

HOUDINI FOUNDATIONS

モデリング、レンダリング、アニメーション

Houdini へようこそ。このレッスンは、サッカーボールのモデリング、レンダリング、アニメーション、シミュレーションをゼロから行います。スクワッシュ&ストレッチ (潰しと伸ばし) の理論を使用して昔ながらのバウンシングボールのアニメーションを作成し、テクスチャと材料を適用し、ライトとカメラを追加します。また、ダイナミクスを使用してサッカーボールの一群をシミュレートする方法も探ります。

はじめての Houdini シーンを作成し、Houdini のさまざまな部分を確認し、インターフェースや必須のツールなどを実際に使っていきます。**Scene View** でインタラクティブに作業する方法と、**ネットワークビュー** を使用してノードを管理しながら、モデルを整えたり、アニメーションリグを構築する方法を学びます。また、**Solaris ステージ** で材料とテクスチャをセットアップしてから、Houdini の組み込みのレンダラである **Karma** を使ってレンダリングし、最後には**リジッドボディシミュレーション**を作成します。

レッスンの目標

Houdini のプロシージャルなノードベースのワークフローを使用して、サッカーボールのモデリング、レンダリング、アニメーション、シミュレーションを行います。

学習内容

- **View ツール**を使用する方法
- シェルフ、**Radial メニュー**、**Tab キー**を使用する方法
- **ジオメトリ**を作成する方法
- **ノードとネットワーク**を使用する方法
- **カスタムアトリビュート**と **For-Each ループ**をセットアップする方法
- **材料**と**テクスチャ UV**をセットアップする方法
- ショットの**レイアウト**を作成し、**Karma**でレンダリングする方法
- **キーフレーム**を設定し、**Motion FX**を追加する方法
- **リジッドボディダイナミクス**を使用する方法

使用する機能とソフトウェア

Houdini 19.5+ の機能を前提として、書かれています。

このレッスンの手順は、以下の Houdini 製品で実行可能です。

Houdini Core	✓
Houdini FX	✓
Houdini Indie	✓
Houdini Apprentice	✓
Houdini Education	✓

ドキュメントバージョン 4.0 | 2022 年 7 月
© SideFX Software



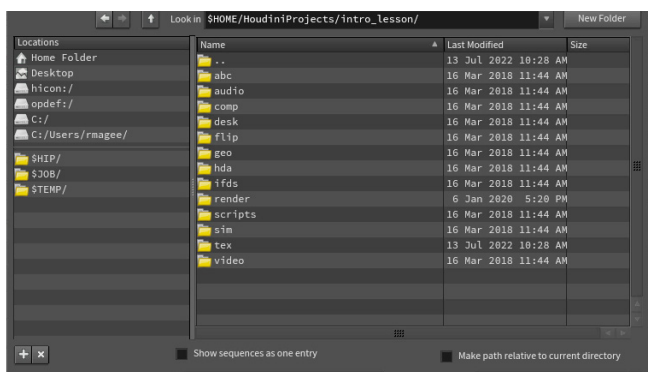
パート 1

Houdini UI の確認

はじめに、Houdini ワークスペースと、頻繁に使うことになる3つのペインを理解しておきましょう。ビューポートではオブジェクトをインタラクティブに作成でき、パラメータエディタではノードプロパティを編集でき、ネットワークエディタではノードネットワークを直接操作することができます。

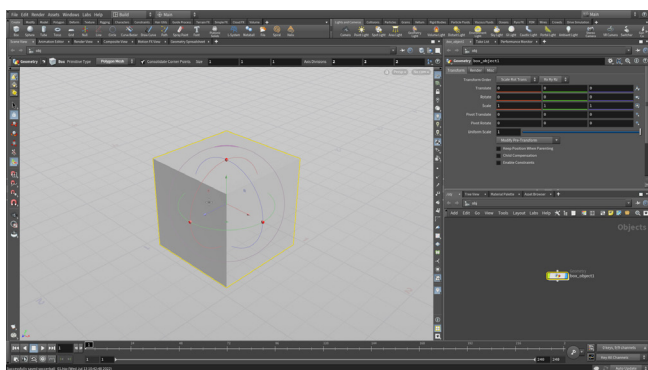
プロジェクトファイル

SideFX.com の サッカーボールチュートリアル のページ (このドキュメントを入手した場所) から、*intro_lesson* ディレクトリをダウンロードします。home または documents ディレクトリにある、*Houdini Projects* ディレクトリに配置してください。



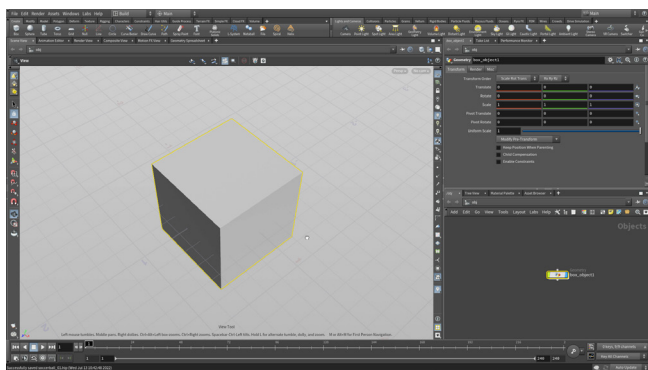
01 **File > Set Project** を選択します。ダウンロードした *intro_lesson* ディレクトリを見つけ、**Accept** を押します。これで、このショットに関連するファイルがすべて、先ほどコピーしたプロジェクトディレクトリとそのサブフォルダに配置されるようになります。

File > Save As... を選択すると、新しい *intro_lesson* ディレクトリが表示されます。ファイル名を *soccerball_01.hip* (または *football_01.hip*) に設定し、**Accept** をクリックして保存します。



02 ビューポートで、**C** を押して Radial メニューを表示します。このメニューから **Create > Geometry > Box** を選択します。カーソルの位置に、シーン内への配置待ちの状態にあるボックスの輪郭が表示されます。**Enter** を押して、原点の位置に配置します。

Scene View にボックスが作成され、ネットワークエディタにノードが追加され、パラメータエディタにはオブジェクトパラメータが表示されます。これらの各種インターフェース要素を使用しながら、プロジェクトを進めていきます。



03 Houdini の **View** ツールを見ていきましょう。次のホットキーを押してください。

- **タンブル** スペースバーまたは Alt (Opt) + LMB クリック + ドラッグ
- **パン** スペースバーまたは Alt (Opt) + MMB クリック + ドラッグ
- **ドリー** スペースバーまたは Alt (Opt) + RMB クリック + ドラッグ

ホームをすることで、ビューをリセットしたい場合もあります。そうした場合に便利なホットキーもあります。

- **Home Grid** スペースバー + H
- **Home All** スペースバー + A
- **Home Selected** スペースバー + G

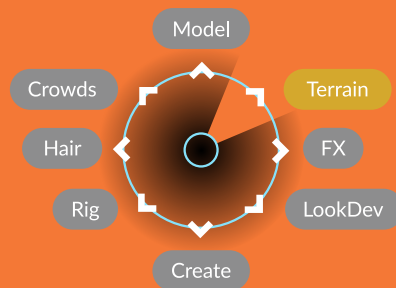


Radial メニュー

Houdini のツールには、**X**、**C**、**V** のホットキーを使ってアクセスできる Radial メニューからもアクセスできます。いずれかのホットキーを押すと、Radial メニューが表示され、各種オプションを選択できます。各メニューの主な内容は次の通りです。

スナップ
メイン (またはカスタム)
ビュー

X
C
V

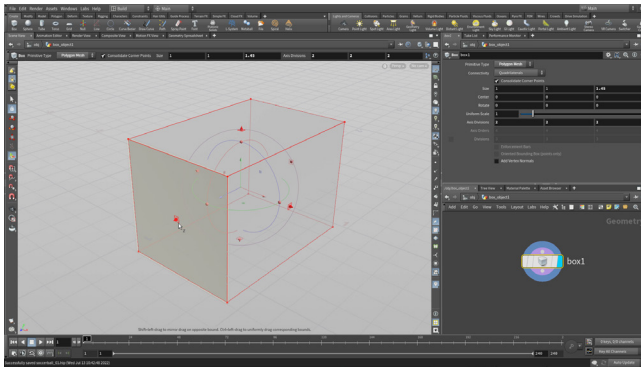
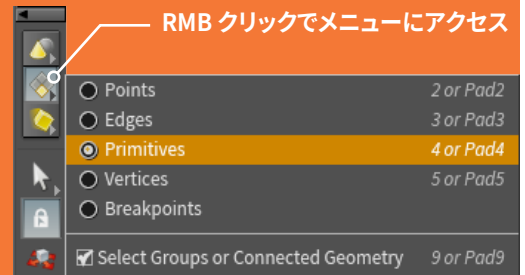




選択のホットキー

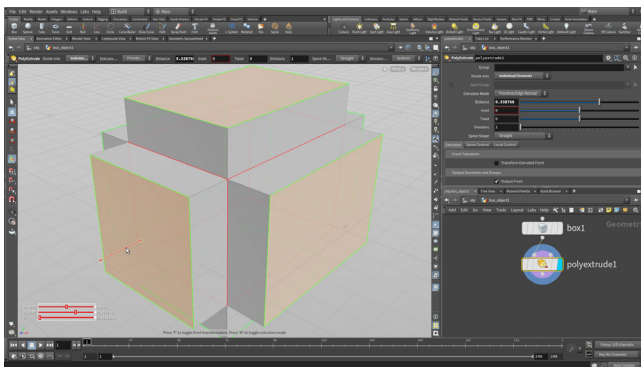
Select、Move、Rotate、Scale、Handles ツールを使用する場合、次のホットキーで選択モードと作業するレベルを決定します。

オブジェクト	オブジェクトレベル	1
ポイント	ジオメトリレベル	2
エッジ	ジオメトリレベル	3
プリミティブ/面	ジオメトリレベル	4
頂点	ジオメトリレベル	5



04 オブジェクトを選択した状態で、**I**を押し、そのジオメトリレベルに移動します。**Shift** キーを使ってハンドルをドラッグし、原点を中心に Z 軸に沿って長くします。

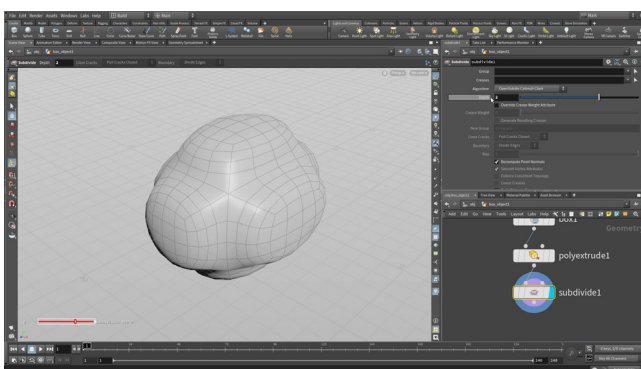
Houdini でオブジェクトを作成すると、オブジェクトのトランスフォームを管理する**オブジェクトレベル**と、形状を定義する**ジオメトリレベル**ができます。**I**を押しと、このオブジェクトのジオメトリレベルに入ることができます。また、ネットワークエディタでオブジェクトノードをダブルクリックしても、ジオメトリレベルに入れます。後でオブジェクトレベルに戻るには、**U**を押します。



05 **S**を押しして Select ツールに切り替えたら、**4**を押ししてプリミティブの選択にアクセスします。**N**を押ししてすべてを選択してから、**C**を押しして Radial メニューを表示し、**Model > Polygons > Poly Extrude**を選択します。

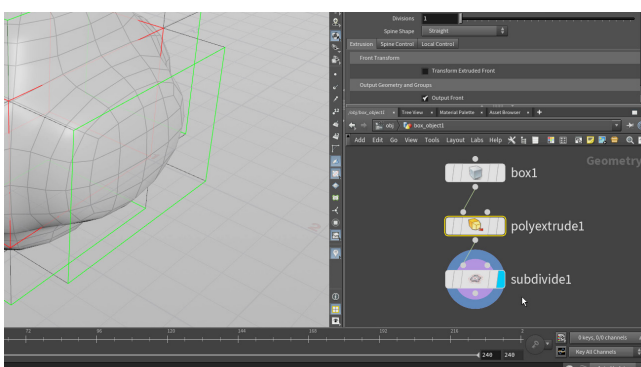
Scene View の上部にある**オペレーションコントロールツールバー**または**パラメータエディタ**で、**Divide Into**を**Individual Elements**に設定し、ハンドルを使用して**Distance**を約**0.4**に設定します。これにより、各プリミティブの法線に沿って面が押し出されます。

ネットワークビューには2つのノードが表示されています。Houdini では手順を進めるたびにノードが作成され、それらのノードでシーンを調整していきます。



06 **N**を押しして新しい面をすべて選択したら、**Tab**を押しして **sub...**と入力し、リストから**Subdivide**を選択します。Houdini のツールにアクセスするには、**Tab** キーも便利な方法です。ツール名を入力すると、リストに候補のツールが表示されるので、サブメニューをナビゲートしなくても必要なツールを簡単に見つけることができます。

パラメータエディタで、**Depth**を**2**に設定します。ジオメトリが細分化され、ポリゴンの数が増えます。Houdini には、オブジェクトレベルで細分化を表示できるオプションもあり、ジオメトリを実際に追加しなくても細分化を確認できます。しかしここでやりたいことは、ポリゴンの作成です。



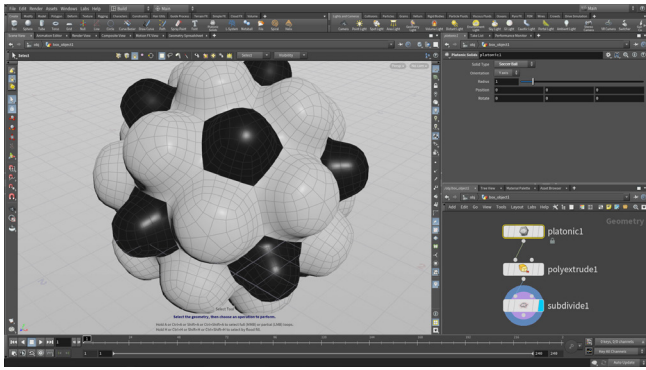
07 チェーン内のノードをいくつか選択してみましょう。選択したノードのハンドルが表示されますが、表示は最終的な形状のままです。各ノードの**Display フラグ**を設定し、表示ノードを変更してみましょう。**バイパス**や**テンプレート**といったフラグも試してください。**polyextrude**ノードをネットワークから**移動**して、その後同じ場所に戻します。

最後に、すべてを元に戻し、**subdivide** ノードの**Display フラグ**を設定します。これは非常に重要です。Display フラグによって、オブジェクトレベルでの表示が決まるからです。**Display フラグは必ず正しく設定してください!**

