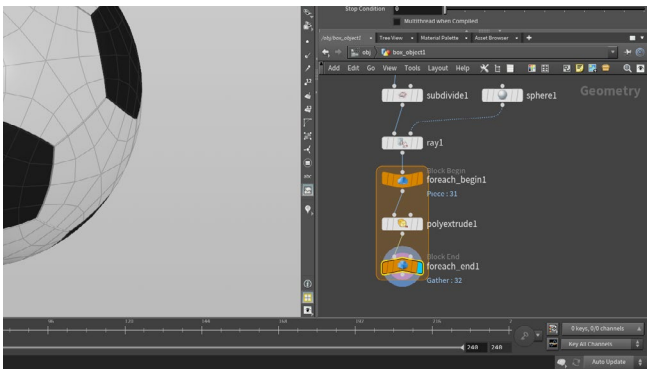
attributecreate **node** *ray* **node**

07 Now make sure the **Visualizer display** button is on and you will see the patch values on the soccerball in the viewport. You can see that the prim number from the original platonic solid have been transferred to the subdivided faces. Change the **display flag** to different nodes to see the relationship. This information will be used to polyextrude the patches properly using a for-each loop.

Turn **OFF** the Visualizer display and **save** your work. These steps have been a bit abstract and probably feel a bit too technical, but don't worry the payoff is coming.

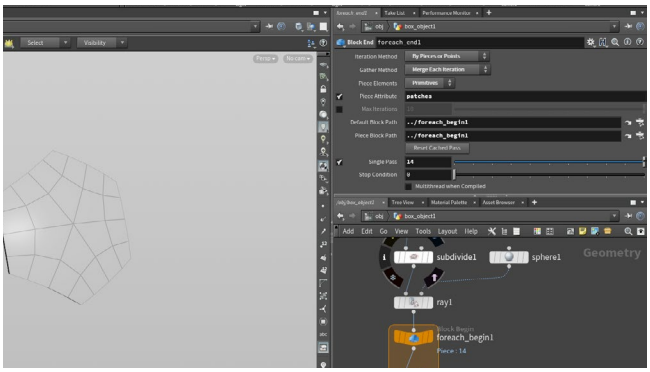
PART FOUR:
The For-Each Node

Now you get to see the magic, as the attributes you just created in the last part are fed into a for-each loop where the original patches are extruded even though each contains many polygons. This will provide a more leathery look for the soccerball once you subdivide it once more time after the poly extrudes.



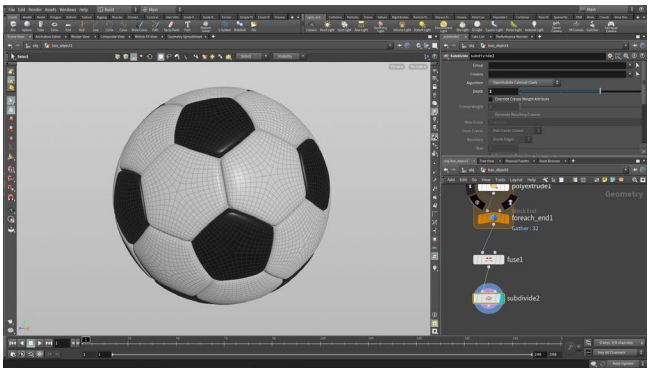
01 In the Network editor, press **tab** and start typing **For-each Named Primitive** to access two nodes that you can then place into the scene. Wire the *foreach_begin* between the *ray* node and the *polyextrude* node and then the *foreach_end* after the *polyextrude* node. Select the *foreach_end* node and in the parameter pane leave **Piece Elements** set to **Primitives** and set **Piece Attribute** to *patches*. Set display on the *foreach_end* node.

You should now see the original patches being extruded together based on the *patches* attribute. If they are not make sure **Divide to** is set to **Connected Components** on the *polyextrude* node.



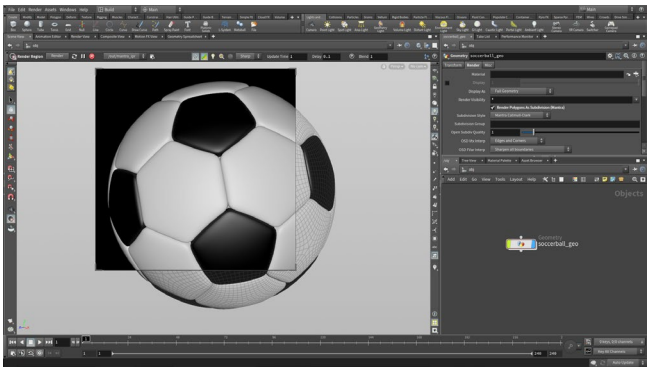
02 Click the checkbox next to **Single Pass** to explore what is happening. Drag on the slider to watch as each of the patches is polyextruded individually. You can also set values higher than 10 to see more of the patches.

Turn off **Single Pass** to see the full shape. The *for-each* nodes create all of the patches then return the final geometry. The for-each loop is a powerful set of nodes that you will use often with Houdini.



03 Add a **Fuse** node after the *foreach_end* node and set its **Display flag**. This connects the pieces into a single topology. The *for-each* nodes broke them into the different patches but didn't fuse them back together.

Add a **Subdivide** node after the Fuse. Set the **depth** to **2**. This will give you more detail in the viewport that you can use to evaluate your model. This adds more polygons but will not yet render as a true subdivision surface. Houdini also lets you set Subdivision display in the viewport without adding geometry but this Subdivide node is needed for later in the lesson.

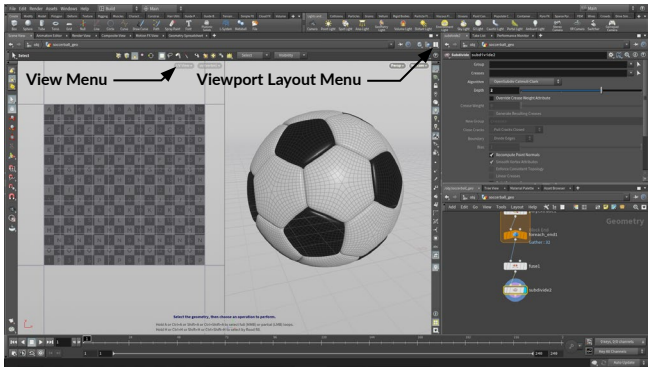


04 Play with *polyextrude* values to get a nice leathery soccer ball. Here we set a **Distance** of **0.1** and an **Inset** of **-0.02**. This gives us nice rounded patches that look much better.

Go to the **Object level** and rename the object *soccerball_geo* in the Parameter pane. Either select the node and press **F2** or **double click** on the name. Click on the **Render** tab and turn on **Render Polygons as Subdivisions (Mantra)** to set up true subdivisions at render time. Select the **Render Region** tool, then draw a box around the soccer ball in the viewport create a preview rendering. To cancel, click on the **x button** in the top right of the region.

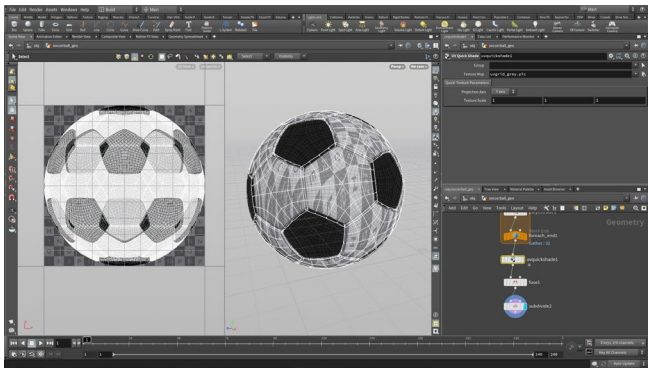
PART FIVE:
Setting up UVs

In order to set up materials and textures, it is important to make sure that there are proper UVs set up on your object. Geometry in Houdini does not come with UVs, therefore you must create them yourself. This means adding extra nodes to the network, which in this case means adding UV Flatten and UV Quickshade nodes.



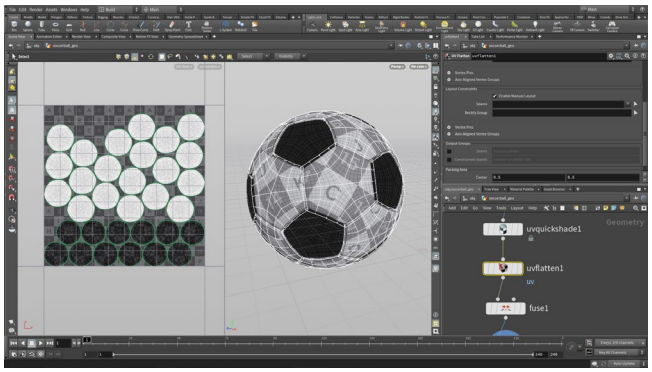
01 In the top right of the Scene View, use the **Viewport Layout menu** to choose the **Two Views Side by Side** option. You could also us the hotkeys **Ctrl - 4** to get these views.

In the left panel, click the **View menu** Top and choose **Set View > UV Viewport** to change this panel to a UV view. You can also hover your mouse over the panel and press **spacebar-5**.



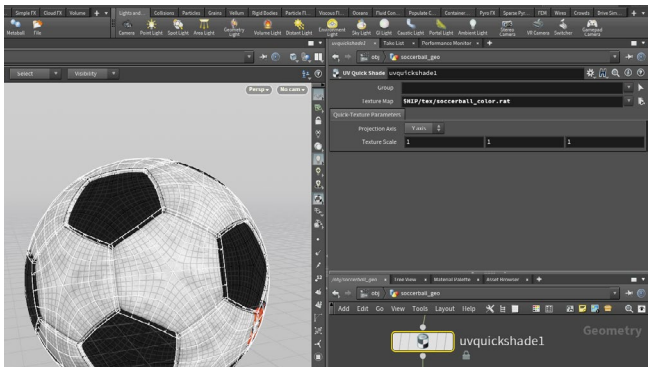
02 To set up UVs for the soccerball, you can choose from a number of different techniques. The fastest is to add a **UV Quickshade** node into the soccerball chain. Place it right after the *foreach_end* node.

Now you will see that this node created UVs by projecting down along the Y axis which is not creating useful results. This is fine because you are only using this node to assign a UV grid for previewing purposes and will replace the UV layout with a better solution in the next step.



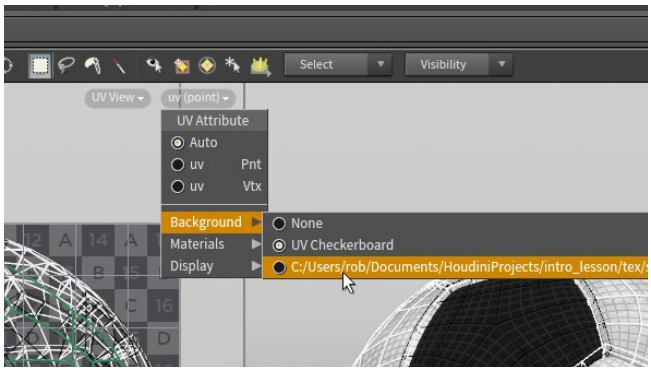
03 Add a **UV Flatten** node after the *uvquickshade* node but before the *Fuse* node. This will flatten the soccerball geometry using the patch boundaries to lay out the UVs. In the perspective view you can see how this affects the look of the uvgrid on the ball.

If we had placed this node after the *fuse* or after the *subdivide* then there wouldn't have been any boundaries to work with. Houdini's ability to let you set up UVs in the middle of the network can be extremely useful when working with different kinds of geometry.



04 Select the *quickshade* node and click on the **File Selector** button next to **Texture Map**. Navigate into the `/tex` folder and select the *soccerball_color.rat* image file. Now you can see this texture map on your geometry. You will use this to help you position the UVs to match the existing texture design.

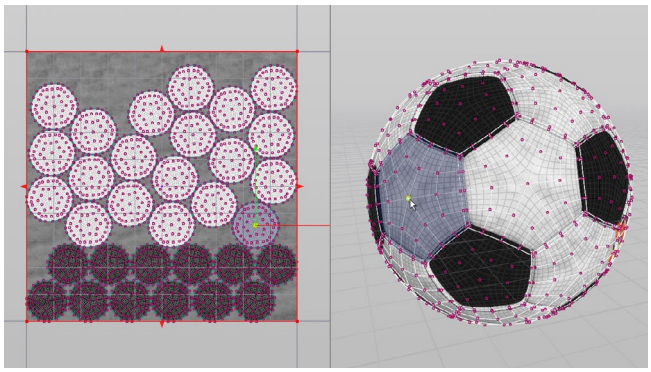
Note: *Setting up UVs to match an existing texture is not the normal order of operations. In practice, the UVs are generated then passed on to a texture artist who builds the texture. We are taking this approach in the tutorial so that you can use an existing texture and don't need to paint it yourself.*



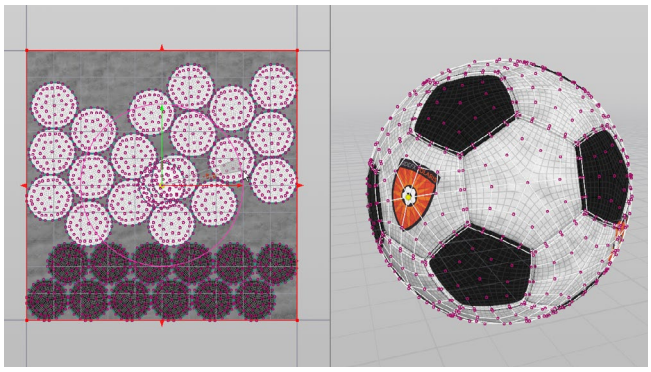
05 In the UV view, click on the UV menu in the top right and choose **Background >** *soccerball_color.rat*. Now you can see this texture in the background of the UV panel which will allow you to make some adjustments to how this view works.

Note: *If you don't see this in the menu, try to recook the nodes by setting the display flag on a diffrent node and back again.*

This texture has a team insignia at the center of the image. You want to line up one of the visible patches to the insignia. To do this you will go back to the **UV Flatten** node and adjust the patches.

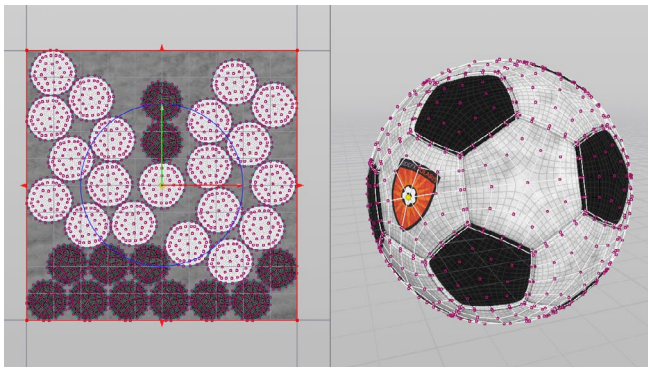


06 Select the *uvflatten* node and click on the **Handle** tool. Mouse over the geometry you can see the patches highlight. Click on the **Pin Vertices** button in the **Operation Controls** bar at the top of the viewport. Mouse over the patch shown in this image and in the 3D view, click on the center vertex of the patch shown to select it. Press **Enter** to create the Pin.



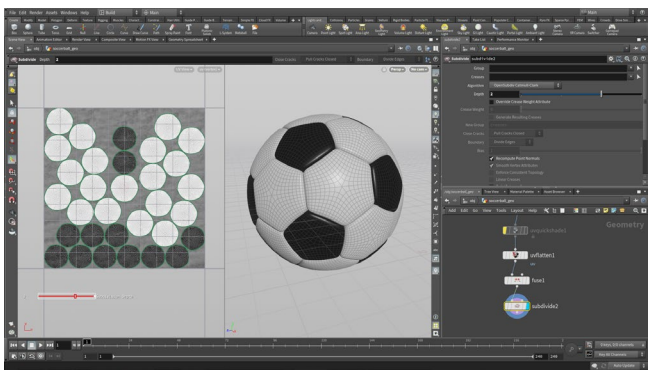
07 Now go to the UV view and **move** the pin and the patch so that it is centered on the logo. **Press Y** to get a rotate handle and rotate the patch until you line up the logo the way you want.

By pinning vertices on patches in the UV view, you can lock down their position. At first there is some overlap of the neighboring patches but you easily can fix this by repacking.



08 Click on the **Repack** button in the Operation Controls bar to reorganize the other patches around the new one. You can continue moving the patch around using the pin but may need another **Repack** if you create overlapping UVs.

Note: *Setting up UVs to match an existing texture is not the normal order of operations. In practice, the UVs are generated then passed on to a texture artist who builds the texture. We are taking this approach in the tutorial so that you can use an existing texture and don't need to paint it yourself.*

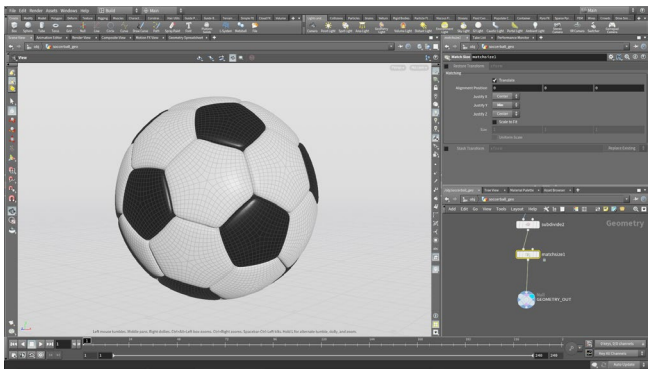


09 Set the *quickshade* node to **Bypass** to hide the assignment of the UV grid. This node was only needed when you were evaluating UVs. It could be deleted but using bypass means that we can turn it back on later if we want to tweak the UVs. The UV Flatten node is now creating the UVs for the model.

The **Bypass flag** is a useful way of blocking the influence of a node to either test out a design idea or to get rid of nodes that aren't needed. You could also delete the node but then you won't be able to go back and retrace your steps if it is needed again.

PART SIX:
Layout: Cameras and Lights

To create a scene for rendering, you are going to bring the geometry into the Solaris or LOPS context of Houdini. This is an environment dedicated to lookdev, layout and lighting and is built on the foundation of USD (Universal Scene Description). This will allow you to render to the Karma renderer which works right in the Scene View as part of the Solaris workflow.

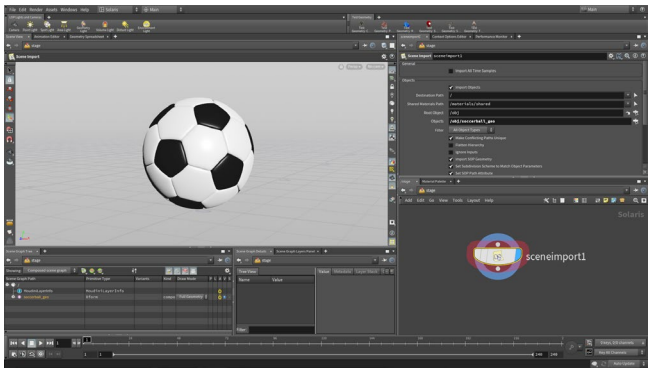


01

Mouse over the perspective view and press **Spacebar-b** to expand that view to just show the soccerball.

In the Network view, **RMB-click** on the *subdivide* node's output and type **Match Size** then press **Shift-Enter** to place the node. Set **Justify Y** to **Min** which will raise the ball up to sit on the ground.

Add a **Null** node to the end of the chain and call it *GEOMETRY_OUT*. It is a good idea to have this kind of node to define the end of the network chain.

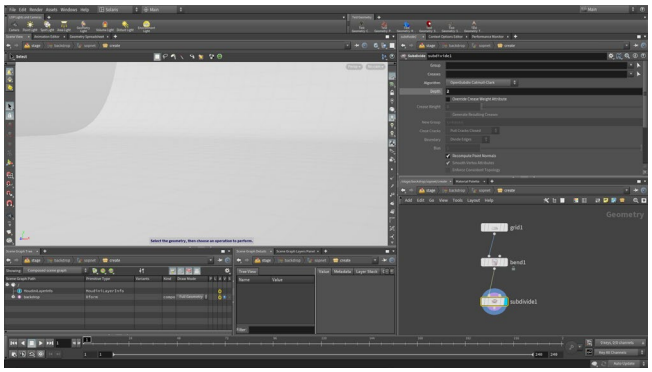


02

Change the desktop to **Solaris**. Choose **Stage** from the path bar.

In the Network view, press **tab > Scene Import** and click to place the node down. Next to the **Objects** field, click on the **node selector** button and from the pop up window, select the *soccerball_geo* object then click **Accept Pattern**.

In the Scene View, use your view tools such as **spacebar-h** for homing the view to get a better look at the soccerball.

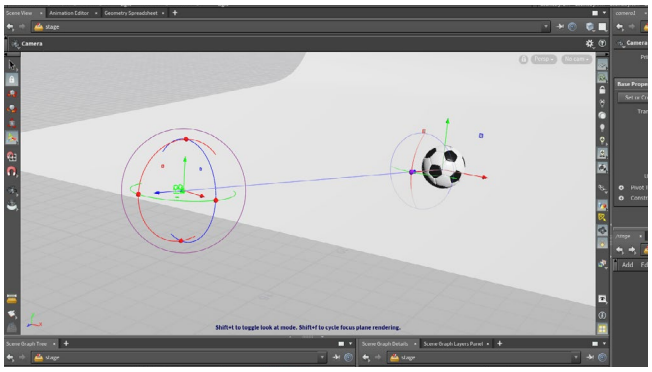


03

In the Network view, press tab and type out Grid. Click to place down the node and rename it *backdrop*. Double-click on the *backdrop* node to dive down to the geometry level.

Select the *Grid* node and set the size to **60, 60** and the **Center** to **0, 0, -20**. **RMB-click** on the *grid* node's output and type **Bend**. Click to place a bend node and set its **Display Flag** then set: **Bend** to **75, Capture Origin** to **0, 0, -30, Capture Direction** to **0, 0, -1,** and **Capture Length** to **5.**

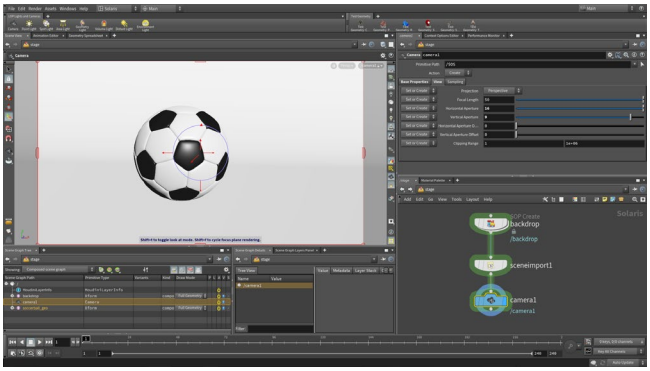
RMB-click on the *grid* node's output and type **Subdivide.** Set its **Display Flag** then set **Depth** to **2.**.



04

Go back to the Stage level. Wire the *backdrop* node into the *sceneimport* node. **RMB-click** on the output of sceneimport and type out Camera. Press **Shift-Enter** to place the node with its Display Flag on.