パート2

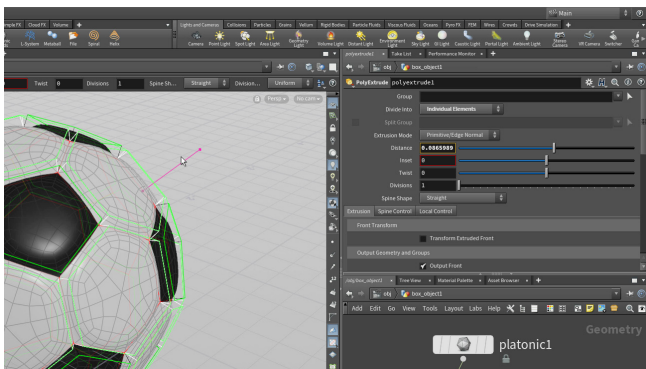
サッカーボールの作成

次は、ボックスをサッカーボールの形をしたプラトン立体に置き換えます。Houdiniのプロシージャルなアプローチなら、Box ノードを Platonic Solids ノードに置き換えられます。そこから他のノードを調整して、サッカーボールらしい見た目になります。入力ノードの置き換えができるため、シンプルなジオメトリでネットワークのプロトタイプを作成しておけば、柔軟性が高まります。



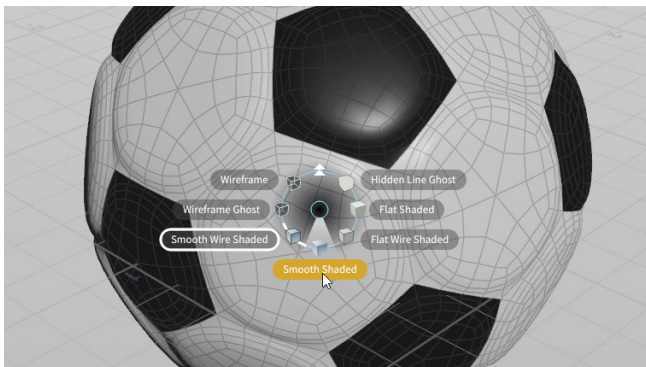
01 ネットワークエディタで、Tab キーを使用して Platonic Solids ノードをネットワークに追加します。クリックしてチェーンの最上部付近に配置します。platic ノードを polyextrude に接続します。パラメータエディタで、Solid Type を Soccer Ball に設定します。box ノードを選択して削除します。

Houdini はプロシージャルのため、入力ノードを変えても、ネットワーク全体が適切に機能することは珍しくありません。作業の柔軟性が高まるだけでなく、変更後の結果が気に入らなければいつでも、元の形状を接続し直せばよいわけです。



02 polyextrude ノードを選択します。Handle ツールがアクティブになっていることを確認したら、ビューポートでハンドルを使用して、Distance の値を下げます。パラメータエディタでパラメータ値を設定してもかまいません。これでサッカーボールらしい見た目になりました。subdivide ノードを表示している場合でも、polyextrude ノードを選択すると、そのハンドルとパラメータにアクセスできることを覚えておきましょう。

このプリミティブタイプで準備がすべて整ったと思うかもしれませんが、実際には、平らな面で構成された切頂正二十面体にすぎません。丸いサッカーボールが必要なので、もう少し手を加える必要があります。



03 ビューポートで V を押し、Radial メニューで Shading > Smooth Shaded を選択します。ビューポートの右上のメニューを使用しても、シェーディングを変更できます。

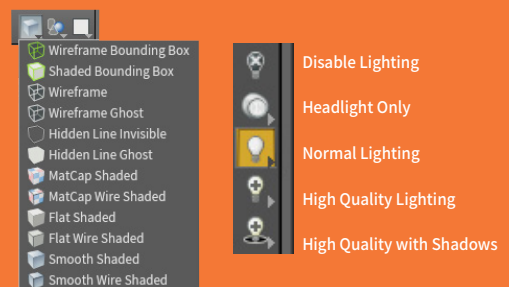
このサッカーボールは、正式な革製のサッカーボールではなく、安いビニールボールのように見えます。これからノードを追加し、分岐させながら見た目を向上させていきます。

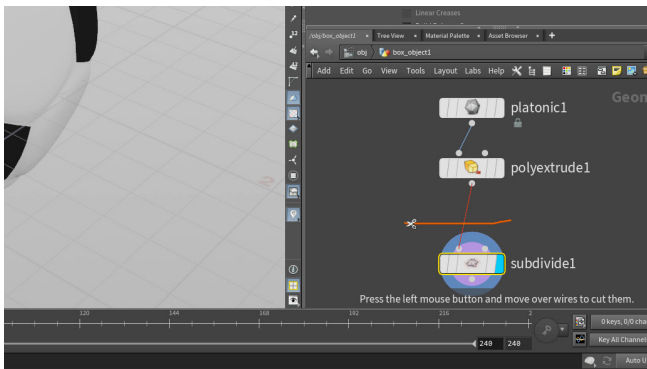
分析を終えたら、シェーディングの設定を Smooth Wire Shaded に戻します。



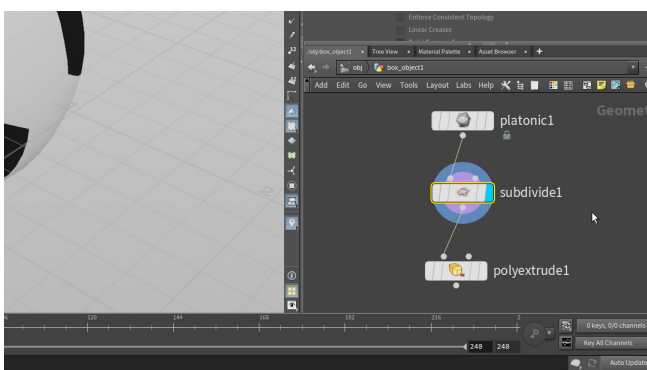
シェーディングオプション

View Radial メニューまたはビューポートの右上にある Shading メニューから、さまざまなシェーディングオプションを使用できます。オブジェクトのシェーディングでは、ビューポートの右端にある表示オプションでライティングが決まります。ヘッドライト、通常のライティング、シャドウ付きの高品質なライティングのいずれかを選択できます。シェーディングビューからワイヤーフレームに素早く切り替えるには、W キーを押します。

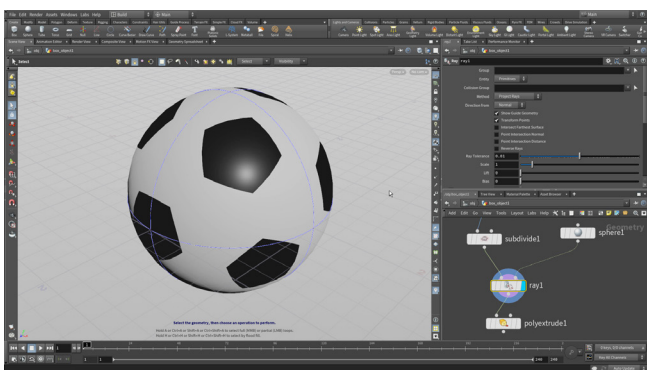




04 ネットワークビューで **Y** を押し、**subdivide** ノードと **polyextrude** ノードをつなぐワイヤーを横切るようにドラッグして、接続を解除します。より丸みのあるサッカーボールになるように、**subdivide** を残り2つのノードの間に移動します。

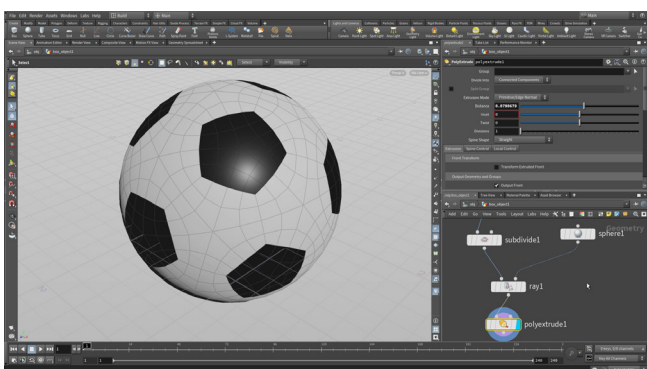


05 **platonic** ノードと **polyextrude** ノードの間に、**subdivide** ノードをドラッグします。接続されたワイヤー上にドロップすると、自動的に挿入されます。挿入されない場合は、接続されるまで少し動かしてみてください。押し出される前に、球のディテールが増加します。



06 ネットワークエディタで **Tab** キーを使用して **Ray** ノードを追加し、**subdivide** の後に接続します。次に、**sphere** ノードをネットワークに追加して、**Radius** を **1, 1, 1** に、**Primitive Type** を **Primitive** に設定します。sphere を **ray** ノードの2つ目の入力に接続します。これにより、細分化されたボールが完全な球に投影されます。

これは、Houdini の非常に強力なノードです。あるジオメトリのポイントを別のジオメトリに投影します。細分化されたサッカーボールが完全には丸くないという、ここでの問題にぴったりの解決策です。



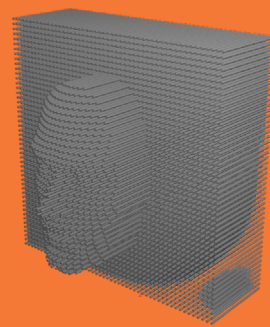
07 **polyextrude** ノードに **Display** フラグを再度設定します。**Divide Into** を **Individual Elements** に設定すると、すべての小さいポゴンが押し出されますが、それは望ましくありません。**Connected Components** に設定して、すべてのポリゴンが押し出されるようにします。このネットワークでは、ボールが細分化された後で、サッカーボールの元のパッチを押し出す方法が必要です。これを行うには、元のジオメトリのプリミティブ番号を使用します。



Ray ノード

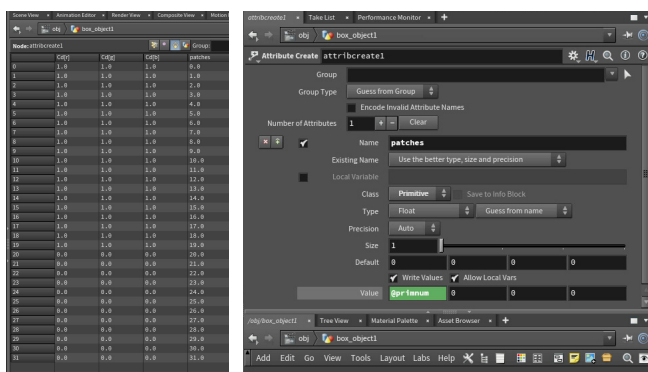
Ray ノードは、ポイントを別のジオメトリに投影するツールです。おもちゃのピンアートに似ています(子供の頃に遊んだことはありませんか)。実際、これは Houdini でピンアートのボードをセットアップするために使用するノードです。

ヘルプの確認 | 各ノードについて詳しく知りたい場合は、パラメータエディタの右上にある **?** ヘルプボタンをクリックすると、そのノードのオンラインドキュメントが表示されます。シェルフのツール上にカーソルを置いて、**F1** を押しでもヘルプを確認できます。Houdini で開けるサンプルファイルが用意され、そのノードの機能を学べるようになっているヘルプもたくさんあります。

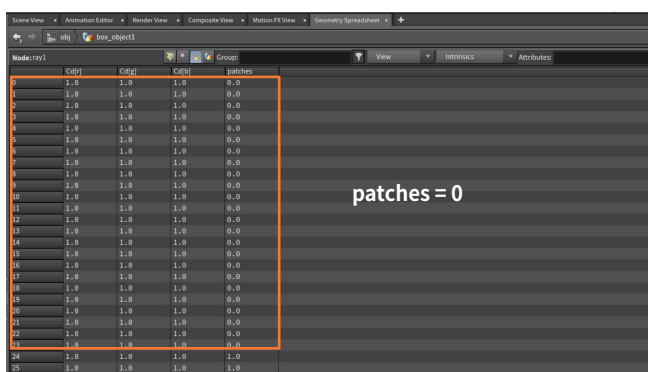


パート3 For Each ノード

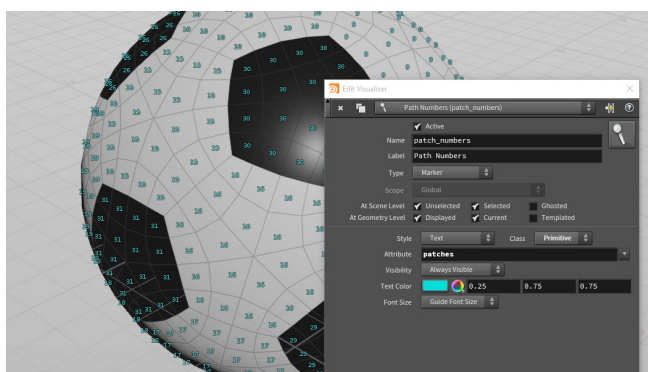
ここで魔法が働きます。最後に作成したアトリビュートを For-Each ループに送り、それぞれの元のパッチに多数のポリゴンが含まれていても、それらのパッチが押し出されるようにします。ポリゴンを押し出した後、もう一度細分化すると、サッカーボールが革製らしく見えるようになります。



01 Attribute Create ノードを **platic** ノードの後に追加します。Name を **patches** に、Class を **Primitive** に設定します。Value フィールドの最初の値を **@primnum** に設定します。このエクプレッションは、プリミティブ番号アトリビュートを取得し、それを **patches** という新しいアトリビュートに変換します。



02 **attributecreate** ノードを選択した状態で、メインのビューポートの横にある **Geometry Spreadsheet** タブをクリックします。**Primitive** ボタンをクリックすると、左側にプリミティブ番号が表示され、パッチの色を示す3つのカラーアトリビュートと、プリミティブ番号と一致する **patches** アトリビュートも表示されます。**ray** ノードをクリックします。このアトリビュートは、形状が細分化されるときに繰り越されます。もっと多くのプリミティブがあることが分かりますが、**patches** アトリビュートは 31 までで、その後は 0 に戻ります。