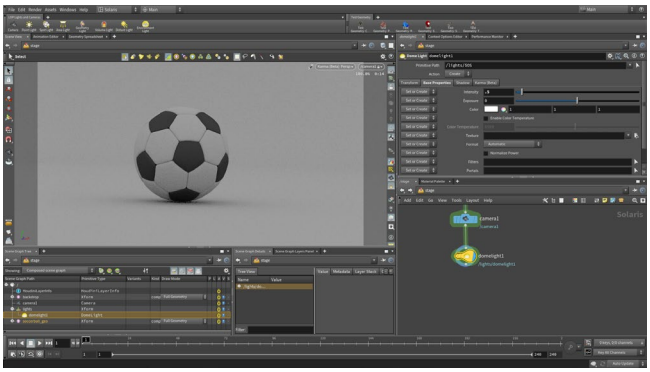
In the Scene View, you will see camera handles at the origin. Zoom out and look down at the scene then adjust the handles so the camera is looking at the soccerball from the left. You may want to activate the Construction plane so that you move the handles along the ground. You can then use the axis handles to lift up from the ground.



05 In the top right of the **Scene View**, click on the *No cam* menu and choose *camera1*. Now you are looking through the camera and can adjust how it looks. In the Parameter pane, set **Horizontal Aperture** to **19** and **Vertical Aperture** to **9** to get the proper aspect ratio.

This is probably not the view you are looking for therefore some view changes are needed. On the right side of the Scene View, click on the **Lock camera to view** button. Now use the **View tools [Spacebar-LMB/MMB/RMB]** to reposition the camera.

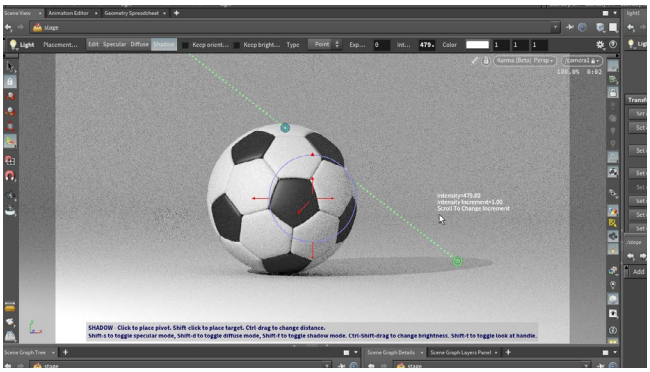
IMPORTANT: When you finish, toggle off this button.



06 From the LOP **Lights and Camera** shelf, **Ctrl-click** on the **Environment Light** tool. This adds a domelight node. Connect the *camera* node into the *domelight* and set the **Display Flag**. The camera view may become lost in the Scene view therefore click on the *No cam* menu and choose *camera1*.

Now click on the menu just to the left of the camera menu, and set it to **Karma**. Now you are using the Karma renderer in the viewport.

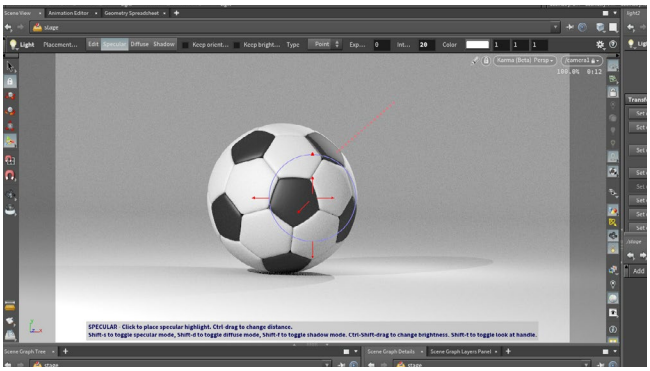
Set the **Intensity** to **0.5** on the *domelight* to tone it down a bit.



07 In the Network view, **RMB-click** on the *domelight* node's output and type **Light** then press **Shift-Enter** to place the node. It is at the origin by default. Set the view back to Houdini GL.

With the node active, press **Shift-F** to turn on the **Shadow** mode. You can also click on it in the Operation Control bar. Now **click** on the top of the soccerball to set a pivot point then **Shift-click** to place a larget on the ground. **Ctrl-Drag** to set the light distance.

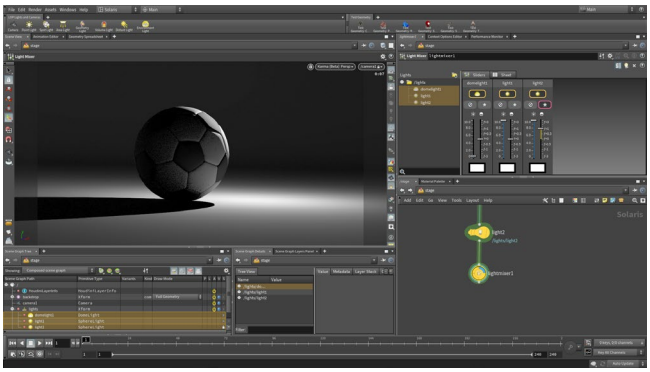
Set the view back to **Karma**. Now you can use **Ctrl-Shift-drag** to change the intensity of the light. You may need to set it quite high go see some impact on the look of the soccerball.



08 In the Network view, **RMB-click** on the *lights* node's output and type **Light** then press **Shift-Enter** to place the node. Set the view back to Houdini GL.

With the node active, press **Shift-S** to turn on the **Specular** mode. Now click on the right side of the soccerball to define a specular area of focus.

Set the view back to **Karma**. Now you can use **Ctrl-Shift-drag** to change the intensity of the light.

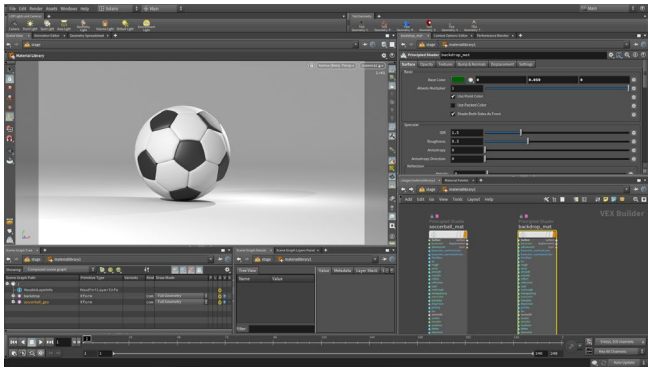


09 In the Network view, **RMB-click** on the *lights* node's output and type **Light Mixer** then press **Shift-Enter** to place the node. This will create a special panel in the Parameter pane which has a list of lights on the left side.

Open the folder and drag the three lights to the area on the right. You can now use these to tweak the lighting. Click on the Star icons to Solo each light to determine its contribution then tweak Exposure to adjust the lighting. Since the **intensities** are so high you can click on the icon above the intensity bar and from the pop-up set a Max value that works for your shot.

PART SEVEN:
Lookdev: Materials

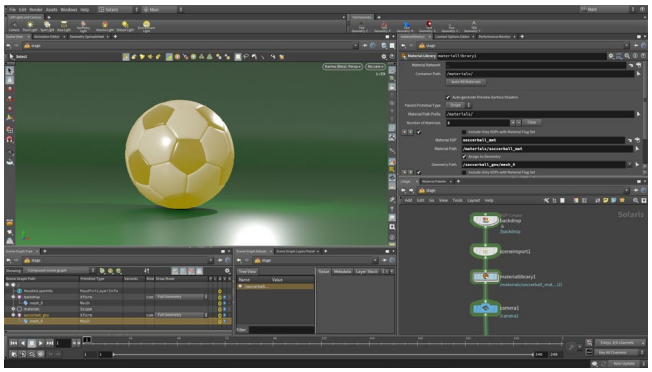
Materials and shaders can also be created within the LOPS/Solaris context. This involves adding the materials to the Scene Graph then assigning them to the geometry. The Materials are created inside a Material Library node then assigned at the LOPS/Solaris level. To add textures to the ground surface, UVs will have to be created to position the maps properly.



01 In the Network view, press **tab > Material Library** and place the node inbetween the *sceneimport* and the *camera* nodes. **Double click** on the node to go down to the **VEX Builder** level.

Press **tab > Principled Shader** and place the node down. Rename it *soccerball_mat*. Change its **Base Color** to **white (1, 1, 1).**

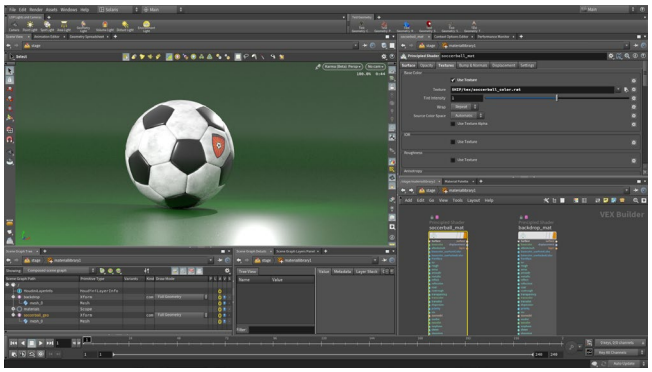
Alt-drag on this node to create a second principled Shader and rename this one *backdrop_mat*. Change its **Base Color** to a dark green.



02 Go back to the **Stage** level. On the Material Library node, click on the **Auto-fill Materials** button. This adds the principled shaders to the material list. They are not yet unassigned.

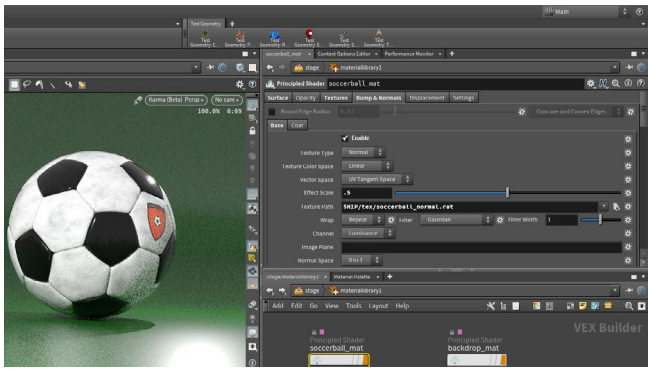
In the Scene view, get the **Select** tool and select the *backdrop* geometry. In the **Geometry Path** section under the *backdrop_mat* click on the **arrow** button to use this primitive.

Now select the soccerball geometry and In the **Geometry Path** section under the *soccerball_mat* click on the **arrow** button. Now both materials are assigned to the objects in our shot.



03 Double-click on the *materiallibrary* node to dive into it and select the *soccerball_mat* node. Click on the **Textures** tab and under **Base Color** click on **Use Texture** then use the button next to **Texture** to call up the file window. Click on **$HIP** in the side list then click on the *tex* folder to open it and then click once on *soccerball_color.rat* to select it. Click **Accept** to assign the texture to the material.

The $HIP reference makes sure that the reference is relative to the location of your scene file. That way if you were to move your project directories to another computer the reference will still work.



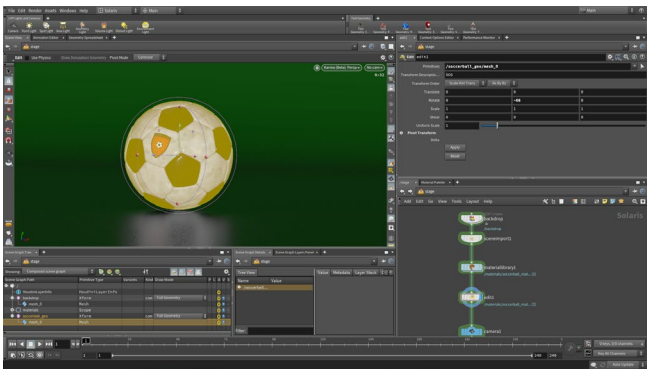
04 Under the **Textures** tab and use the technique you learned in the last step to assign textures to **Roughness** and **Reflectivity**. You will find the appropriate textures in the *tex* folder.

Go to the **Bumps & Normals** tab on the material and click the **Enable** button. Click on the arrow next to **Texture Path** and from the tex directory choose the *soccerball_normal.rat* file. Set the **Effect Scale** to around **0.5** and see how it looks.

MATERIALS IN HOUDINI

Materials in Houdini live in the **VEX Builder** context which in this case is nested inside the **Material Library** node. A material is made up of VOP nodes that define all of its material qualities.

The **Principled Shader** is an uber material that can be used on its own to assign texture maps and achieve a large variety of looks. You can also build your own shaders and materials for more advanced looks.

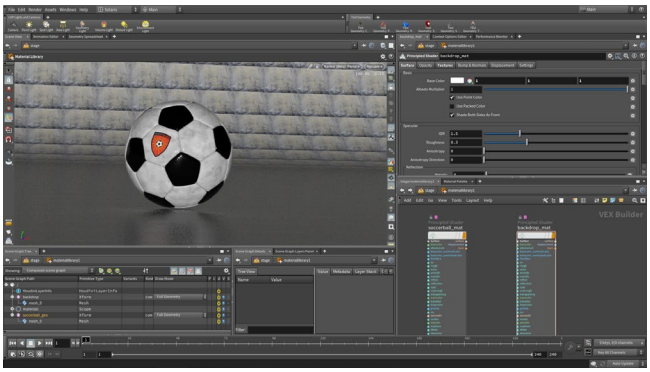


05

In the Scene view, look through camera1 to see your shot. The position of the soccerball's logo isn't quite right in relation to this camera therefore you need to rotate the ball.

Get the **Select** tool and click on the Soccerball to highlight it. Now click on the **Rotate** tool and use either the handle or the Parameter pane to rotate the ball until you like the logo position. You can do this in the **Karma** view or switch to **Houdini GL** for quicker updates.

This adds an *edit* node to the network. If you don't like the edits you make then you can delete the node to clear the slate. To deselect the soccerball, press **Ctrl** and click on it.

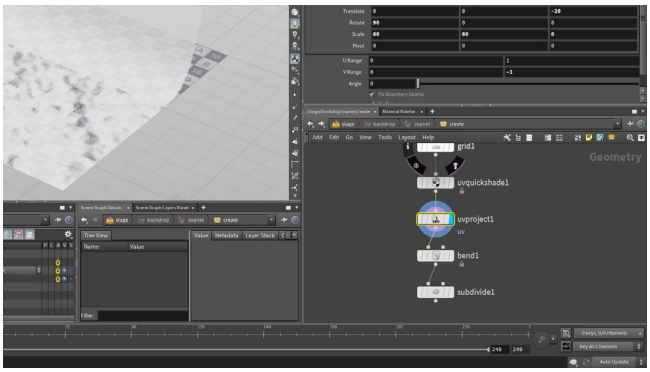


06

Now lets add some texture maps to the Backdrop material. **Double-click** on the *materiallibrary* node and select the *backdrop_mat* node.

Set the **Base Color** to **1, 1, 1** because this color will be multiplied with the texture map. Now click on the **Textures** tab and under **Base Color** click **Use Texture**. Click the **File Selector** button and use `$HIP` to go to the `/tex` directory and choose *backdrop_color.rat*. You can also add the *backdrop_reflect.rat* texture to **Reflectivity**.

You can see in the **Scene View** that UVs are not set up properly and the texture map isn't working properly.

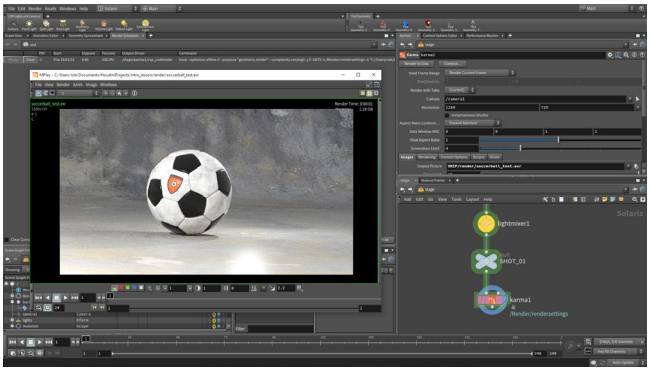


07

Go back to the Stage level then double-click on the *backdrop* node to go to the geometry level. Add a UV **Quickshade** node between the *grid* and *bend* nodes. Now add a **UV Project** node between the *quickshade* and the *bend*. We are putting the nodes here so they are created before the surface is bent.

On the *uvproject* node, click on the **Initialize** tab and click the **Initialize** button. Go back to the **Transformation** tab and set **V Range** to **0, -1**. This will orient the UVs properly.

IMPORTANT: Set the **Display Flag** back to the *subdivide* node.



08

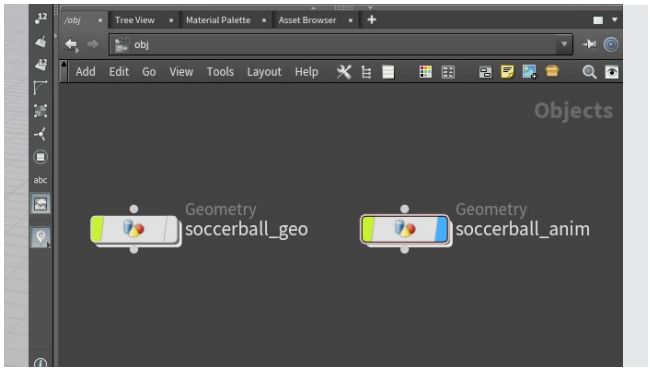
Go back to the Stage level and check out the rendering in the viewport. In the **Network View**, add a **Null** node and call it *SHOT_01*.

Then add a **Karma** node which you can use to render to disk. Set **Output PIcture** to `$HIP/render/soccerball_test.exr`. Click on the **Render to Disk** button at the top of the parameter pane.

Select **Render > Scheduler...** to get a panel that shows you the render progress. When you finish, choose **Render > Mplay > Load Disk Files** and open up the **EXR** file to review the final image.

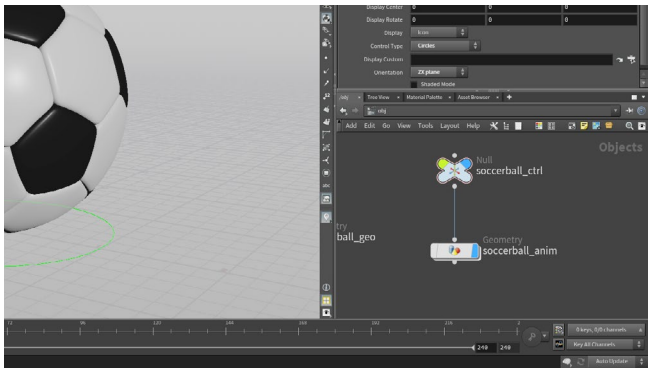
PART EIGHT:
Rig the Soccerball

In order to create an animation of the ball bouncing, you will start by building a simple rig that will make it easier to keyframe. This will involve setting up null objects so that you can work interactively in the viewport and adding nodes to the soccerball geometry network to accommodate the ball rotation along with squash and stretch.



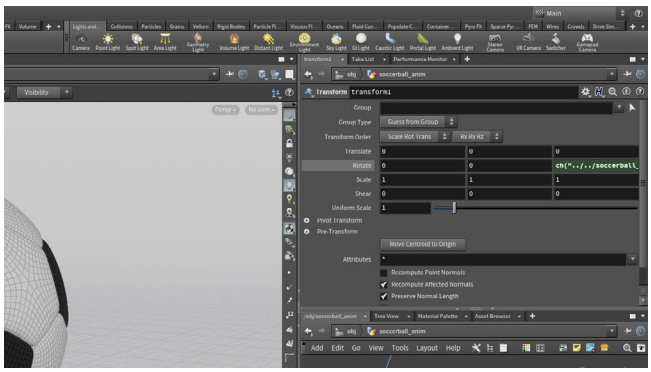
01 Change back to the **Build** desktop and navigate to the object level by clicking on one of the path bars and choosing obj. Now in the network editor, **Alt-drag** on *soccerball_geo* to make a copy of it. Rename this node socccerball_anim.

You will use *soccer_anim* to build your rig. Now turn off the Display Flag on the *soccer_geo* node to hide it. You don't want to make changes to the original setup because that object is being used in SHOT 1 in the Solaris context. This new soccerball will be used for an animated SHOT 2.



02 On the **Create** shelf, click on the **Null** tool then press **Enter** to place it at the origin. Name it *soccerball_ctrl*. Go to the **Misc** tab and set **Control Type** to **Circles** and **Orientation** to **ZX Plane**. Set the **Display Uniform Scale** to **4**. This creates a handle for the rig that is easy to select that won't render later on.

In the network editor, connect the input of the *soccerball_anim* object to the output of the *soccerball_ctrl* null to create a child/parent relationship. Moving the null will move the ball. Turn off the **selection flag** on the *soccerball_anim* so that you don't select it by accident in the viewport while animating. You will use *soccerball_ctrl* instead.

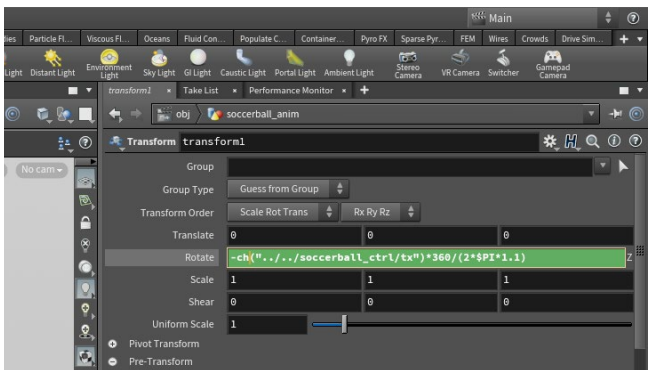


03 Select the *soccerball_ctrl* node. In the Parameter pane, click on the **Transform** tab then **RMB-click** on the **Translate X** parameter. Choose **Copy Parameter**.

Dive into the *soccerball_anim* object. Add a **Transform** node between the *subdivide* and the *matchsize* nodes. **RMB-click** on the **Rotate Z** and choose **Paste Relative References**. This places a channel reference expression in this parameter.

```
ch("../../soccerball_ctrl/tx")
```

This will connect the movement of the control object to the rotation on this node.

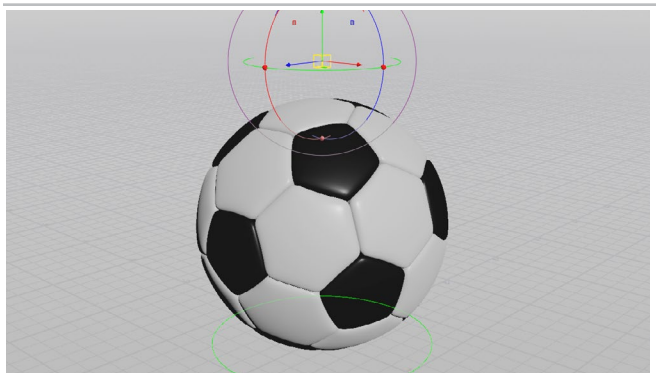


04 Click on the parameter to expand the channel. You are now going to use the **ball's circumference (2πr)** to determine the ball's rotation as it moves forward.

Edit the expression to read:

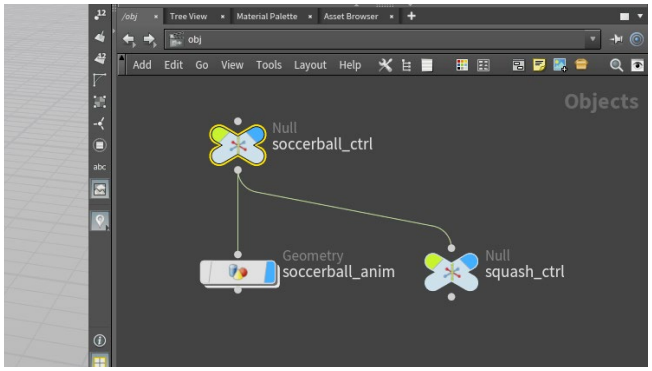
```
-ch("../../soccerball_ctrl/tx")*360/(2*$PI*1.1)
```

First you add a negative **(-)** at the front. You then multiply the position of the ball by **360** degrees and divide by **2πr**. At the object level, move the *soccerball_ctrl* along the **X axis**. The expression will rotate the ball to match the motion. Put it back at the origin when you are finished.