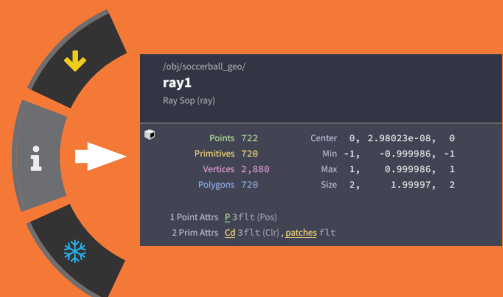


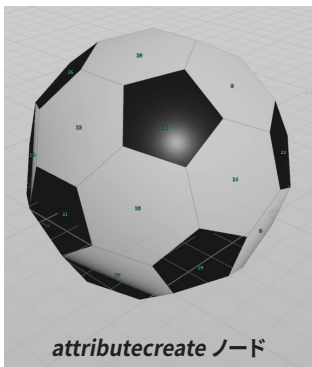
03 Scene View タブに戻り、**Display Options** バーの **Visualization** ボタンを **RMB** クリックし、**Scene** の横にある **(+)** 記号をクリックして、**Marker** を選択します。Edit Visualizer パネルで、**Name** と **Label** を **Patch Numbers** に、**Type** を **Marker** に、**Class** を **Primitive** に、そして **Attribute** を **patches** に設定します。



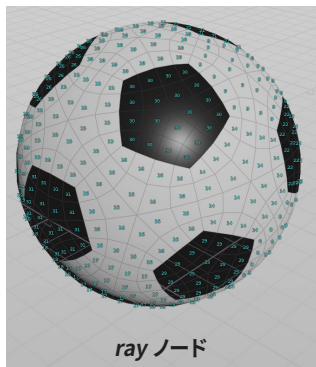
アトリビュートの使用

アトリビュートは、ポイント、プリミティブ、頂点に割り当てることができます。一般的なアトリビュートには、**カラー (Cd)** や **UV** などがあります。ノードにカーソルを置いて Radial メニューから **i** を選択すると、チェーン内の任意の場所でアトリビュートを確認できます。また、**Geometry Spreadsheet** パネルでもアトリビュート値を確認することができます。このレッスンでは、**patches** というカスタムアトリビュートを作成し、For-Each ループで活躍してもらいます。





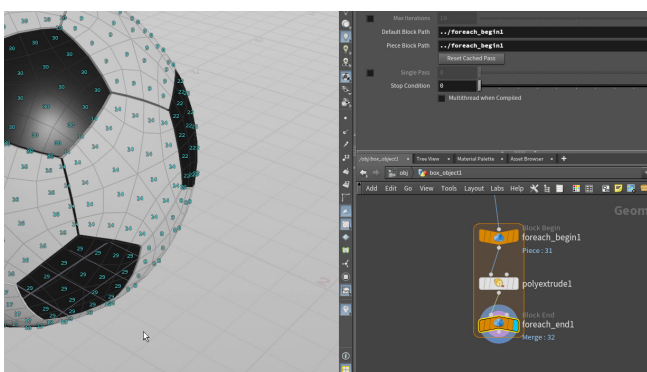
attributecreate ノード



ray ノード

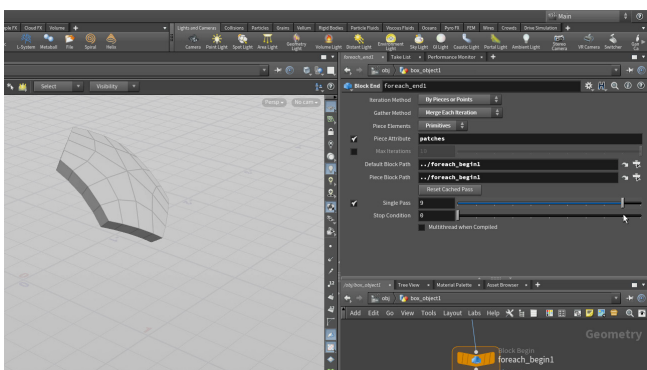
04 **Visualization** ボタンがオンになっていることを確認します。ビューポートのサッカーボール上にパッチの値が表示されています。元のプラトン立体からのプリミティブ番号が、細分化された面に転送されています。**Display フラグ**を別のノードに設定して、関係を確認してみましょう。この情報は、For-Each ループを使用して適切にパッチを押し出すために使用されます。

Visualization ボタンを**オフ**にして、作業内容を**保存**します。ここまでの手順について、やや抽象的で専門的すぎると感じているかもしれませんが、大丈夫です。努力はもうすぐ実を結びます。



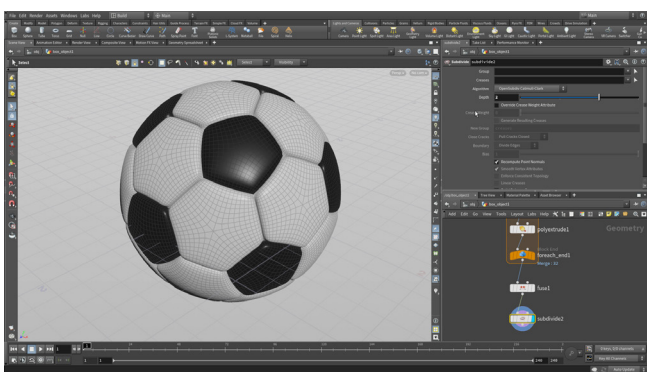
05 ネットワークエディタで **Tab** を押し、**For-each Named Primitive** と入力して、2つのノードにアクセスしてシーンに配置します。**ray** ノードと **polyextrude** ノードの間に **foreach_begin** を接続し、**polyextrude** ノードの後に **foreach_end** を接続します。**foreach_end** ノードを選択し、パラメータエディタで **Piece Elements** は **Primitives** に設定したままにし、**Piece Attribute** を **patches** に設定します。**foreach_end** ノードに **Display** フラグを設定します。

patches アトリビュートに基づいて、元のパッチが一斉に押し出されているのを確認できるはずですが、押し出されていない場合は、**polyextrude** ノードで **Divide to** が **Connected Components** に設定されていることを確認してください。



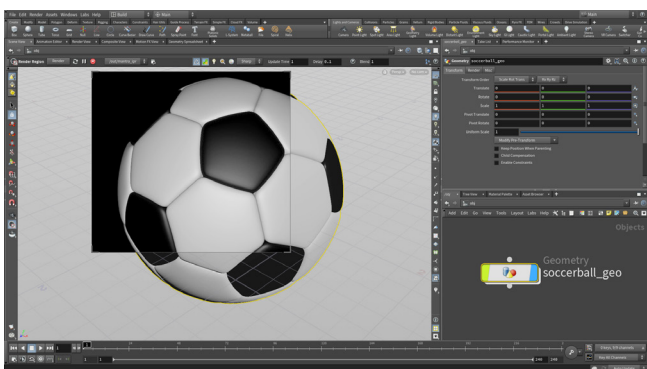
06 **Single Pass** の横にあるチェックボックスをクリックし、どうなるかを確認します。スライダをドラッグして、それぞれのパッチが個別に押し出される様子を確認してください。10 よりも大きい値に設定すると、さらに多くのパッチを確認できます。

Single Pass を**オフ**にして、完全な形状を表示します。**for-each** ノードがすべてのパッチを作成し、最終的なジオメトリを返します。For-Each ループは、Houdini で頻繁に使用する強力なノードセットです。



07 **foreach_end** ノードの後に **Fuse** ノードを追加し、その **Display フラグ**を設定します。これは、ピースをつなげて単一のトポロジにするノードです。**for-each** ノードでは、バラバラのパッチに分割しましたが、それらを再結合することはしませんでした。

Fuseの後に **Subdivide** ノードを追加します。**depth** を **2** に設定します。ビューポートでディテールが加わり、これを使用してモデルを評価できるようになります。ポリゴンの数は増えますが、まだ真の細分化サーフェスとしてレンダリングされるわけではありません。Houdini では、ジオメトリを追加することなく、ビューポートで細分化表示を設定することも可能ですが、この Subdivide ノードはレッスンの後半で必要となります。



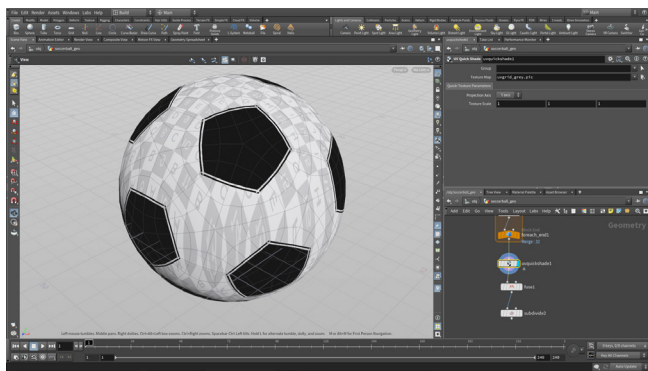
08 サッカーボールが革製に見えるように、**polyextrude** の値を変えてみましょう。ここでは **Distance** を **0.1**、**Inset** を **-0.02** に設定してください。パッチが丸みを帯びて、ルックが向上します。

オブジェクトレベルに移動し、パラメータエディタでオブジェクトの名前を **soccerball_geo** に変更します。ノードを選択するか、**F2** を押すか、名前を**ダブルクリック**します。**Render** タブをクリックし、**Render Polygons as Subdivisions (Mantra)** をオンにして、レンダリング時の真の細分化をセットアップします。**Render Region** ツールを選択したら、ビューポートでサッカーボールの周りにボックスを描き、プレビューレンダリングを作成します。キャンセルするには、その領域の右上にある **X ボタン** をクリックします。

パート4

UV のセットアップ

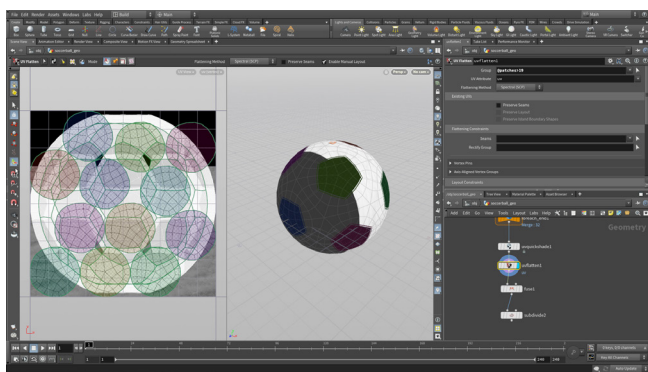
マテリアルやテクスチャをセットアップするには、オブジェクトに適切な UV がセットアップされていることが大切です。Houdini のジオメトリには UV が付属していないため、自分で作成しなければなりません。つまり、ネットワークにノードを追加する必要があり、ここでは UV Quickshade と UV Flatten ノードを追加します。



01 `soccerball_geo` オブジェクトの中に入ります。ネットワークビューで、Tab > **UV Quickshade** を押し、`foreach_end` ノードのすぐ後に新しいノードを追加します。**Display フラグ**を設定します。

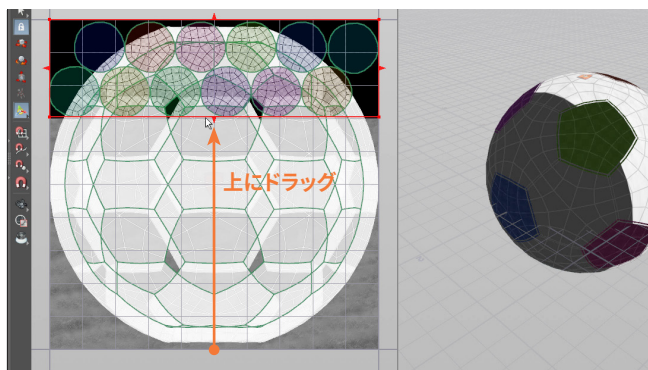
Texture Map の横にある**ファイル選択**ボタンをクリックします。/tex フォルダに移動し、`soccerball_color.rat` を選択します。このテクスチャマップがジオメトリに表示されますが、投影方法を使用して UV を作成したため、全体が引き伸ばされたように見えます。

注：既存のテクスチャに合わせて UV をセットアップするのは、通常の作業順ではありません。逆の順番をあえて使っているのは、自分でテクスチャをペイントする必要がないからです。



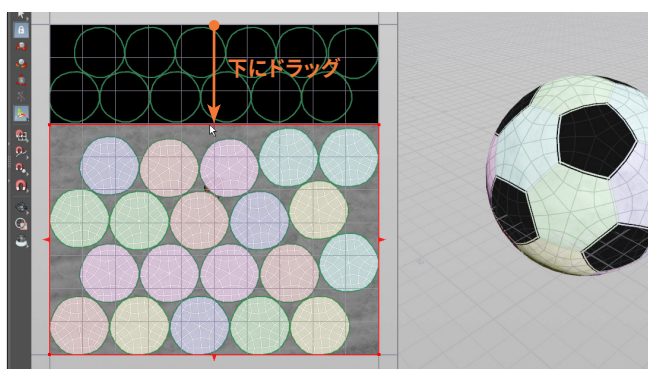
02 Scene View で、**N** を押してすべてを選択してから、Tab > **UV Flatten** を選択します。Group フィールドで、`@patches>19` というエクスプレッションを入力します。これにより、パッチ境界を使用してサッカーボールジオメトリの暗いパッチが平坦化され、UV がレイアウトされます。

このツールを使用すると、UV ビューが表示されます。右上の **UV (vertex)** メニューをクリックして、**Background > soccerball_color.rat** を選択します。すると UV パネルの背景にこのテクスチャが表示されます。このテクチャには、画像の中央にチームのロゴが、上部には暗いパッチのための暗い領域が含まれています。



03 Handle ツールを **RMB** クリックし、**Min:** ハンドルをオンにします。下部にある矢印ハンドルを、すべてのパッチがテクスチャマップ上部の暗い領域内に入るまで引き上げます。

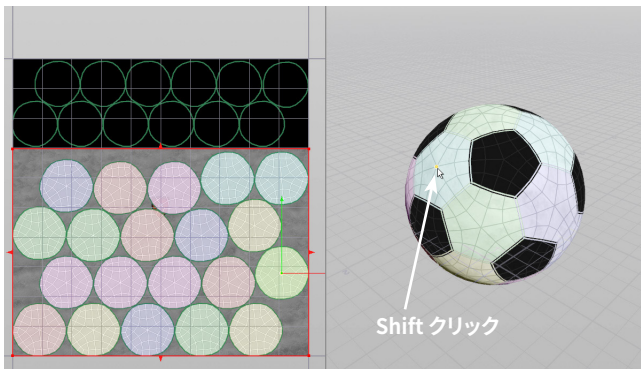
このノードを `fuse` または `subdivide` の後に配置していたら、使用する境界はなかったでしょう。Houdini では既存のネットワークの途中で UV をセットアップでき、作業に柔軟性を持たせられます。



04 `uvflatten` ノードの後、`Fuse` ノードの前に、もう1つ **UV Flatten** ノードを追加します。Group フィールドで、`@patches<20` というエクスプレッションを入力します。これにより、明るいパッチが平坦化されます。