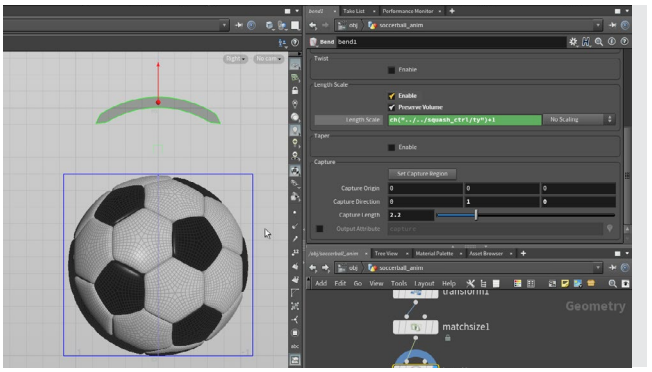
05 In the Viewport, create another Null object at the origin. Call it *squash_ctrl*. Go to the Misc tab and set **Control Type** to **Box** and the **Display Uniform Scale** to **0.2**.

Move the null up just above the ball. **Translate Y** should be about **2.5**. In the Parameter pane, choose the **Modify Pre-Transforms** menu and select **Clean Translates**. This sets the Translate Y value of the null to **0** even though it is above the ground. In order for this null to drive the squash and stretch, it needs a default value of 0.



06 Parent the *squash_ctrl* null to the *soccerball_ctrl* null. This will ensure that this secondary null moves when you animate the control null.

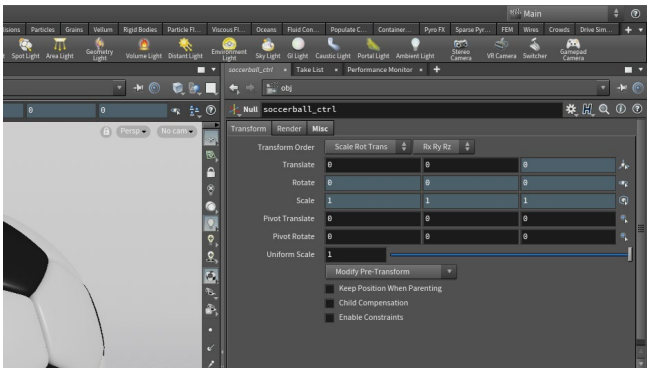
RMB-click on the *squash_ctrl* node's **Translate Y** parameter. Choose **Copy Parameter**. You will use this parameter to drive the squash and stretch of the ball. This will allow you to control the squash and stretch from the viewport interactively.



07 Go into the *soccerball_anim* object. Add a **Bend** node after *matchsize* and set its **Display Flag**. Set **Limit Deformation to Capture Region** to off.

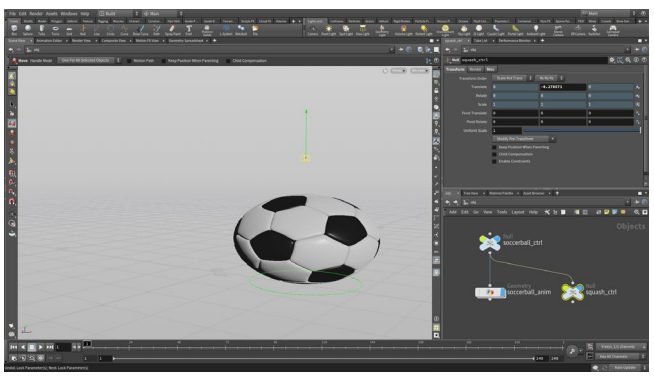
Go to a **Right View** then click the **Set Capture Region** button. Turn on **Grid Snapping** and place a point at the base of the ball and another at the top. This should set **Up Vector** to **0, 0, 1**, **Capture Direction** to **0, 1, 0** and **Capture Length** to **2.2**.

Turn on **Length Scale** and **Preserve Volume** then **RMB-click** on **Length Scale** and choose **Paste Relative References**. Add a **+1** at the end of the expression.



08 Go to the Object level and a perspective view. **RMB-click** on the **Transform X** and **Transform Z** parameters and choose **Lock Parameter** to lock these parameters on *squash_ctrl*. **RMB-click** on the **Scale** and **Rotate** parameters and choose **Lock Parameter** to lock all three channels.

Select the *soccerball_ctrl* object. **Lock** all the channels except **Translate X** and **Translate Y.** Now when you select the controls you will only see handles for the unlocked channels. This will make it easier to work with the rig because the animator can only manipulate the chosen parameters.



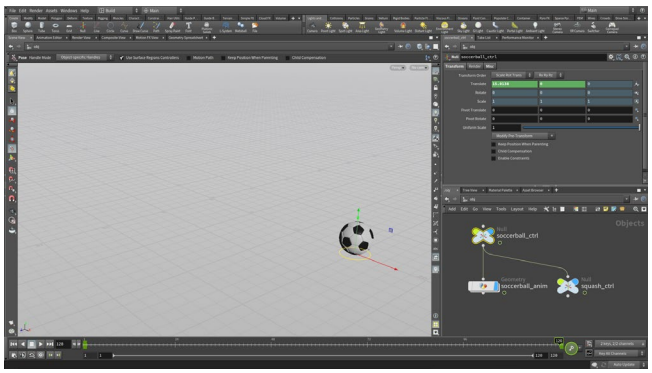
09 Now test out the rig by moving it around in X and Y and using the second handle to squash and stretch it. Once you are sure that all the parts are working, return all the values to 0 and get ready to animate.

You may want to turn off **Secure Selection** in the toolbar on the left side of the Scene view. This will make it easier to select the two control nulls while in the **Move** tool. If not then you will need to press the **S key** everytime you want to switch selections.

Save your scene file before proceeding.

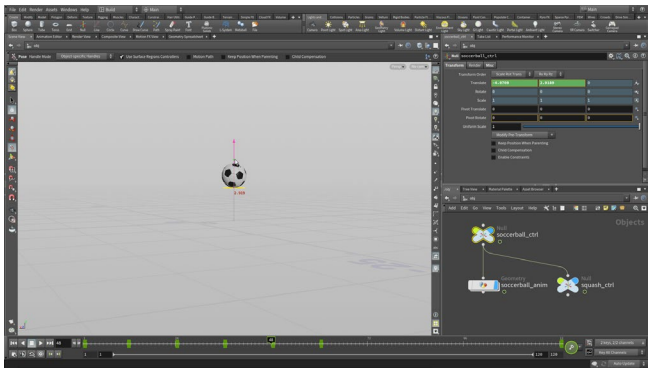
PART NINE:
Animate a Bouncing Ball

You can now take the soccer ball rig and use it to animate the ball bouncing. You will learn how to set keyframes, adjust animation curves and work with time-space handles in the viewport. The bouncing ball is a classic animation exercise that offers a great opportunity to learning the basics of animating in Houdini.



01 At the bottom left edge of the **Timeline**, click on the **Global Animation Options** button. Set the *End* to **120** and click **Close**. This will set the timeline range to 120 frames.

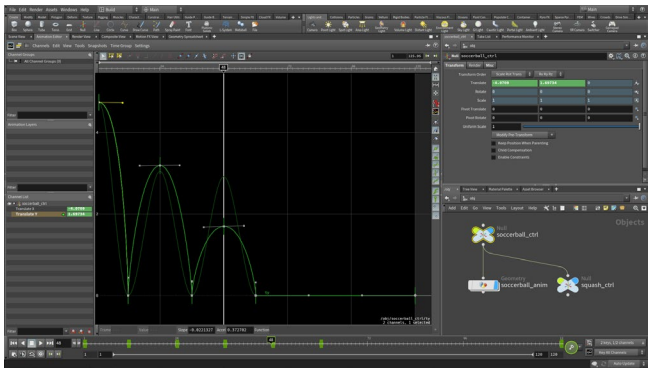
Make sure you are on **frame 1**. Click on the **Pose** tool on the toolbar to your left and select the *soccerball_ctrl*. Move the ball to around **-15** in X and **press K** to keyframe it. Move the timeline to **frame 120.** Move the ball to the around **15** in X and **press K** to set a second keyframe. Scrub through the timeline to make sure that the ball is animating. It should be moving and rotating based on the rig design.



02 Move the timeline to frame **12** and **press K** to set an intermediate key. Repeat at frames **36** and **60**. All of these keyframes are sitting on the ground.

Now go to **frame 1** and lift the ball up in the Y direction. You don't need to set another key because this move simply updates the keyframe you already set at frame 1 .

Move to **frame 24** and lift up the ball in the Y direction a little less than you did at frame 1. **Press K** to set a keyframe. Move to **frame 48** and lift the ball up even less. **Press K** to set another keyframe.



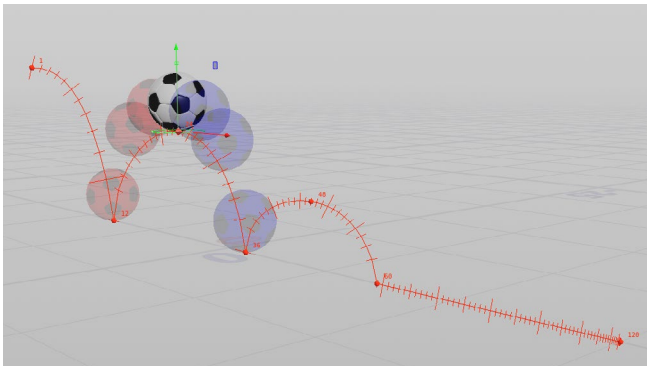
03 **Scrub** through the timeline to see that the ball appears to float whereas you want hard hits when the ball contacts the ground. Click on the **Animation Editor** pane tab.

From the Scoped parameter list, click on the *Translate Y* channel. **Press H** to home the view of the curve. Select the three keyframes where the ball contacts the ground and press the **Untie Handles** button on the **Functions** bar found just above the graph. Now click in empty space to deselect then start tweaking the tangent handles to create a sharp bounce at each point. You can also stretch out the handles at the top to slow the ball down at the peak.

THE POSE TOOL

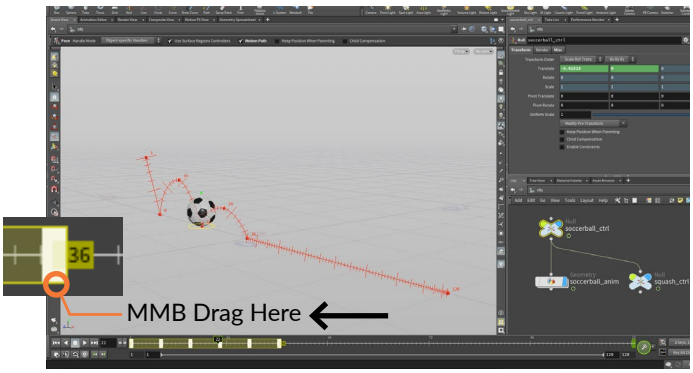
You can easily manipulate the two control objects in the soccerball rig using the **Move** or **Handle** tool. The advantage of using the **Pose** tool is that it gives you access to the **Motion Path** handle and if you were working with Inverse kinematics, there are special handles that you can use to control the system. Therefore be sure to remember this tool when you start setting keyframes on your rigs. The **Secure Selection** does not restrict the **Pose** tool from selecting different objects.