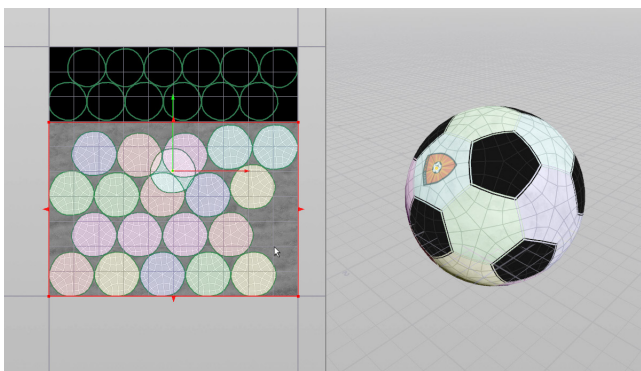
Handle ツールを **RMB** クリックし、**Min:** ハンドルをオンにします。上部にある矢印ハンドルを、すべてのパッチがテクスチャマップ下部の明るい領域内に入るまで引き下げます。

完了したら、Handle ツールを **RMB** クリックし、**Min:** ハンドルをオフにします。Scene View で **RMB** クリックし、メニューで **Texture Visualization > Off** を選択します。これでグリッドが非表示になります。



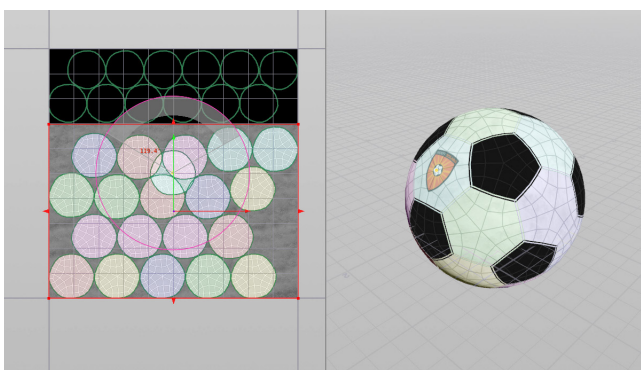
05 uvflatten ノードを選択し、**Handle** ツールをクリックします。ジオメトリ上にカーソルを置くと、パッチがハイライトされるのが分かります。ビューポート上部にある**オペレーションコントロール**ツールバーの**Pin Vertices** ボタンをクリックします。

この画像に表示されたパッチにカーソルを置き、3D ビューで、目的のパッチの中央の頂点を **Shift クリック** して選択します。UV ビューに、ピンと操作用のハンドルが追加されます。

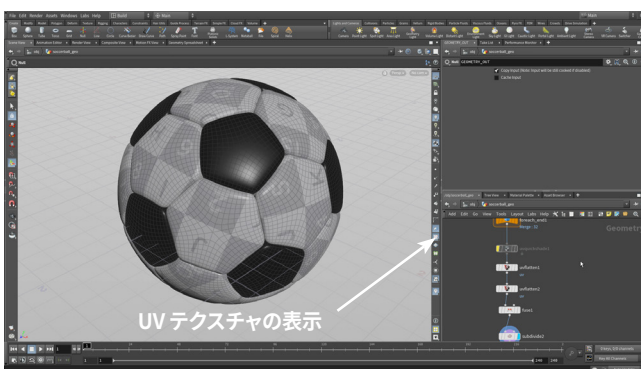


06 UV ビューに**移動**し、ピンとパッチをロゴの中央になるように移動します。**Y** を押し、回転ハンドルに切り替え、パッチを回転してロゴの位置を調整します。

UV ビューでパッチの頂点をピン留めすると、位置を固定できます。最初は隣接するパッチとのオーバーラップがいくらかありますが、再パックによって簡単に修正できます。



07 オペレーションコントロールツールバーの **Repack** ボタンをクリックして、新しいパッチを中心に残りのパッチを再編成します。引き続きピンを使用してパッチを動かしますが、UV のオーバーラップを回避するには、もう一度**再パック**が必要です。



08 quickshade ノードを**バイパス**に設定し、テクスチャマップの割り当てを非表示にします。このノードが必要なのは、UV をセットアップするときのみです。サッカーボールに UV グリッドが表示されます。

パースビューで UV を非表示にするため、**Display Options** バーで **Show UV Texture when UV's Present** ボタンを**オフ**にします。

Null ノードをチェーンの終端に追加して、名前を **GEOMETRY_OUT** に変更します。ネットワークチェーンの終端の定義には、この種のノードを使うことをお勧めします。作業内容を**保存**します。



UV FLATTEN

UV Flatten ノードの動作は、2つの工程で構成されています。シーム(継ぎ目)によって定義された、個々のテクスチャのピースを受け取り、それらを 2D テクスチャ空間へと平坦化して、ポリゴンサイズの均等化を試みます。

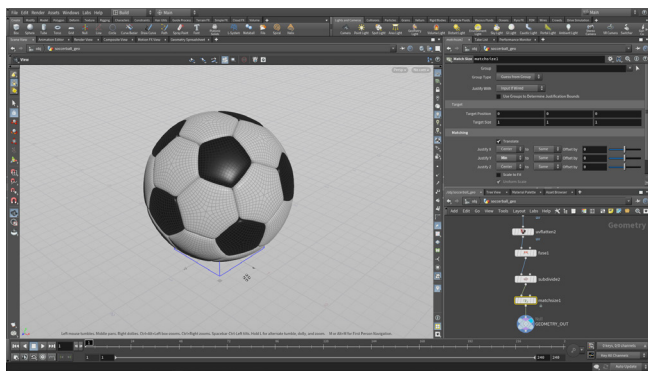
このノードで、平坦化アルゴリズム用の拘束を追加することができます。この拘束により、レイアウトアルゴリズムが強制的に特定の条件を満たすようにすることで、最終的な UV レイアウトをさらに制御できるようになります。このノードの状態、ツールを使って拘束を指定すれば、ノードをインタラクティブに使用できます。または、**Enable Manual Layout** をオフにすると、このノードをプロシージャルに使用できるようになります。



パート5

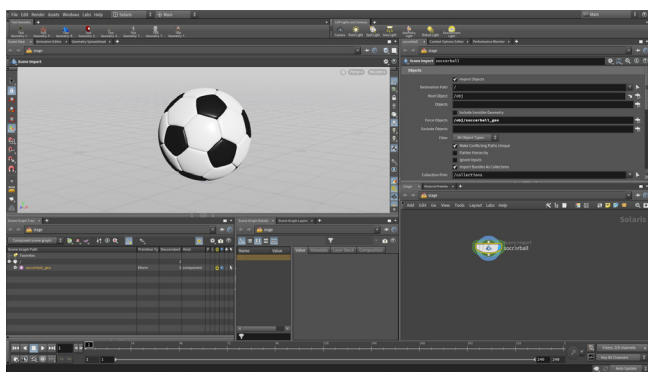
レイアウト : カメラとライト

レンダリング用のシーンを作成するには、ジオメトリを Houdini の Solaris または LOP コンテキストに取り込むことになります。これは、ルックデブ、レイアウト、ライティングに特化した環境で、USD (Universal Scene Description) を基盤に構築されています。これを使用すると、Solaris ワークフローの一部として、Scene View で適切に動作する Karma レンダラでレンダリングすることができます。



01 ネットワークビューで、**Tab > Match Size** を押し、**subdivide** と **GEOMETRY_OUT null** の間にこのノードを追加します。**GEOMETRY_OUT** ノードに **Display フラグ** を設定します。

matchsize ノードを選択し、**Justify Y** を **Min** に設定して、地上に乗るようにボールを引き上げます。これで、正しい位置でレンダリングできます。

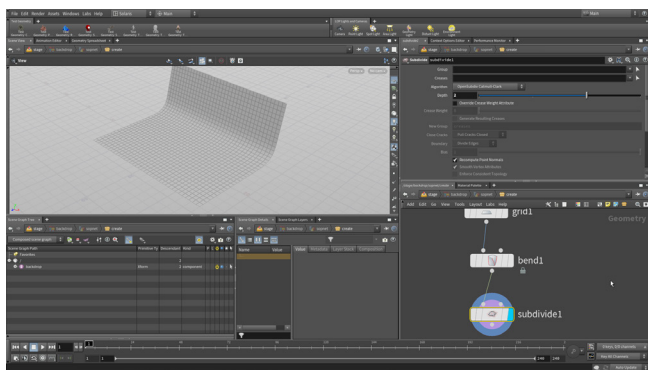


02 デスクトップを **Solaris** に変更します。パスバーに **stage** と表示されていることを確認してください。

ネットワークビューで **Tab > Scene Import** を選択し、クリックしてノードを配置します。**Force Objects** フィールドの横にある **ノードセレクト** ボタンをクリックし、ポップアップウィンドウで **soccerball_geo** オブジェクトを選択してから **Accept Pattern** をクリックします。

Scene View で、**スペースバー + H** など、ビューのリセット用ツールを使用して、サッカーボールがよく見えるようにします。

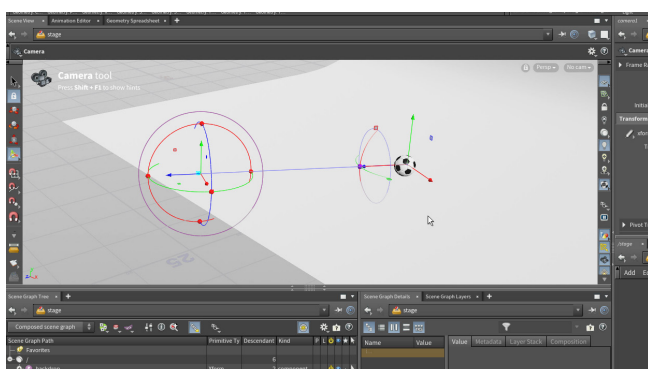
注 : Objects フィールドではなく Force Objects を使用します。Force Objects は、オブジェクトの Display フラグがオフになっていても、LOP にオブジェクトを取り込むからです。



03 ネットワークビューで、**Tab** を押して **Grid** と入力します。クリックしてノードを配置し、名前を **backdrop** に変更します。**backdrop** ノードをダブルクリックして、ジオメトリレベルに入ります。

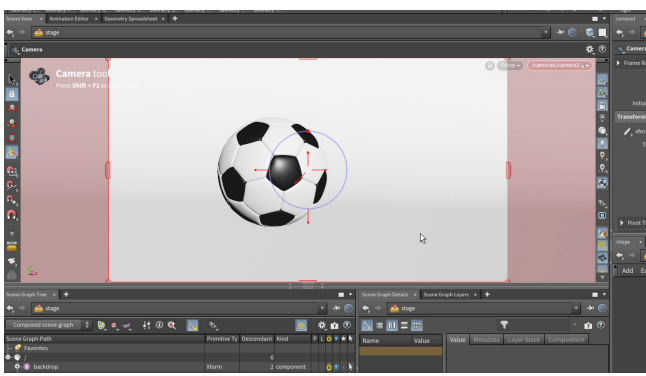
grid ノードを選択し、**Size** を **80, 80**、**Center** を **0, 0, -20** に設定します。**grid** ノードの出力を **RMB クリック** して、**Bend** と入力します。クリックしてノードを配置したら、**Display フラグ** を設定します。**Bend** を **75**、**Capture** セクションで **Capture Origin** を **0, 0, -30**、**Capture Direction** を **0, 0, -1**、**Capture Length** を **5** に設定します。

grid ノードの出力を **RMB クリック** して、**Subdivide** と入力します。**Display フラグ** を設定し、**Depth** を **2** に設定します。



04 ステージレベルに戻ります。**backdrop** ノードを **sceneimport** に接続します。**sceneimport** の出力を **RMB クリック** し、**Camera** と入力します。**Enter** を押してノードを配置したら、その **Display フラグ** を設定します。

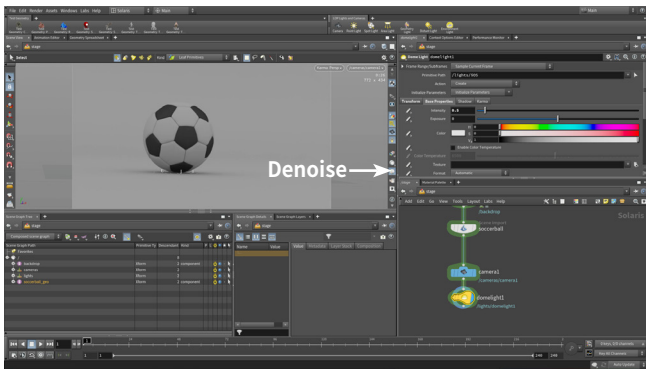
Scene View で、原点にカメラハンドルが表示されます。ズームアウトしてシーンを見下ろすようにしたら、カメラが左側からサッカーボールを見るようにハンドルを調整します。地面に沿ってハンドルを動かしたい場合には、**コンストラクション平面** をアクティブにすると操作がしやすくなります。また、軸ハンドルで方向を制御したり、カメラを地面から持ち上げることも可能です。



05 Scene View の右上の **No cam** メニューをクリックし、**camera1** を選択します。これでカメラ越しに見えるようになり、見え方を調整できます。

求めているのはこのビューではないでしょう。いくら変更します。Scene View の右側にある **Lock camera to view** ボタンをクリックします。**表示ツール (スペースバー + LMB/MMB/RMB)** を使用してカメラの位置を調整します。

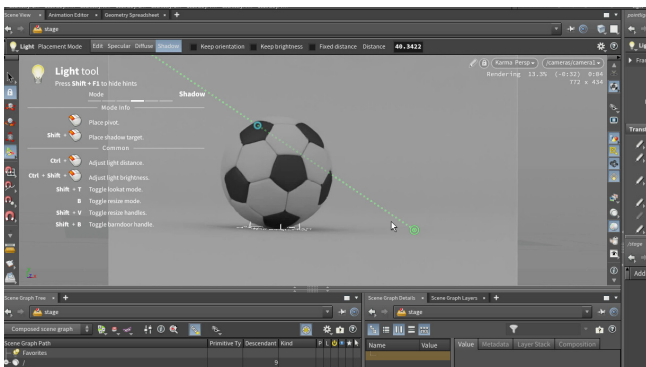
重要 : 完了したら、**Lock Camera** ボタンをオフにします。



06 LOP Lights and Camera シェルフで **Environment Light** ツールをクリックし、**Enter** を押してそれを原点に配置します。**Intensity** を **0.5** に設定し、少し下げます。

カメラメニューの左にあるメニューをクリックし、**Karma** に設定します。これで、ビューポートで Karma レンダラが使用されるようになります。

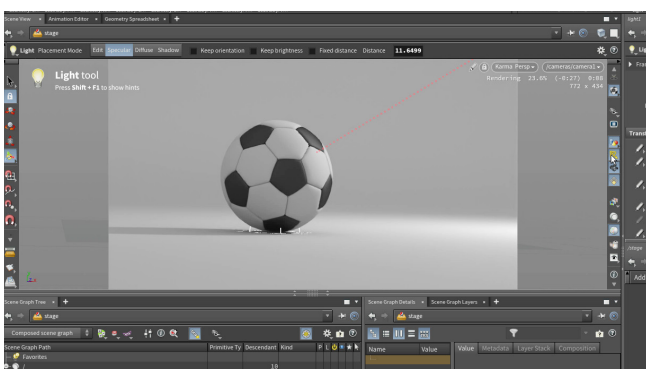
Nvidia グラフィックカードがあり、最新のドライバをインストールしている場合は、Optix Denoiser をオンにすると、画像をより素早く処理できます。**Display Options** バーでそれをオンにするか、**D** を押して **Render Display Options** の **Enable Denoising** を設定します。



07 LOP Lights and Camera シェルフで **Point Light** ツールをクリックし、**Enter** を押してそれを原点に配置します。ライト越しに見えるようになります。カメラを再度 **camera1** に設定します。

ノードをアクティブにした状態で、**Shift + F** を押して **Shadow** モードをオンにします。**オペレーションコントロール** ツールバーでクリックしても同じです。サッカーボールの上部をクリックしてピボットポイントを設定したら、**Shift** クリックして地面にターゲットを配置します。**Ctrl** ドラッグでライトの距離を設定します。

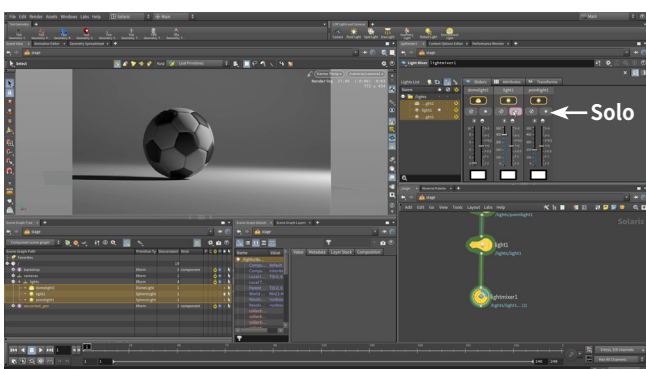
Ctrl + Shift ドラッグで、ライトの強度を変更できます。サッカーボールですぐに見て取れるほどの影響を与えるには、かなり高く設定する必要があります。



08 ネットワークビューで、**pointlights** ノードの出力を **RMB** クリックし、**Light** と入力してから **Enter** を押してノードを配置し、その **Display フラグ** を設定します。

ノードをアクティブにした状態で、**Shift + S** を押して **Specular** モードをオンにします。サッカーボールの右側をクリックして、鏡面反射が集中する領域を定義します。

Ctrl ドラッグでライトをサッカーボールから離し、**Ctrl + Shift** ドラッグでライトの強度を変更します。



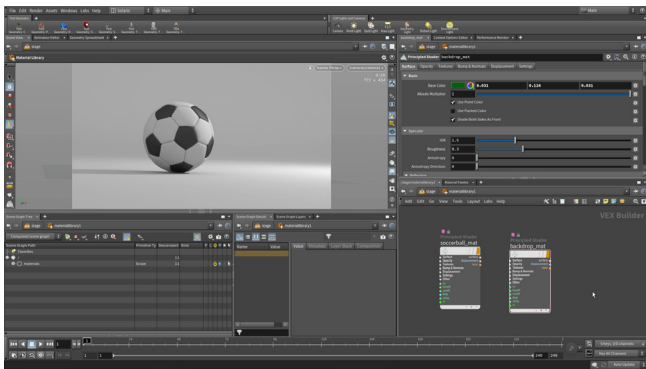
09 ネットワークビューで、**lights** ノードの出力を **RMB** クリックし、**Light Mixer** と入力してから **Enter** を押してノードを配置し、その **Display フラグ** を設定します。パラメータエディタに特殊なパネルが作成され、左側にライトがリストされます。

3つのライトをリストから右側の領域に**ドラッグ**します。**星** アイコンをクリックして各ライトを**ソロ**にして寄与度を定めたら、**Exposure** でライティングを微調整します。**intensities** がとても高いので、強度バーの上のアイコンをクリックし、ポップアップメニューでショットに適した **Max Value** を設定します。完了したら、**Solo** ボタンを**オフ**にしてすべてのライトを表示します。

パート6

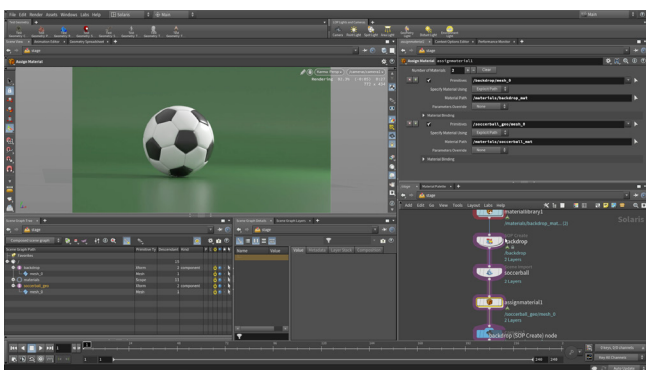
ルックデブ: マテリアル

マテリアルとシェーダは、LOP/Solaris のコンテキスト内で作成することもできます。この場合、シーングラフにマテリアルを追加してから、ジオメトリに割り当てます。マテリアルは、Material Library ノード内で作成され、LOP/Solaris レベルで割り当てられます。背景にテクスチャを追加するには、UV を作成して、マップを適切に配置できるようにする必要があります。



01 ネットワークビューで **Tab > Material Library** を押し、既存のノードのネットワークのすぐ上にノードを配置したら、それを **backdrop** ノードに接続します。ノードを **ダブルクリック** して、**VEX Builder** レベルに下がります。

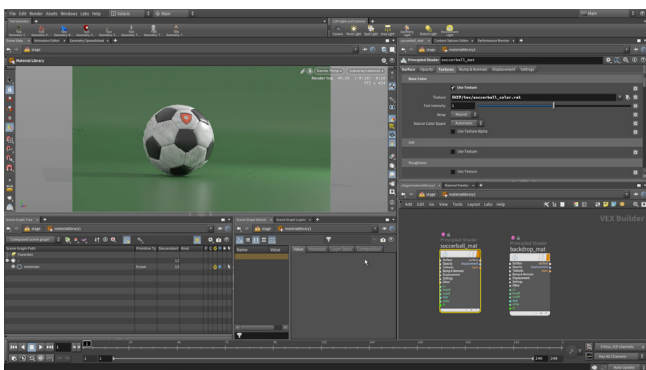
Tab > Principled Shader を押してノードを配置します。その名前を **soccerball_mat** に変更します。 **Base Color** を **白(1, 1, 1)** に設定します。このノードを **Alt ドラッグ** して、2つ目の Principled Shader を作成し、名前を **backdrop_mat** に変更します。 **Base Color** を暗い緑色に変更します。



02 ステージレベルに戻り、**Tab > Assign Material** を押して、**sceneimport** と **camera** ノードの間にノードを配置します。

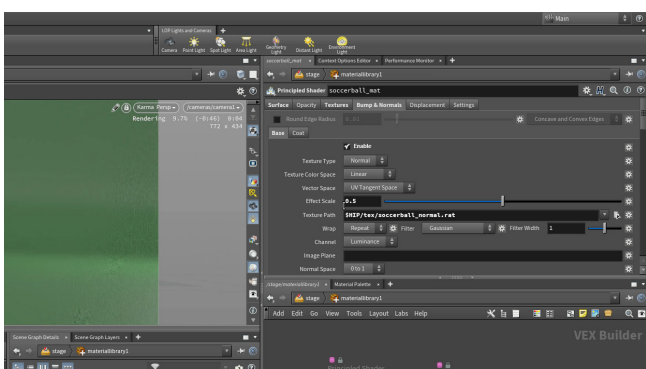
パラメータエディタで、**Primitives** の横の矢印ボタンをクリックし、ビューポートで **soccerball** ジオメトリを選択します。 **Enter** を押し、**Primitives** フィールドにそのパスを追加します。 **Material Path** の横にある矢印をクリックし、ポップアップウィンドウで **materials > soccerball_mat** を選択してから **OK** をクリックします。

+**(プラス)ボタン** をクリックし、**backdrop** と **backdrop_mat** についてもこの手順を繰り返します。



03 **materiallibrary** ノードをダブルクリックして中に入り、**soccerball_mat** ノードを選択します。 **Textures** タブをクリックし、**Base Color** の **Use Texture** をクリックしたら、**Texture** の横のボタンを使用してファイルウィンドウを開きます。横のリストで **\$HIP** をクリックしたら、**tex** フォルダをクリックして開き、**soccerball_color.rat** をワンクリックして選択します。 **Accept** をクリックし、テクスチャをマテリアルに割り当てます。

\$HIP 参照により、シーンファイルの位置に対して相対的に参照ようになります。このようにすると、プロジェクトディレクトリを別のコンピュータに移動しても、引き続き参照が機能します。



04 **Textures** タブで、前の手順で学んだテクニックを使用して、**Roughness** と **Reflectivity** にテクスチャを割り当てます。 **tex** フォルダで適切なテクスチャを見つけます。

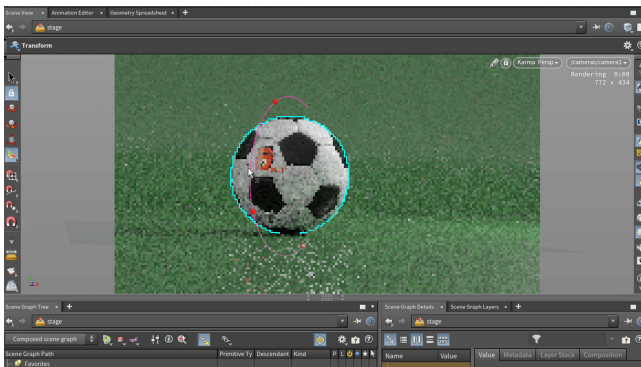
マテリアルの **Bumps & Normals** タブに移動して、**Enable** ボタンをクリックします。 **Texture Path** の横にある矢印をクリックして、**tex** ディレクトリから **soccerball_normal.rat** ファイルを選択します。 **Effect Scale** を約 **0.5** に設定し、どうなるかを確認します。



Houdini のマテリアル

Houdini のマテリアルは **VEX Builder** コンテキストにあり、この場合は **Material Library** ノード内にネストされます。マテリアルは、マテリアルの品質を定義する **VOP** ノードまたは **Material X** ノードで構成されています。

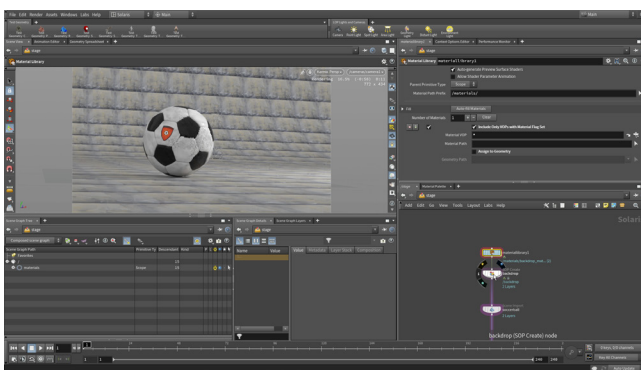
Principled Shader は、単体でテクスチャマップを割り当て、多様なルックを生成できる優れたマテリアルです。独自のシェーダやマテリアルを構築して、より高度なルックを実現することも可能です。



05 Scene View で、**camera1** 越しにショットを確認します。サッカーボールのロゴがカメラに対してよい位置にないので、ボールを回転する必要があります。

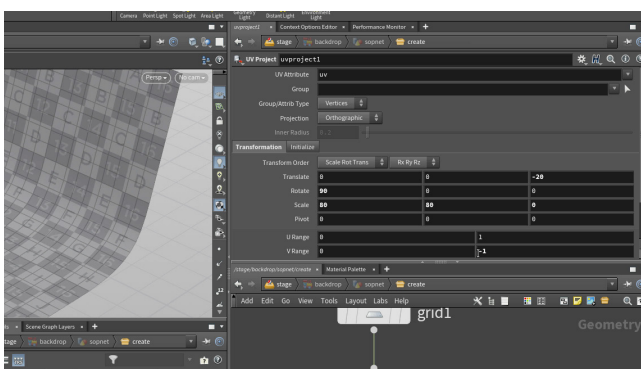
Scene View で、サッカーボールを選択します。**Tab > transform** を押し、チェーンの終端に transform ノードが追加されます。**Handle** ツールで、**R** を押し **Rotate** ハンドルにしたら、ロゴがきれいに見えるようにボールを回転します。タンブルする必要がある場合もあります。

これは、**Karma** または **Houdini GL** ビューで行うことができます。変更が気に入らない場合は、ノードを削除してもう一度試します。**soccerball_geo** を選択解除するには、**シーングラフ** でそれを **Ctrl** クリックします。



06 backdrop マテリアルにいくつかテクスチャマップを追加しましょう。**materiallibrary** ノードを **ダブルクリック** して、**backdrop_mat** ノードを選択します。

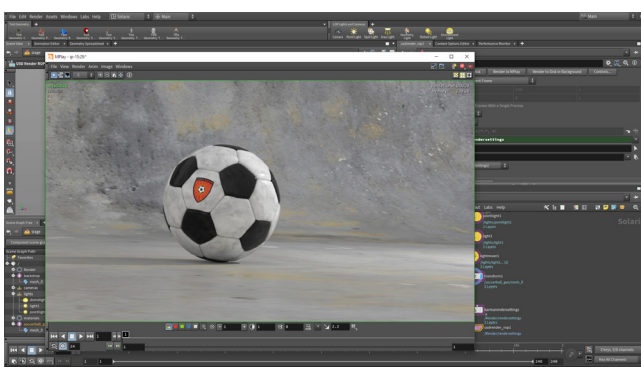
Base Color を **1, 1, 1** に設定し、この色がテクスチャマップに乗算されるようにします。**Textures** タブをクリックして、**Base Color** の **Use Texture** をクリックします。**ファイル選択** ボタンをクリックして、**\$HIP** から **/tex** ディレクトリに移動し、**backdrop_color.rat** を選択します。**Reflectivity** に **backdrop_reflect.rat** テクスチャを追加してもよいでしょう。**Scene View** を見ると、UV が適切にセットアップされておらず、テクスチャマップも機能していません。



07 ステージレベルに戻り、**backdrop** ノードを **ダブルクリック** してジオメトリレベルに移動します。**grid** と **bend** の間に **UV Project** ノードを追加します。サーフェスを曲げる前に UV が作成されるよう、ここにノードを追加していきます。

uvproject ノードで、**Initialize** タブをクリックして **Initialize** ボタンを押します。**Transformation** タブに戻り、**V Range** を **0, -1** に設定します。これにより、UV の方向が適切になります。