Select
Secure Selection [~]
Move [T]
Rotate [R]
Scale [E]
POSE [Y or Ctrl R]
Handle [Enter]



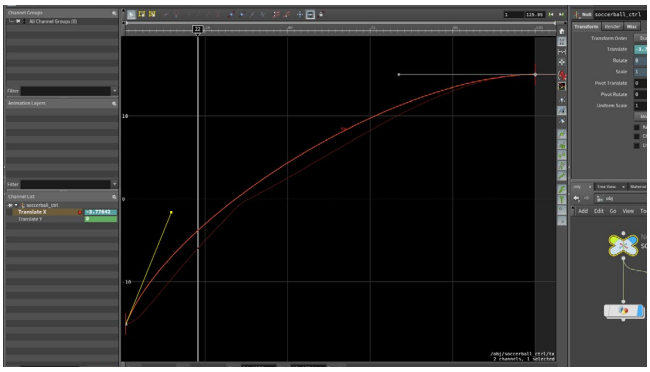
04 Go back to the **Scene view** and preview the results. You now have a sharper hit each time the ball hits the ground. Make sure you have the **Pose** tool active and the *soccerball_ctrl* node selected. In the top bar, turn **On** the **Motion Path** handle. This shows a path outlined where the ball is bouncing. Click on each keyframe marker to tweak the pbounce. **RMB-click** on the handle to display tangents for more control over the curve.

Go to the *soccerball_anim* object's **Misc** tab, set **Onion Skinning** to **Full Deformation**. Press **spacebar-d** and from the **Scene** tab adjust **Frame increment** and the **Frames Before** and **After** color.



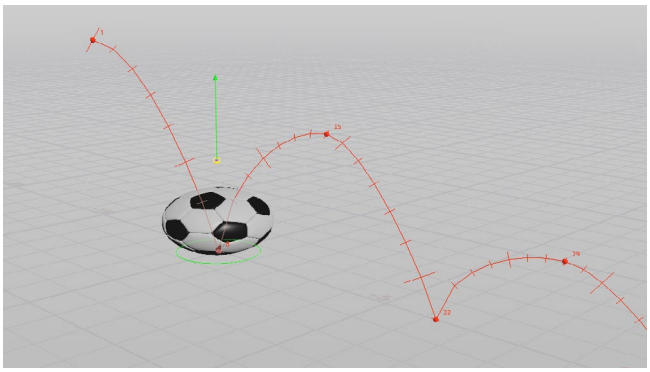
05 To adjust the timing, you can also use the timeline. Press **Shift** and drag a bounding box from frame 1 to the last key in the timeline **to select** all the keys. Next, **MMB drag** on the end of the box underneath the handle to scale the timing of the bounces to speed them up. You can also select each key using **MMB** and then drag with **MMB** to time each keyframe the way you want.

This is where you will determine the timing of the bounces. Keep exploring until you get the look that you want. Note that there may be some awkwardness in the bouncing because of the translate X values. You will fix that in the next step.



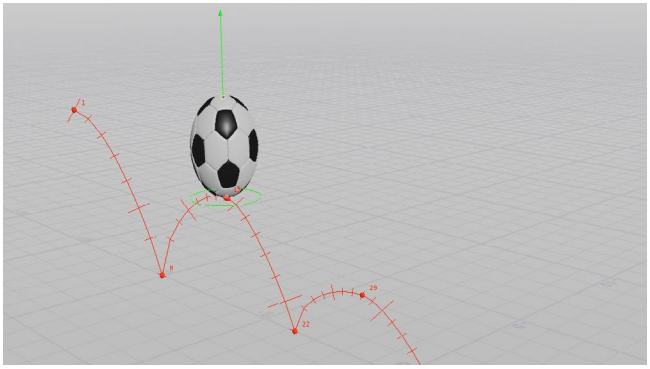
06 Click on the **Animation Editor** pane tab and you will see two curves. From the Scoped Parameters list, click on **Translate X** for *soccerball_ctrl*. Now select all the keys except for the first and the last. Press **Delete**. Now use the curve handles to go from a high slope to a low slope. This will have the ball moving faster at the beginning and slower at the end.

Note that if you go back to tweak the points in X on the motion path handle, you will get strange results because there are no longer intermediate keys in that direction - only use it to tweak in Y from this point on.



07 Go back to the Scene view. **RMB-click** on the **Motion Path** handle and choose **Persistent**. This will keep it around as a guide as you set keyframes on the squash and stretch.

Select the *squash_ctrl* and turn off its **Motion Path**. Go to the first bounce and move back one frame. Select the *squash_ctrl* handle and stretch out the ball a bit. **Set a Keyframe** using the **K** key. Now go to the bounce frame and move the handle down to create squash. **Set another key**. Go two frames forward and stretch the ball out again. **Set another key**. Repeat for all the bounces.



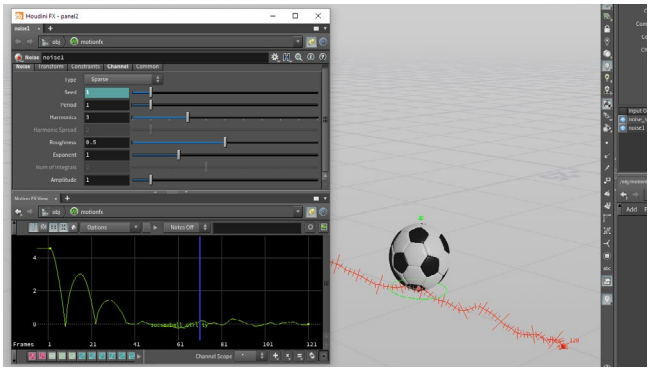
08 Now go to the peaks of the bounces and add in a bit more stretching. **Set keyframes**. When you are finished, scrub and playback the motion to preview the results. Make sure the **Real Time Toggle** is **On** in the Timeline to properly evaluate the motion.

You can now use the **Animation editor** to make tweaks to the squash and stretch. Make sure you keep the keyframes aligned with the bouncing of the soccer balls.

When you are finished, **RMB-click** on the **Motion Path** handle and turn off **Persistent** so that it isn't visible when you deselect.

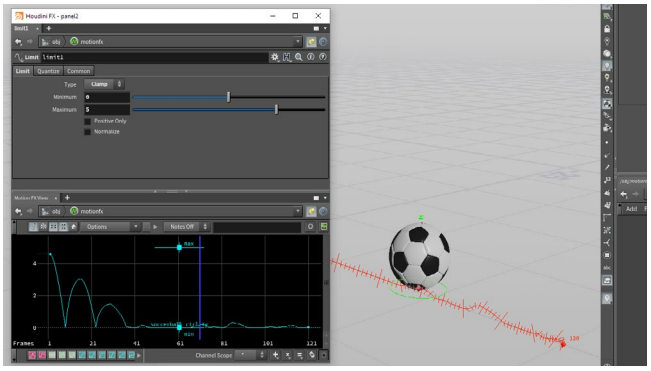
PART TEN:
Add Motion FX

In addition to keyframing everything by hand, you can also use the Motion FX menu to create procedural animation nodes that make it easier to add detail to the motion. These nodes are Channel Operators or CHOPs and they offer a whole new way for animators to create procedural motion. You can add effects that act on all of the motion at the same time or focus on one channel.



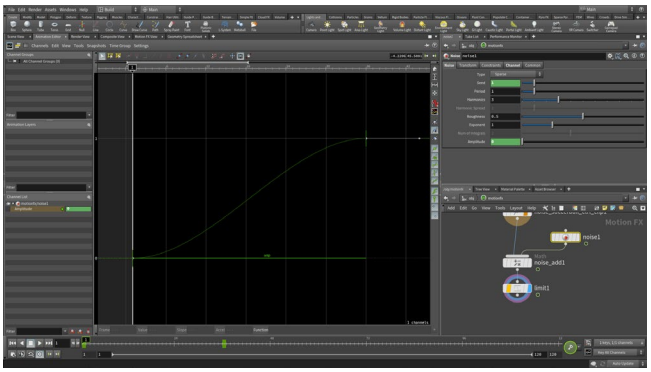
01 Select the *soccerball_ctrl* null object. **RMB-click** on the **Translate Y** and choose **Motion FX > Noise**. A panel pops up with parameters that you can use to control the noise. A new subnetwork was created with the CHOP nodes that send their information back to the *soccerball_ctrl's* **Translate Y** channel.

Set the **Amplitude** to **5** and press **Play** to see how this looks. This adds dramatic up and down motion which make it feel like there is some serious turbulence. Set the **Amplitude** to **1.** This offers a more subtle bump to the motion of the ball.



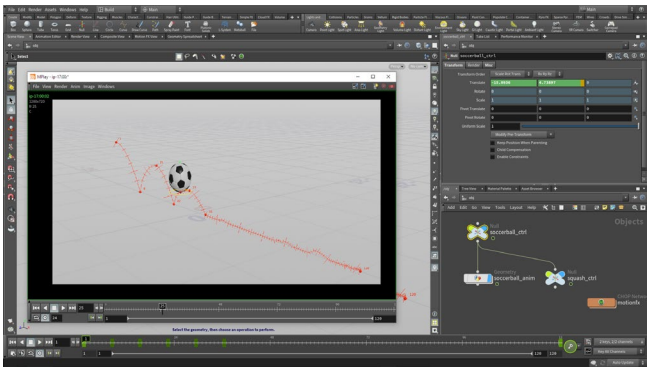
02 Some of this motion goes below the ground. You will want to focus on creating bumps above the ground.

Go back to the object level. Select the *soccerball_ctrl* null object. **RMB-click** on the **Translate Y** and choose **Motion FX > Limit**. Set **Minimum** to 0 and **Maximum** to 5. Now the ball moves flat with some bumps instead of going up and down the whole way.



03 There should not be any noise while the ball is bouncing. It is only needed when the ball is rolling. You can keyframe the **Amplitude** to turn the noise on and off.

In the newly created *motionfx* network, select the *noise1* CHOP node. Go to the frame where the ball stops bouncing and starts rolling. **Alt-click** on **Amplitude** to set a keyframe. Go to **Frame 1** and set **Amplitude** to **0**. **Alt-click** on **Amplitude** again to set a second keyframe. Go to the **Animation Editor** and select the curve. Click on the **Constant** button in the Functions bar. This creates a sharp cut from no amplitude to an amplitude of 1.



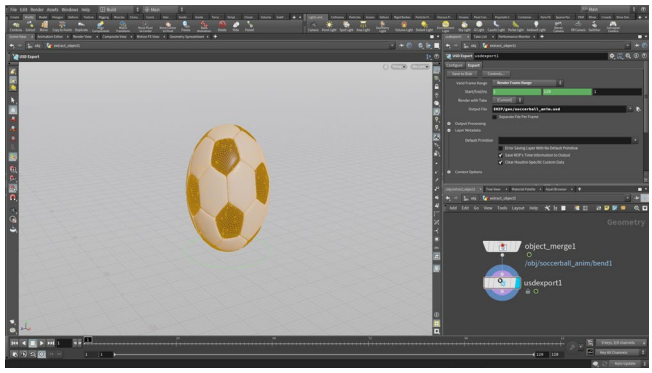
04 In the toolbar at the left side of the Scene view, click on the **Render Flipbook** button. Leave the default settings and click **Start**. Wait while the sequence is captured and then you will see the flipbook in an **Mplay** window. You can **Play** this back and scrub through to evaluate your motion.

You can now go back and adjust the curves on the motion or the channel operators in CHOPs to get the result you are looking for. Then you can re-flipbook to evaluate.

Save your work.

PART ELEVEN:
Lights, Camera, Action!

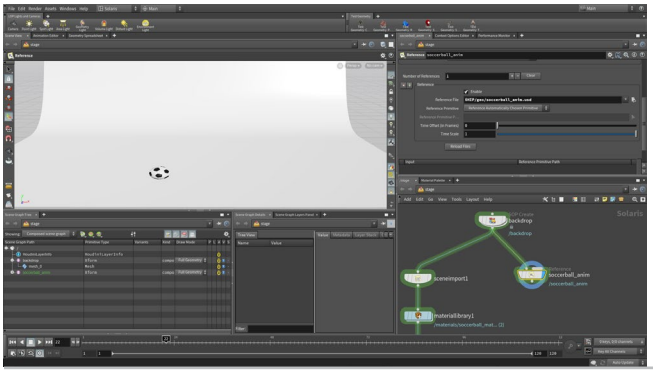
To render out the animated soccer ball, you will need to go back to the Solaris environment and set up a second shot. You will begin by branching off new LOP nodes from backdrop geometry then adjust the lights and cameras to suit the bouncing soccer-ball animation. You will also set up motion blur for the deforming geometry.