重要 : **subdivide** ノードに **Display フラグ** を再度設定します。



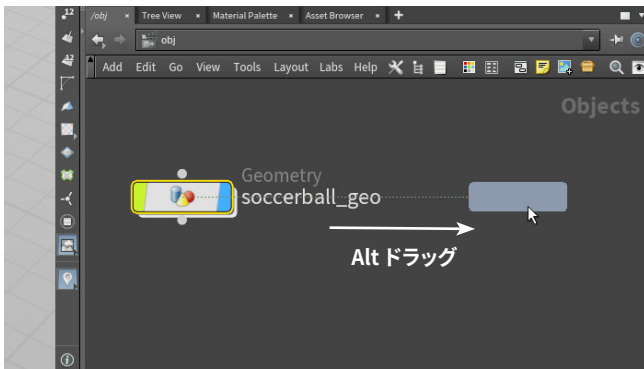
08 ステージレベルに戻ります。**ネットワークビュー** で、**Tab > Karma** を押し、**Karma Render Settings** と **USD Render ROP** ノードを追加します。それらをチェーンの終端に接続します。**karmarendersettings** ノードを選択して、**Image Output > Filters** タブで **Denoyer** を **nvidia Optix Denoyer** に設定してデノイザをオンに戻します。

usdrender_rop ノードを選択します。**Render to Mplay** ボタンをクリックします。**Mplay** が開き、進行中のレンダリングが表示されます。**File > Save Frame As** を選択し、画像をディスクに保存します。

パート7

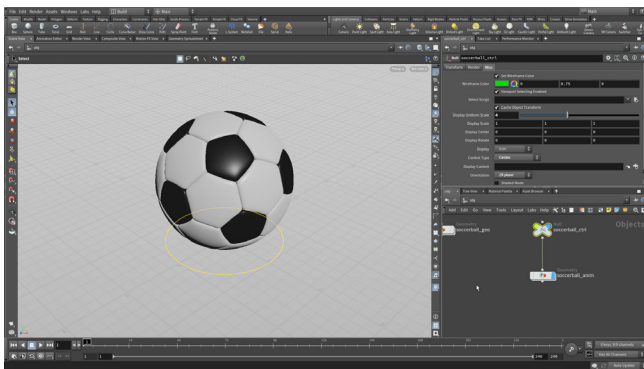
サッカーボールのリギング

バウンシングボールのアニメーションを作成するには、シンプルなリグの作成からはじめます。これで、キーフレームが設定しやすくなります。そのためには、ビューポートでインタラクティブに作業できるように、Null オブジェクトをセットアップします。また、サッカーボールのジオメトリネットワークにノードを追加して、ボールの回転、スクワッシュとストレッチを適用します。



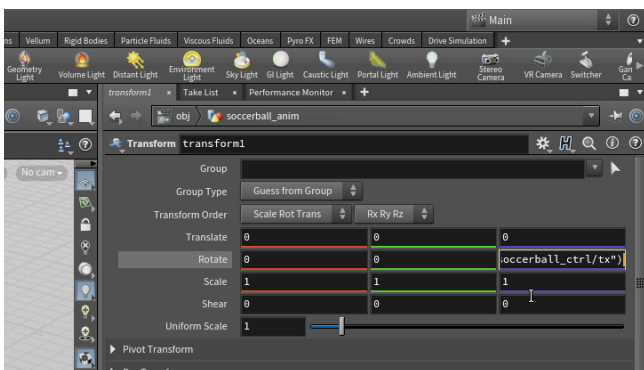
01 Build デスクトップに戻り、パスバーの1つをクリックしてオブジェクトレベルに移動し、**obj** を選択します。ネットワークエディタで、**soccerball_geo** を **Alt ドラッグ** してコピーを作成します。このノードの名前を **soccerball_anim** に変更します。

soccer_anim を使用して、リグを構築します。**soccer_geo** ノードの **Display** フラグをオフにして、非表示にします。元のセットアップは変更したくありません。そのオブジェクトは、Solaris コンテキストのショット1で使用されているからです。この新しいサッカーボールは、アニメートされるショット2で使用します。



02 Create シェルフで、**Null** ツールをクリックしてから **Enter** を押し、それを原点に配置します。名前を **soccerball_ctrl** に変更します。**Misc** タブに移動し、**Control Type** を **Circles** に、**Orientation** を **ZX Plane** に設定します。**Display Uniform Scale** を **4** に設定します。これでリグのハンドルが作成されます。簡単に選択でき、後のレンダリング工程ではレンダリングされません。

ネットワークエディタで、**soccerball_anim** オブジェクトの入力を **soccerball_ctrl** Null の出力に接続し、親子関係を作成します。Null を動かすと、ボールも動くようになります。**soccerball_anim** の **選択フラグ** をオフにして、アニメーションの間ビューポートで誤って選択しないようにします。代わりに **soccerball_ctrl** を使用します。

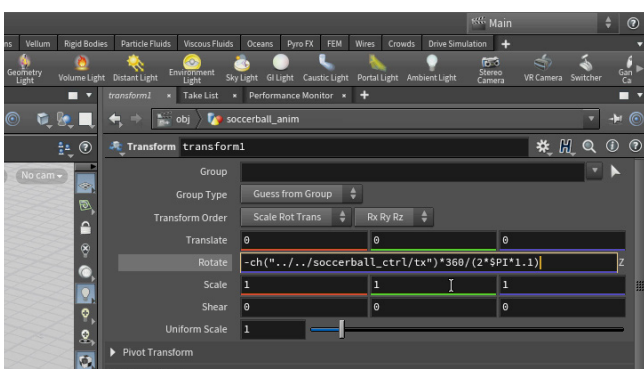


03 soccerball_ctrl ノードを選択します。パラメータエディタで、**Transform** タブをクリックしてから **Translate X** パラメータを **RMB** クリックします。**Copy Parameter** を選択します。

soccerball_anim オブジェクトの中に入ります。**subdivide** と **matchsize** ノードの間に **Transform** ノードを追加します。**Rotate Z** を **RMB** クリックし、**Paste Relative References** を選択します。このパラメータに、チャンネル参照のエクプレッションが配置されます。

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

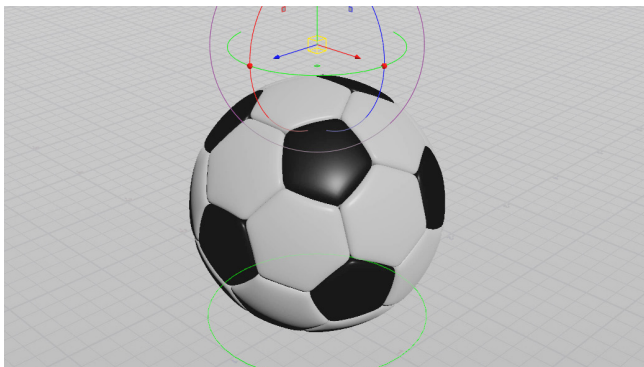
これで、コントロールオブジェクトの動きがこのノードの回転とつながるようになります。



04 パラメータをクリックしてチャンネルを展開します。**ボールの外周 (2πr)** を使用して、ボールが前進するときの回転を定めます。**エクプレッション** を次のように変更します。

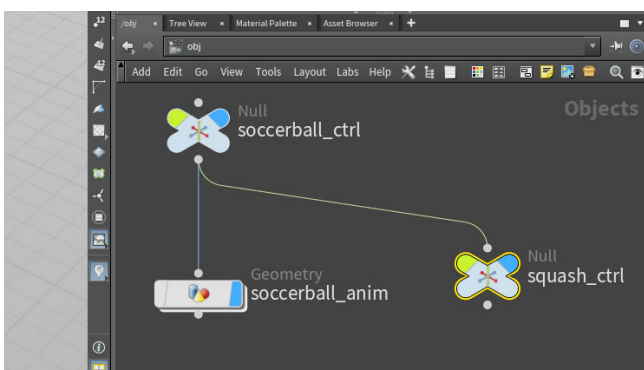
```
-ch("../././soccerball_ctrl/tx")*360/(2*$PI*1.1)
```

まず、先頭にマイナス (-) 記号を追加します。その後、エクプレッションでボールの位置に **360** 度を乗算し、**2πr** (πは \$PI) で除算します。オブジェクトレベルで、**soccerball_ctrl** を **X 軸** に沿って動かします。エクプレッションにより、ボールが動きに合わせて回転します。完了したら、ボールを原点に戻します。



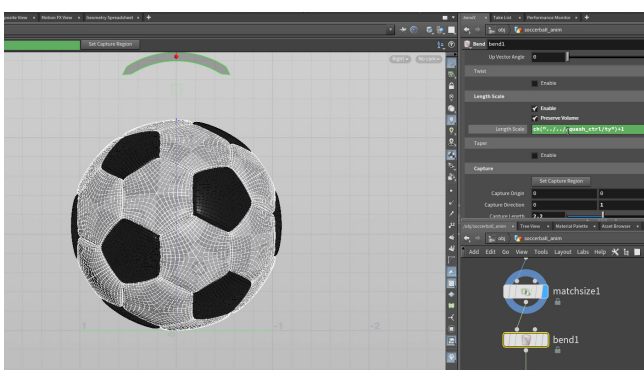
05 ビューポートで、もう1つ Null オブジェクトを原点に作成します。名前を **squash_ctrl** に変更します。Misc タブに移動して、**Control Type** を **Box** に、**Display Uniform Scale** を **0.2** に設定します。

Null をボールの真上に移動します。**Translate Y** は約 **2.5** になるはずですが、パラメータエディタで、**Modify Pre-Transforms** メニューを選び、**Clean Translates** を選択します。これにより、Null が地面より上にあっても、その **Translate Y** 値が **0** に設定されます。Null でスクワッシュとストレッチを駆動するためには、0 のデフォルト値が必要となります。



06 **soccerball_ctrl** Null を **squash_ctrl** Null の親にします。これにより、コントロール Null をアニメートすると、このセカンダリの Null が動くようになります。

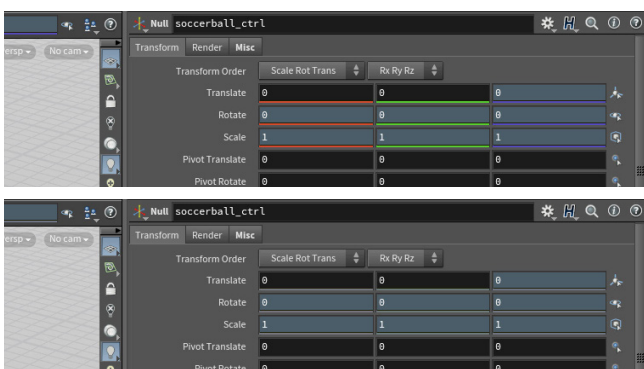
squash_ctrl ノードの **Translate Y** パラメータを **RMB クリック** します。**Copy Parameter** を選択します。このパラメータを使用して、ボールのスクワッシュとストレッチを駆動します。こうすることで、ビューポートからインタラクティブにスクワッシュとストレッチを制御できるようになります。



07 **soccerball_anim** オブジェクトの中に入ります。**matchsize** の後に **Bend** ノードを追加します。**Limit Deformation to Capture Region** チェックボックスを **オフ** にします。

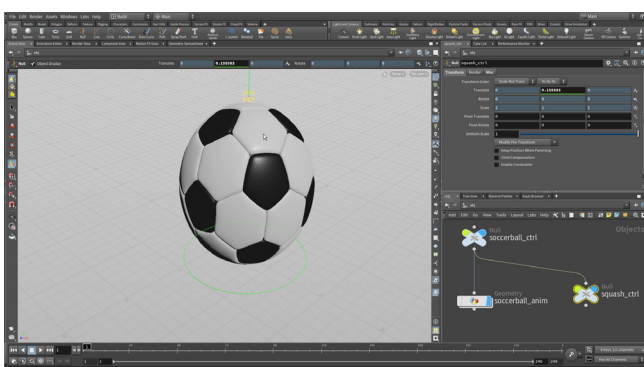
Right ビューに移動して **Set Capture Region** ボタンをクリックします。**グリッドスナッピング** をオンにして、ボールの底部と上部に1つずつポイントを配置します。これにより、**Up Vector** は **0, 0, 1**、**Capture Direction** は **0, 1, 0**、**Capture Length** は **2.2** に設定されるはずですが。

Length Scale と **Preserve Volume** をオンにしたなら、**Length Scale** を **RMB クリック** して、**Paste Relative References** を選択します。エクスプレッションの最後に **+1** を追加します。



08 オブジェクトレベルに移動し、パースビューを表示します。**Transform X** および **Transform Z** パラメータを **RMB クリック** し、**Lock Parameter** を選択して、**squash_ctrl** でこれらのパラメータをロックします **Scale** および **Rotate** パラメータを **RMB クリック** し、**Lock Parameter** を選択して3つすべてのチャンネルをロックします。

soccerball_ctrl オブジェクトを選択します。**Translate X** と **Translate Y** を除くすべてのチャンネルを **ロック** します。これでコントロールを選択すると、ロックされていないチャンネルのハンドルのみが表示されるようになります。アニメータは選択したパラメータのみを操作でき、リグが扱いやすくなります。



09 リグを X と Y で動かして、テストしてみましょう。2つ目のハンドルを使用して、スクワッシュとストレッチを適用します。すべてのパーツがうまく機能していることを確認したら、すべての値を 0 に戻し、アニメーションの準備をします。

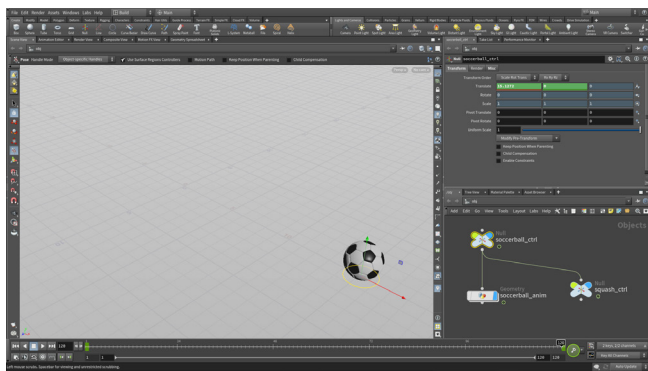
Scene View の左側のツールバーで **Secure Selection** をオフにすることを勧めます。**Move** ツールを使用している間、2つのコントロール Null を選択しやすくなります。オフにしない場合は、選択を切り替えるたびに **S** キーを押す必要があります。

シーンファイルを **保存** してから、次に進みます。

パート 8

バウンシングボールのアニメーション

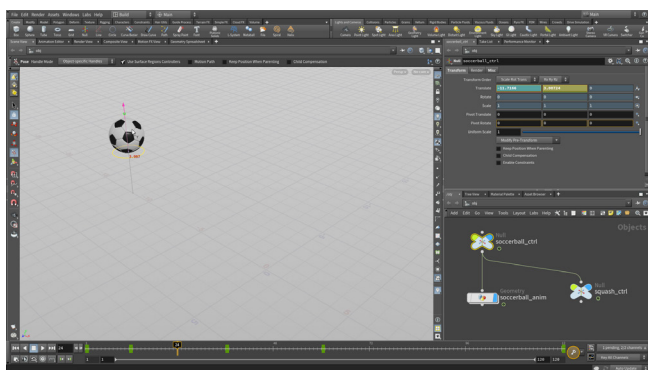
サッカーボールのリグを使用して、バウンドする(弾む)ボールをアニメートします。キーフレームを設定したり、アニメーションカーブを調整したり、ビューポートでハンドルを使って速度を調整する方法を学びます。バウンシングボールは、アニメーションの練習によく使われる課題で、Houdini でのアニメーションの基礎を学ぶのにも最適です。



01 タイムラインの左端の下段で、**Global Animation Options** ボタンをクリックします。**End** を **120** に設定し、**Close** をクリックします。これにより、タイムラインの範囲が 120 フレームに設定されます。

フレーム 1 にいることを確認します。左側のツールバーで **Pose** ツールをクリックし、**soccerball_ctrl** を選択します。ボールを X 方向の約 **-15** に移動したら、**K** を押してキーフレームを設定します。タイムラインを **フレーム 120** に移動します。ボールを X 方向の約 **15** に移動したら、**K** を押して 2 つ目のキーフレームを設定します。

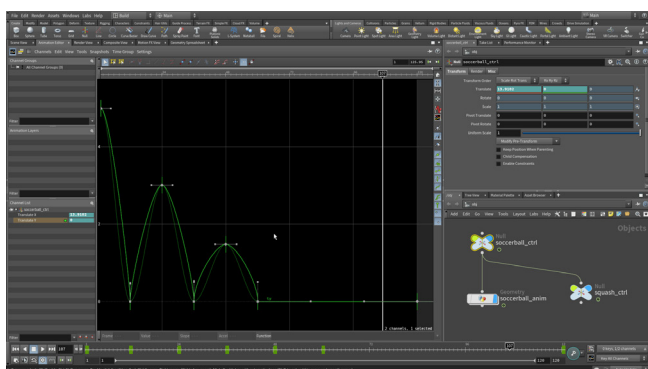
タイムラインを **スクラブ** し、ボールがアニメートされていることを確認します。リグの設計に基づいて、移動および回転するはずですが、



02 タイムラインを **フレーム 12** に移動し、**K** を押して中間のキーを設定します。フレーム **36** と **60** で繰り返します。これらのキーフレームは、すべて地面上にあります。

次に **フレーム 1** に移動し、ボールを Y 方向に上げます。もう 1 つキーを設定する必要はありません。この移動によって、フレーム 1 に既に設定してあるキーフレームが更新されるからです。

フレーム 24 に移動し、ボールをフレーム 1 よりも少し低い位置まで Y 方向に上げます。**K** を押してキーフレームを設定します。**フレーム 48** に移動し、ボールをさらに少し低い位置まで上げます。**K** を押してもう 1 つキーフレームを設定します。



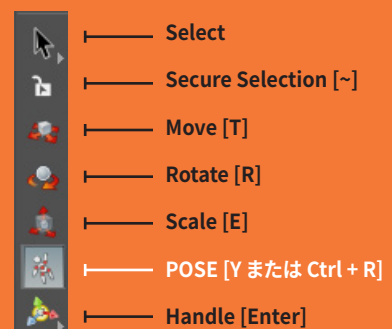
03 タイムラインを **スクラブ** すると、ボールが浮いたように見えますが、ボールが着地するときは、強く当たるようにしたいわけです。**Animation Editor** ペインタブをクリックします。

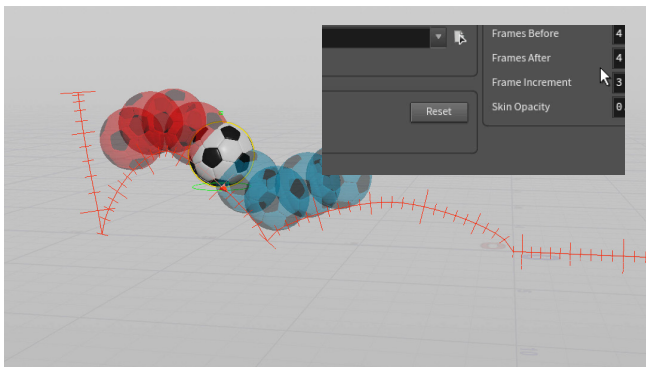
Scoped Parameters リストで **Translate Y** チャンネルをクリックします。**H** を押して、カーブのビューをホーム (リセット) します。ボールが着地する 3 つのキーフレームを選択し、グラフのすぐ上にある **Function** ツールバーで **Untie Handles** ボタンを押します。何も無い空間をクリックして選択を解除したら、接線ハンドルを微調整して、着地の各ポイントのカーブを鋭角にします。また、上部のハンドルは平坦にして、バウンドのピークでボールの速度を遅くします。



Pose ツール

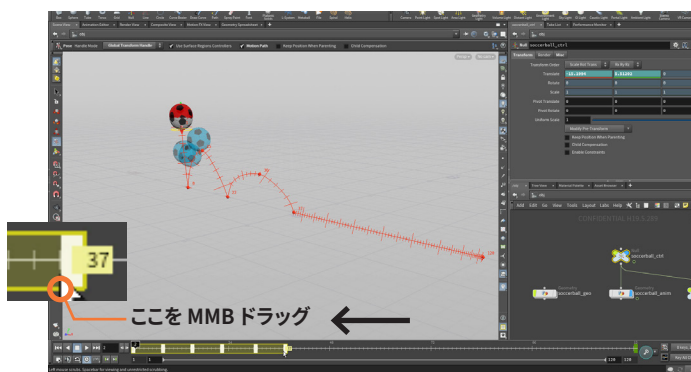
Move または **Handle** ツールを使用すると、サッカーボールのリグの 2 つのコントロールオブジェクトを簡単に操作できます。**Pose** ツールを使用する利点は、**Motion Path** ハンドルにアクセスでき、インバースキネマティクスを使用している場合は、専用のハンドルを使用できることです。そのため、リグにキーフレームを設定する場合は、このツールを必ず思い出してください。**Secure Selection** がオンになっていても、**Pose** ツールで別のオブジェクトを選択可能です。





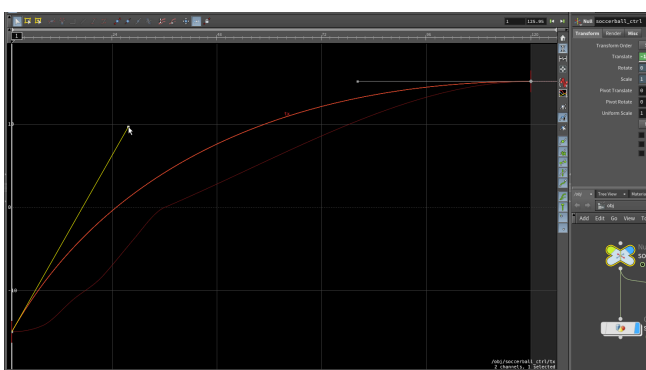
04 Scene View に戻り、結果をプレビューします。ボールが着地するたびに、弾むようになりました。**Pose** ツールがアクティブで、**soccerball_ctrl** ノードが選択されていることを確認します。上部のバーで、**Motion Path** ハンドルを**オン**にします。これにより、ボールのバウンドのパスが表示されます。それぞれのキーフレームマーカーをクリックして、弾み方を微調整します。カーブをより詳細に制御するには、ハンドルを**RMB**クリックして**Show Tangents** をオンにします。

soccerball_anim オブジェクトの **Misc** タブに移動して、**Onion Skinning** を **Full Deformation** に設定します。**スペースバー + D** を押し、**Scene** タブで **Frame Increment**、**Frames Before**、**Frames After** のカラーを調整します。



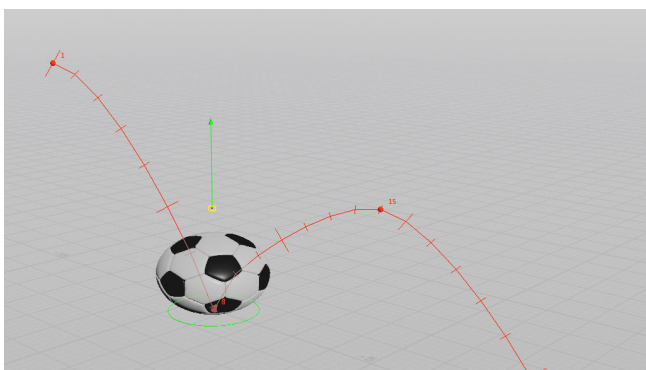
05 タイミングの調整には、タイムラインも使用できます。**Shift** を押し、フレーム 1 からタイムラインの最後のキーまで境界ボックスをドラッグして、すべてのキーを**選択**します。次に、ボックスの端のハンドルを下を **MMB** ドラッグして、バウンドのタイミングをスケールし、スピードを上げます。**MMB** を使用して各キーを選択してから、**MMB** でドラッグしても、それぞれのキーフレームのタイミングを好きなように調整できます。

ここでは、バウンドのタイミングを決定します。希望する動き方になるまで試してください。Translate X の値が原因で、弾み方が自然ではないかもしれません。それは次の手順で修正します。



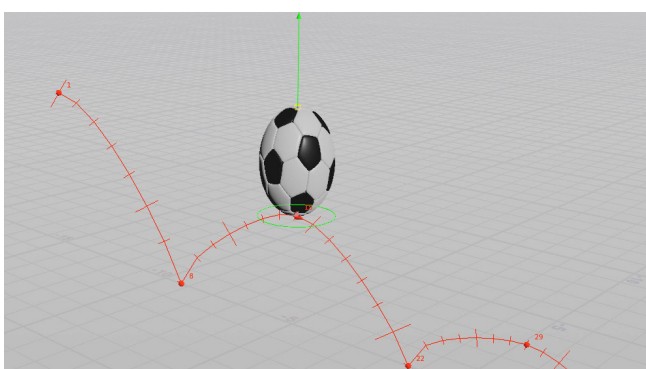
06 Animation Editor ペインタブをクリックして、2つのカーブを表示します。Scoped Parameters リストで、**soccerball_ctrl** の **Translate X** をクリックします。最初と最後のキーを除く、すべてのキーを選択します。**Delete** を押します。カーブハンドルを使用して、急な傾斜が緩い傾斜に移行するようにします。こうすると、ボールの動きは始めが速く、終わりでゆっくりになります。

Motion Path ハンドルで X 方向のポイントを再び調整すると、その方向には中間キーがないため、おかしい結果になることに注意してください。これ以降は、Y 方向の微調整にのみこのハンドルを使用します。

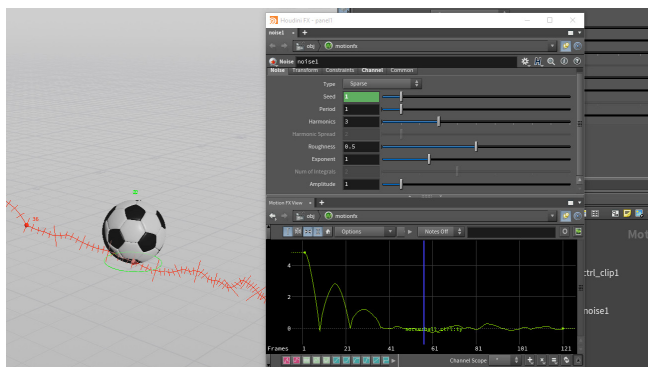


07 Scene View に戻ります。**Motion Path** ハンドルを **RMB** クリックし、**Persistent** を選択します。これにより、スクワッシュとストレッチにキーフレームを設定する際のガイドとして、ハンドルを保持しておけます。

squash_ctrl を選択し、**Motion Path** をオフにします。最初のバウンドに移動して、1 フレーム戻ります。**squash_ctrl** のハンドルを選択し、ボールを少し引き伸ばし(ストレッチ)します。**K** キーをで**キーフレームを設定**します。バウンドのフレームに移動し、ハンドルを下げてスクワッシュ(潰し)を作成します。**もう1つキーを設定**します。1 フレーム進め、ボールが丸くなるまでストレッチし(引き伸ばし)します。**もう1つキーを設定**します。この手順をすべてのバウンドで繰り返します。

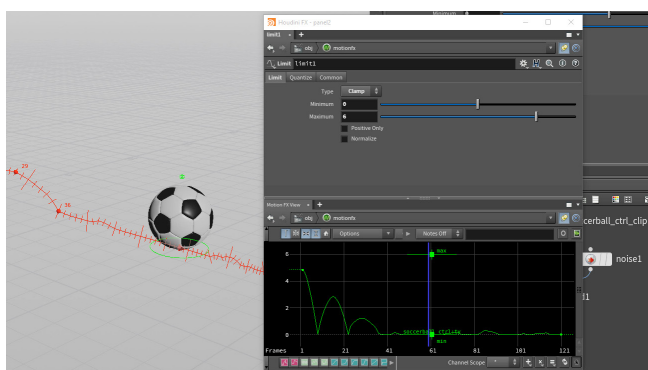


08 完了したら、モーションをスクラブして再生し、結果をプレビューします。バウンドのピークで、ボールが引き伸ばされていることを確認してください(ストレッチ)。モーションを正しく評価するには、タイムラインで **Real Time Toggle** が**オン**になっていることを確認します。次に **Animation Editor** を使用して、スクワッシュとストレッチを微調整します。キーフレームとサッカーボールのバウンドが揃っていることを確認しましょう。



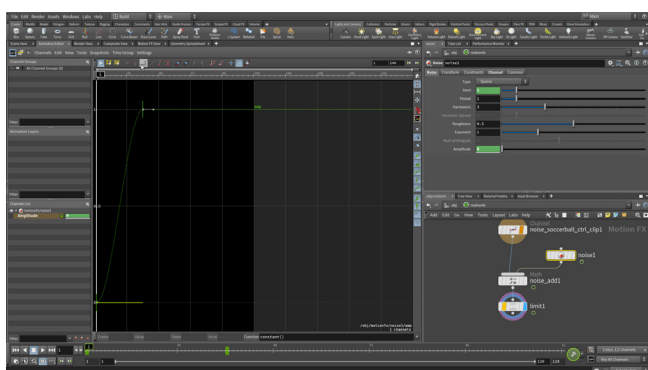
09 **soccerball_ctrl** Null オブジェクトを選択します。Translate Y を RMB クリックし、Motion FX > Noise を選択します。パネルが開き、そこに表示されたパラメータを使用してノイズを制御できます。**soccerball_ctrl** の Translate Y チャンネルに情報を送り返す CHOP ノードを含む、新しいサブネットワークが作成されます。