01 Go to the Object level. **Double-click** on *soccerball_anim* to dive into it. **Press n** to select all the geometry then go to the **Modify** shelf and select **Extract**. This takes all the motion and bending of the ball and puts it in one network.

You can see an *objectmerge* node which is extracting the ball into a new object called *extract_object*. **RMB-click** on the output and find **USD Export** then press **Shift-Enter** to place this node.

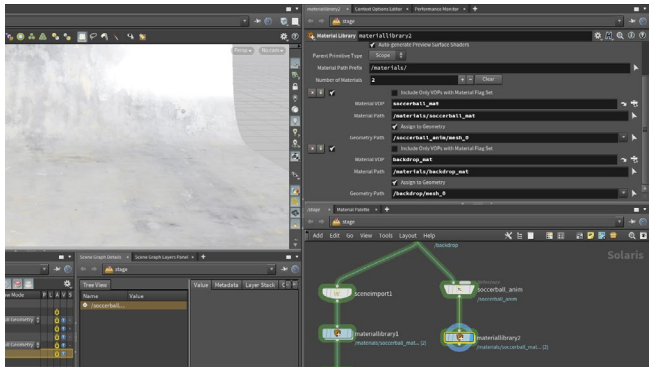
Click on the **Export** tab then set **Valid Frame Range** to **Render Frame Range** and set **Ouput File** to *$HIP/geo/soccerball_anim.usd*. Click the **Save to Disk** button to export the USD file.



02 Go back to the Solaris desktop and point it to /stage. In the **Network View** select move the backdrop node to the right so that you can branch off a second shot.

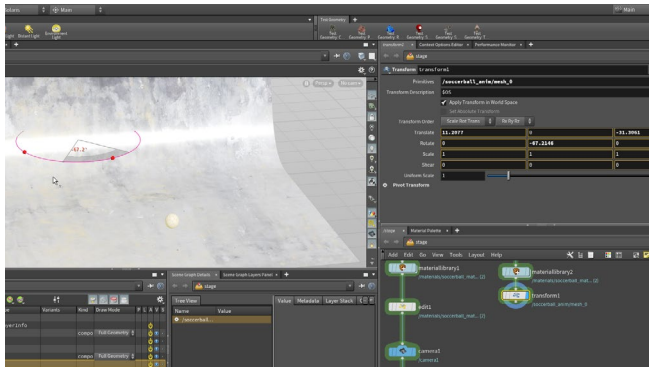
Zoom in and add a **Reference** node. Connect the *backdrop* node to this new node and set its **Display Flag**. Next to **Reference File**, click on the **File Chooser** and find the *soccerball_anim.usd* file. Rename the node to *soccerball_anim*.

Reposition the Scene view so you can see the soccerball geometry. Scrub in the timeline (make sure you are in Houdini GL view) to see the cached animation which is part of the USD file.



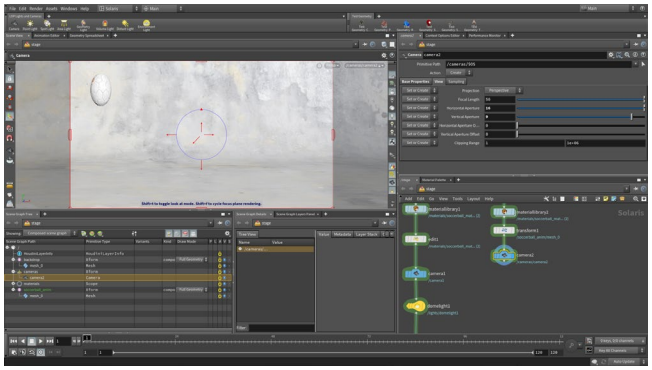
03 In the Network view, **Alt-drag** on the *materiallibrary* node from the SHOT 1 network to create a copy of this node. Wire the *soccerball_anim* node into it then set its **Display Flag**. This will assign the material to the *backdrop* but since the soccerball is now animated it needs to be reassigned.

Use the **Select** tool and click on the new animated soccerball. Click the **Select Primitives** button next to **Geometry Path** for the new *soccerball_mat* to reassign material to the new geometry.



04 Use the **Select** tool and click on the new animated soccerball. In the Scene View, press **tab > Transform** to add a transform node to the graph.

In the Scene view, use the Transform handle to **Move** the Ball to the back of the backdrop in the middle. **Scrub** the timeline to watch the ball bounce to the right. Leave it at somewhere around frame 80. **Press R** to get the rotate handle. **Rotate** to move the ball so that it is bouncing at an angle from the backdrop. **Scrub** the timeline to see if you like its direction.

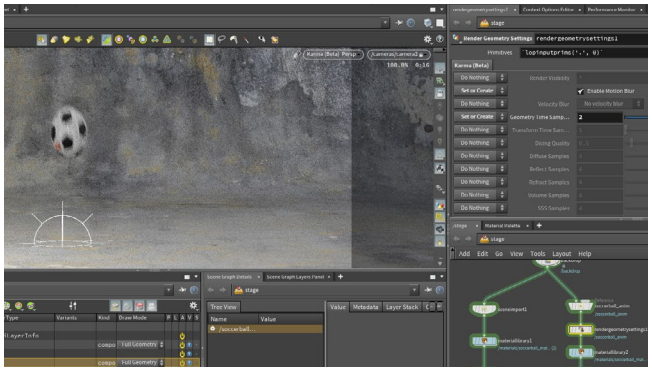


05 **RMB-click** on the output of the *merge* node, type **Camera** then press **Shift-Enter** to place it and set its **Display Flag.** Select this new node and under the **View** tab set **Horizontal Aperture** to **19** and **Vertical Aperture** to **9**. Go to the Scene View and change the **No Cam** menu to *camera2*.

Press the **Lock Camera/Light to View** button so that view changes can be used to reposition the camera. Now **Tumble**, **Pan** and **Dolly** in the viewport to reposition the camera so the ball begins in the top left and bounces down to the lower right. Scrub the timeline to make sure the camera works for the whole sequence.

06 In the Network view, select the last six nodes from SHOT 1 which includes the lights and the Karma node. **Alt-drag** to create copies of these nodes. Wire the *camera2* node into this chain. Set the **Display Flag** on the first light and select it then tweak to suit this shot.

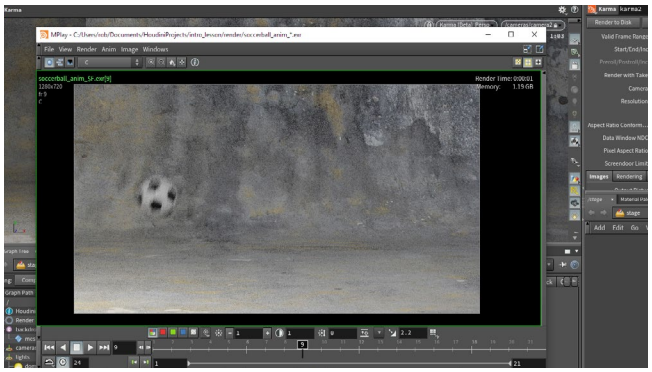
Use the light handles you learned about earlier to make lighting decisions and use the Karma display in the viewport to verify your setup. On the *lightmixer* node you will need to move over the new lights since there are new lights feeding into it.



07 If you go to around frame 18 there should be motion blur on the bouncing ball. Since this is deforming geometry, you need to add some settings for motion blur to work.

RMB-click on the output of *soccerball_anim* and start typing **Render Geometry Settings**. Click to place the node. In the Viewport, select the soccerball. In the Parameter pane, press the **Select Primitive** button next to **Primitives**. Set the **Display flag** back to *SHOT_2*.

Select the *rendergeometrysettings* node and choose **Set or Create** for **Enable Motion Blur** and **Geometry Time Samples**. Set **Geometry Time Samples** to **2**. You will now see motion blur.



08 Select the *karma2* node and make sure that **Camera** is set to */camera2*. Otherwise it will use the old camera. You can now render the sequence to disk.

Be sure to set **Valid Frame Range** to **Render Frame Range** and set the **Output Picture** to *$HIP/render/soccerbal_anim_$F.exr*. The *$F* in the name is needed to add frame numbers to the renderings.

When you finish, choose **Render > Mplay > Load Disk Files** and open up the rendered images to review the final sequence.

Save your work.

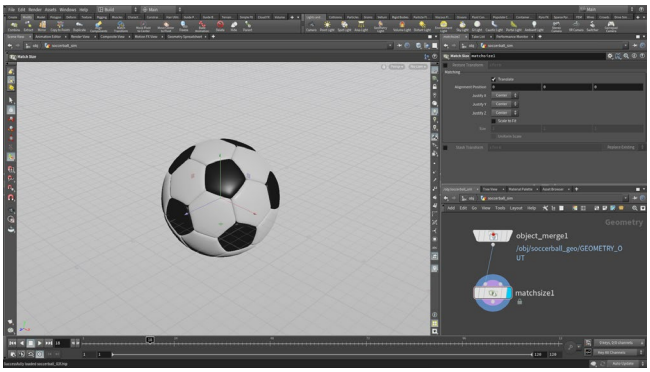
KARMA RENDERER

Karma is the highly advanced renderer included with Houdini for use with the Solaris/LOPS context. It implements **Physically-based** rendering and is a **HYDRA-based** renderer which means that it works with USD files. This allows it to be used in the **Viewport** for interactive updates or to be rendered to disk using a **Karma** node.

Note: With Houdini 18.0. the Karma renderer is in **beta**. You can choose other HYDRA-based renderers such as **Pixar's RenderMan** if you need to work with a production-ready solution.

PART TWELVE:
Set up a Rigid Body Simulation

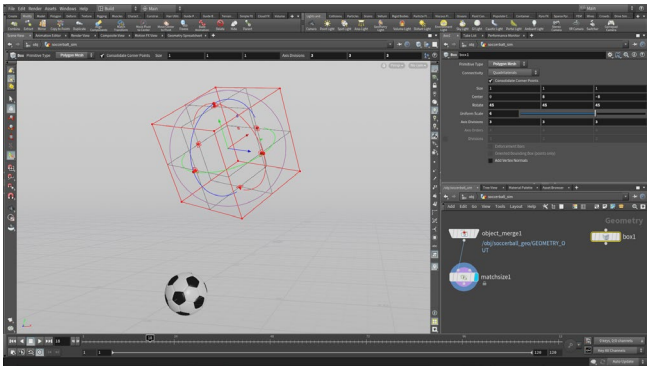
While traditional animation is great for animating a single soccer ball, dynamics would be a better option if you want to animate a bunch of soccer balls. Dynamics requires a simulation so that the solver can go frame by frame determining how each of the participating objects interact with each other. You will use packed geometry to get an efficient result for this simulation.



01 Change back to the **Build** desktop and navigate to the object level. Hide all of the animation rig nodes and the *extract_object* node by turning off their display flags. Turn on the *soccerball_geo* display.

Select the *soccerball_geo* node then from the **Modify** shelf click on the **Extract** tool. This creates a new object with the soccerball object merged. Jump up one level and rename *extract_object* to *soccerball_sim*. Hide the *soccerball_geo* object.

Dive back in to the *soccerball_sim* object to work with the geometry. Add a **Match Size** node to center the ball around the origin.

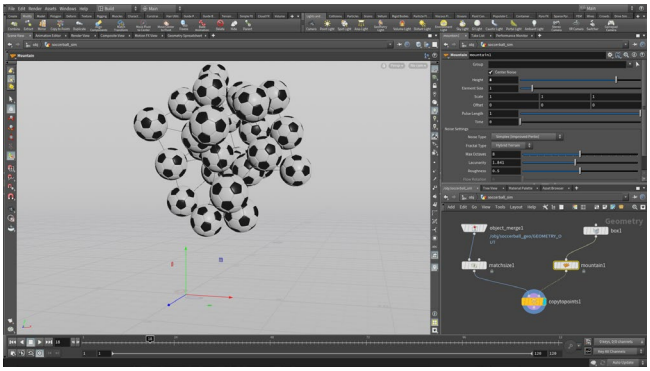


02 In the Network view, press **tab > Box** then place it to the right of the *matchsize* node.

Set the following on the *box* node:

- **Center** to **0, 8, -8**
- **Rotate** to **45, 45, 45**
- **Primitive Type** to **Polygon Mesh**
- **Uniform Scale** to **6**
- **Axis Divisions** to **3, 3, 3**

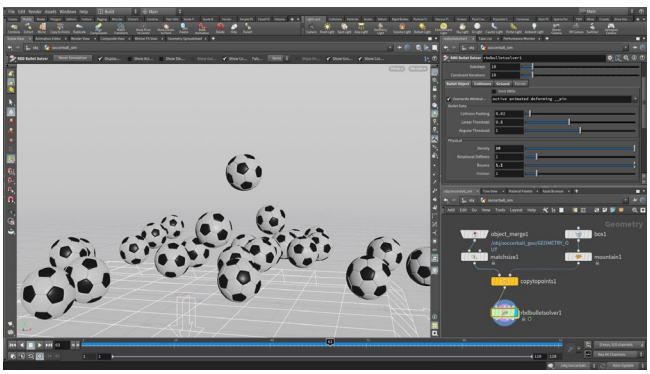
This puts it in the right position for the simulation.



03 In the Network view, add a **Copy to Points** node just below the other nodes. Wire the *matchsize* node into the first input and the *box* node into the second.

Turn **ON** the **Pack and Instance** option. This will create a faster simulation because the geometry is being instanced to the points of the cube. Set the *copytopoints* node's **Display Flag**.

Add a **Mountain** node inbetween the *box* and the *copytopoint* nodes. Set **Height** to **8**. This will jiggle the points so the starting shape doesn't look too regular.



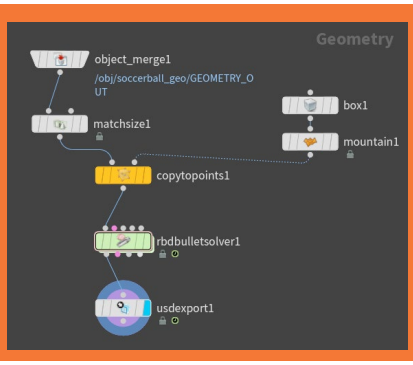
04 Make sure you are on **Frame 1**. Add a **RBD Bullet Solver** node after the *copytopoints* node. Click on the **Solver tab** and then the **Ground** tab, set **Add Ground Plane** to **Ground Plane**. Press **Play** to test out the simulation. The sim is cached which lets you scrub in the timeline to review the results.

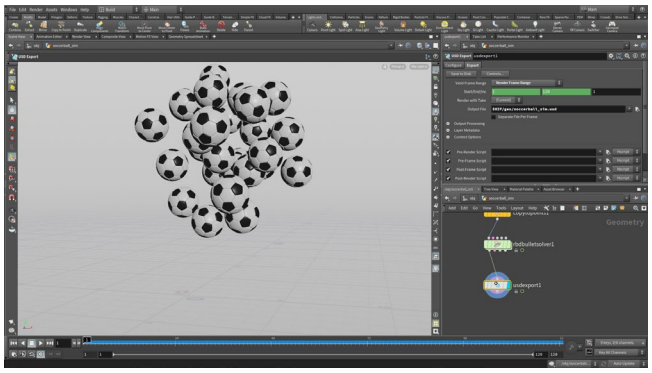
Under the **Bullet Object** tab, set **Density** to **10**, **Bounce** to **1.1** and under the **Collisions** tab, set **Bounce** to **0.8**. At the top of the Parameter pane for this node, click the **Reset Simulation** button and then press **Play** to resim. Scrub to evaluate.

DOPS hidden inside SOPS

In Houdini, simulations are processed using the **Dynamic Operators** or **DOPs**. With the **RBD Bullet Solver** node here in the **Geometry/SOP** context, you are actually working with a node that has a Dynamics network buried inside it.

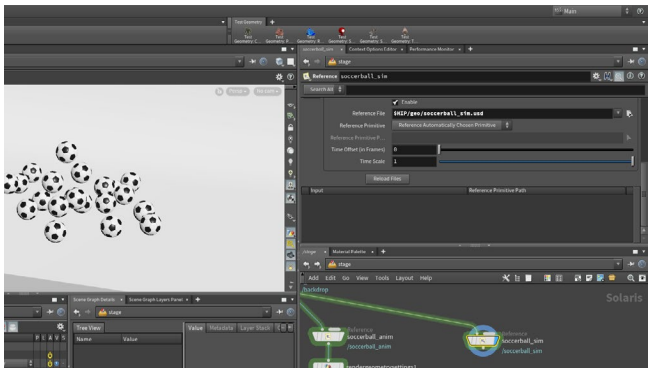
This makes it easy to set up at the geometry level with all the DOP nodes are wired up and ready to go but hidden from view. For simpler setups, working at the geometry level will give you a proper simulation. If you need more control over the different solvers then you would need to work directly in DOPs.





05 At the end of the chain, add a **USD Export** node and set its **display flag**. Set **Valid Frame Range** to **Render Frame Range** and set the **Output File** to *$HIP/geo/soccerball_sim.usd*.

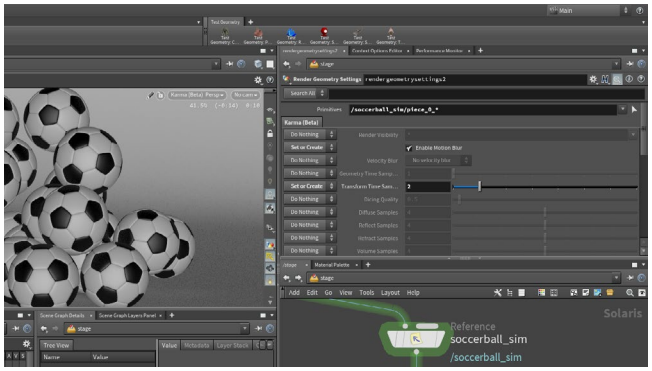
Click on the **Save to Disk** button and this will save the USD file into your geo directory. You will reference this cached asset into the Solaris setup as a third shot.



06 Change your **Desktop** back to **Solaris** and set the path to */stage*. Make sure you are choose **Houdini GL** from the *persp* menu.

Branch off a **Reference** node from the *backdrop* node and set its **Display Flag**. Set **Reference File** to *$HIP/geo/soccerball_sim.usd*.

Rename this node to *soccerball_sim*.



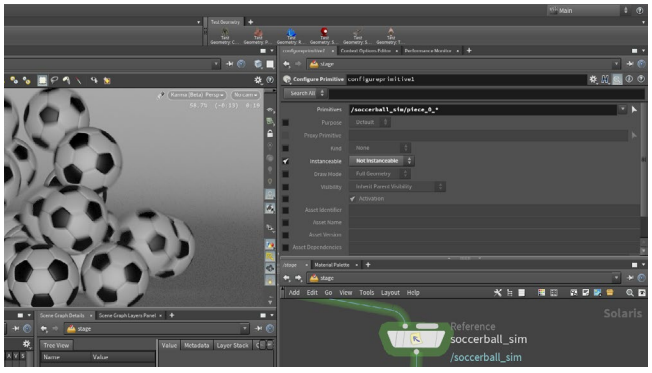
07 Add a **Render Geometry Settings** node after the *soccerball_sim* node and set its Display Flag.

Under **Primitives** enter:

*/soccerball_sim/piece_0_**

Beside the **Enable Motion Blur** checkbox, choose **Set or Create** from the menu. Beside **Transform Time Samples**, choose **Set or Create** then set the **Transform Time Samples** to **2**.

This works differently compared to the bouncing ball because the simulated soccerballs are not deforming.



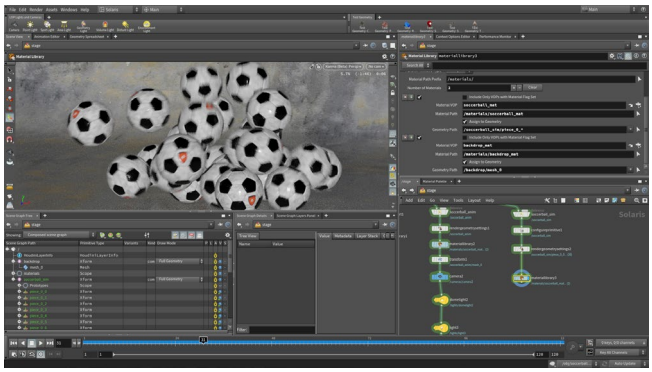
08 Add a **Configure Primitive** node between the *soccerball_sim* and *rendergeometrysetting* nodes.

Under **Primitives** enter:

*/soccerball_sim/piece_0_**

Turn on the checkbox next to **Instanceable** and set it to **Not Instanceable**. Scrub to a frame where the balls are falling and go to Karma display you will see motion blur.

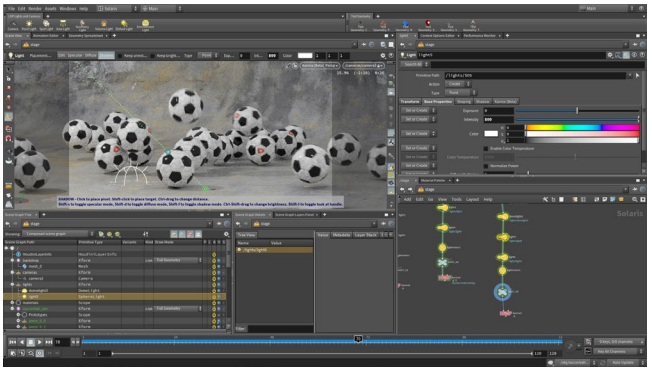
Note: Currently instanced (packed) objects will only motion blur in LOPS if you uninstance them.



09 Select the existing *materiallibrary* node and **Alt-drag** to make a copy of it. Wire the new *rendergeometrysettings* node into this chain.

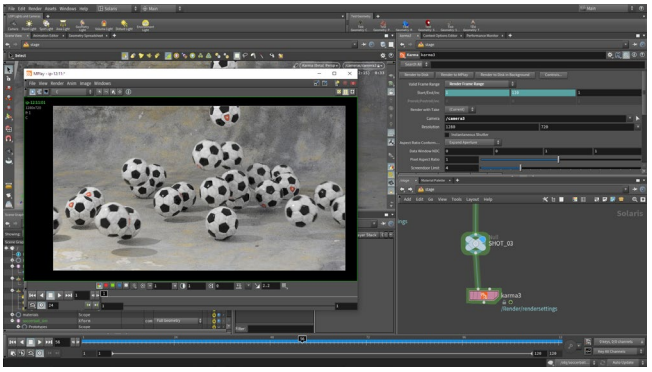
For the *soccerball_mat* **Material VOP,** set **Geometry Path** to */soccerball_sim/piece_0_**

The material will now be assigned to all of the soccerballs.



10 Select the existing *Camera*, *Light*, *Null* and *Karma* node's and **Alt-drag** to make a copy of them. Wire the new *materiallibrary* node into this chain.

Use the techniques you learned earlier in this lesson to adjust the Cameras and lights in the viewport. When you are happy then you can move on to rendering to disk.



11 Select the Karma node at the end of the chain. Make sure **Camera** is set to */camera3*. **Save** your work.

Click on **Render to Disk**. When it is finished, preview in **Mplay**.

CONCLUSION

You have now built a scene from scratch, touching on many different aspects of Houdini. You have modeled, set up textures, animated, rendered and simulated. Along the way you have learned about the different Houdini contexts and how to navigate back and forth between them.

While this lesson doesn't result in blockbuster VFX, it introduces you to fundamental skills which you will carry with you as you dive deeper into Houdini and begin exploring its comprehensive toolset.

There is a wealth of learning material available on the SideFX website to help you take your next steps.

Best of luck on your journey!