Amplitude を 5 に設定し、Play を押してどのように見えるかを確認します。上下に大きく動き、まるで激しい乱流のようです。**Amplitude** を 1 に設定して、ボールの動きに加わる揺れを少なくします。



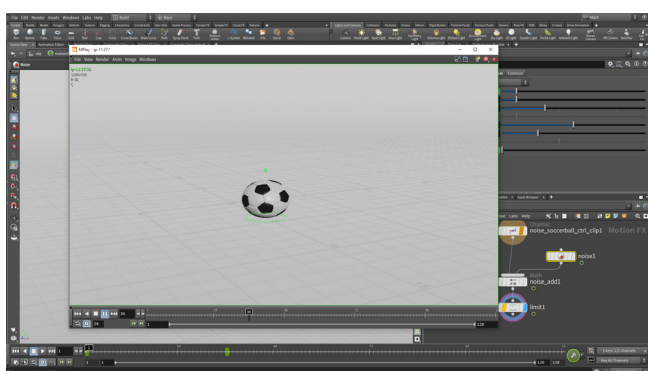
10 動きの一部が地面の下にもぐっています。ボールが地面よりも上で揺れるようにする必要があります。

オブジェクトレベルに戻ります。**soccerball_ctrl** Null オブジェクトを選択します。Translate Y を RMB クリックし、Motion FX > Limit を選択します。**Minimum** を 0 に、**Maximum** を 6 に設定します。これで、ボールはずっと上下に揺れ動くのではなく、いらか揺れながら地上を移動するようになります。



11 ボールが弾んでいる間は、ノイズは必要ありません。ノイズが必要なのは、ボールが転がり始めてからです。**Amplitude** のキーフレームを設定して、ノイズをオン/オフできるようにします。

新しく作成した **motionfx** ネットワークで、**noise1** CHOP ノードを選択します。ボールが弾まなくなり、転がり始めるフレーム 37 に移動します。**Amplitude** を Alt クリックして、キーフレームを設定します。フレーム 1 に移動して、**Amplitude** を 0 に設定します。**Amplitude** を再度 Alt クリックして、2 つ目のキーフレームを設定します。**Animation Editor** でカーブを選択します。Function ツールバーの **Constant** ボタンをクリックします。これで、振幅なしから振幅 1 へとシャープに切り替わります。



12 完了したら、ウィンドウを閉じます。Motion Path ハンドルを RMB クリックして Persistent をオフにし、選択解除したらハンドルが表示されないようにします。

Scene View の左側にあるツールバーで、Render Flipbook ボタンをクリックします。デフォルト設定のまま、Start をクリックします。シーケンスがキャプチャされるまで待つと、Mplay ウィンドウにフリップブックが表示されます。これを再生したりスクラブして、動きを評価します。

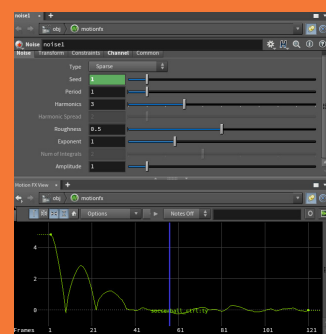
作業内容を保存します。



Motion FX

キーフレームとアニメーションカーブはノードのパラメータに格納されますが、チャンネルオペレータ (CHOP) を使用すると、よりプロシージャルなノードベースのアプローチでモーションを操作することができます。

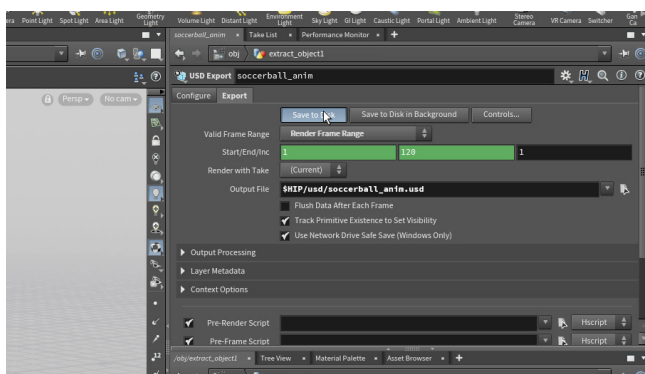
Motion FX は、Channel CHOP に抽出および格納される、キーフレームによる動きに適用できます。その後、Cycle、Noise、Smooth、Limit、Lag などのエフェクトを既存の動きに適用できます。Constraints シェルフにはさまざまなツールがあり、パラメータ設定によってターゲットの方を向くようにしたり、遅延させたり、微震するようにできます。



パート9

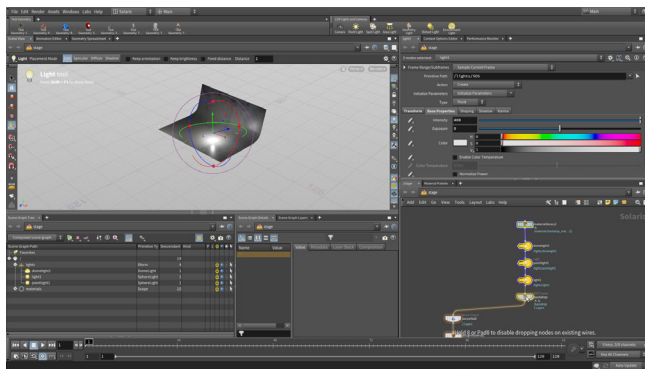
ライト、カメラ、アクション!

アニメートしたサッカーボールをレンダリングするには、Solaris 環境に戻って 2 つ目のショットをセットアップする必要があります。まず、背景ジオメトリから新しい LOP ノードを分岐させてから、バウンドするサッカーボールのアニメーションに合わせてライトとカメラを調整します。また、変形するジオメトリに対してモーションブラーもセットアップします。



01 オブジェクトレベルに移動します。**soccerball_anim** を**ダブルクリック**して中に入ります。**N** を押し、すべてのジオメトリを選択したら、**Modify** シェルフで **Extract** を選択します。これにより、ボールのすべての動きと曲げが 1 つのネットワークにまとめられます。

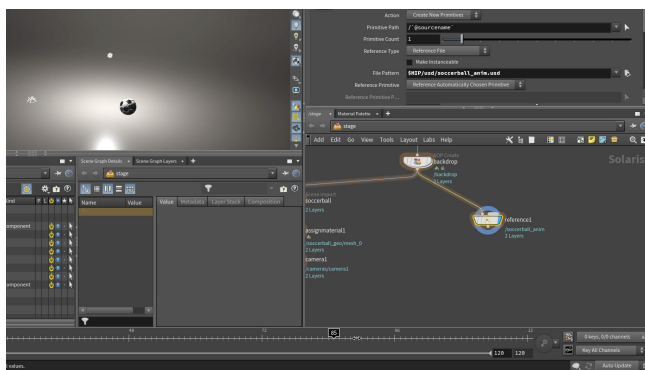
ボールを **extract_object** という新しいオブジェクトに抽出している **objectmerge** ノードが表示されます。その出力を **RMB クリック**し、**USD Export** を見つけたら、クリックしてこのノードを配置します。このノードを **soccerball_anim** という名前に変更します。**Export** タブをクリックし、**Valid Frame Range** を **Render Frame Range** に、**Output File** を **\$HIP/geo/soccerball_anim.usd** に設定します。**Save to Disk** ボタンをクリックします。



02 Solaris デスクトップに戻り、**/stage** を指定します。**ネットワークビュー**で、**karmarendersettings** ノードの直前に **Null** ノードを追加し、その名前を **SHOT_01** に変更します。

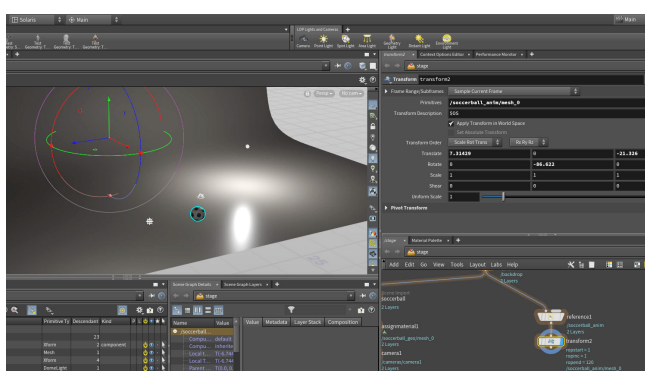
3 つの **light** ノード (**lightmixer** ではありません) を接続解除し、それらを **backdrop** ノードの上に移動します。これにより、最初のショットのルックは変わらず、2 つのショット間でノードを共有できるようになります。

backdrop、**light**、**materiallibrary** ノードを右に移動します。



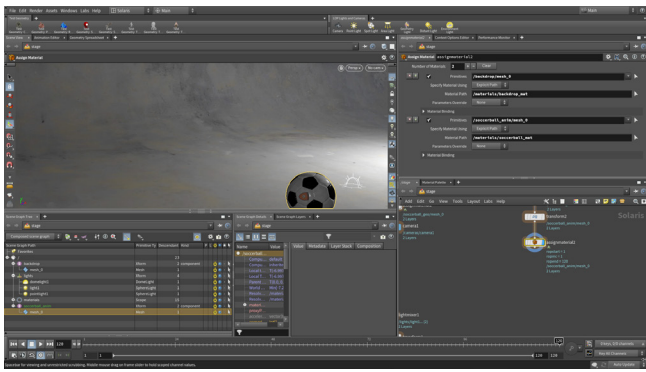
03 ズームインして、**Reference** ノードを **backdrop** ノードの右下に追加します。**backdrop** ノードをこの新しいノードに接続し、**Display フラグ**を設定します。**File Pattern** の横にある **File Chooser** をクリックし、**soccerball_anim.usd** ファイルを見つけます。ノードの名前を **soccerball_anim** に変更します。

Houdini GL ビューでタイムラインをスクラブすると、USD ファイルの一部であるキャッシュ化されたアニメーションを確認できます。



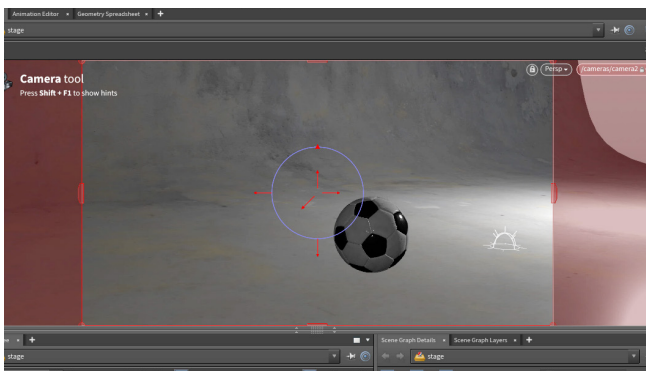
04 **Select** ツールを使用して、新しいアニメートされたサッカーボールをクリックします。Scene View で、**Tab > Transform** を押し、**Transform** ノードをグラフに追加します。

Scene View で、トランスフォームハンドルを使用してボールを中央の背景の後方に**移動**します。タイムラインを**スクラブ**して、ボールが弾みながら右に進むことを確認します。フレーム 80 あたりで止めます。**R** を押し、回転ハンドルを使用できるようにします。ボールを**回転**し、背景から斜めにバウンドしてくるように移動します。タイムラインを**スクラブ**して、方向に問題がないことを確認します。



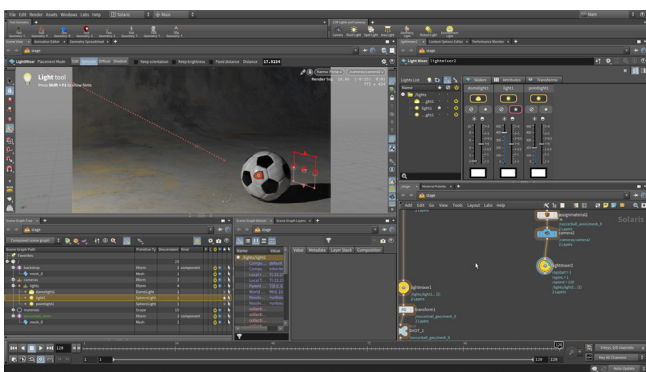
05 ネットワークビューで、SHOT 1 ネットワークから **assignmaterial** ノードを選択し、**Alt** ドラッグしてこのノードのコピーを作成します。**transform** ノードを **assignmaterial** ノードに接続したら、**Display フラグ**を設定します。これによりマテリアルが背景に割り当てられますが、サッカーボールのプリミティブが変更されているため、マテリアルを割り当て直す必要があります。

soccerball_mat の **Primitives** の横にあるフィールドで、プリミティブの名前を **/soccerball_anim** に変更し、新しいジオメトリにマテリアルを割り当て直します。



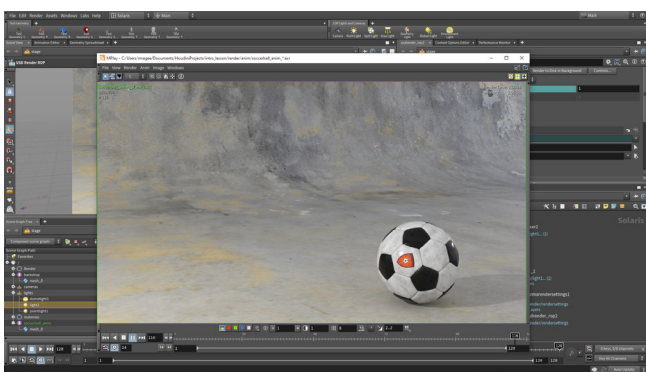
06 左上から右下へと、ボールがカメラに向かってアニメートされる様子が見えるまで、タンブルします。**LOP Lights and Cameras** シェルフで、Camera ツールを **Alt** クリックして、現在見ているアングルにカメラを配置します。

Lock Camera/Light to View ボタンを押し、ビュー変更に応じてカメラの位置が更新されるようにします。次にビューポートで**タンブル**、**パン**、**ドリー**してカメラを微調整し、ショットに適したフレーミングにします。タイムラインをスクラブし、シーケンス全体でカメラが機能していることを確認します。



07 camera の後に **Light Mixer** ノードを追加します。**lightmixer** ノードでは、ライト上にカーソルを移動します。こうすると、前に紹介したライトハンドルを使用して、このショットのライティングを決定したり、ビューポートの **Karma** 表示を使ってセットアップを確認することができます。**lightmixer** ノードを使用すると、このショットの**強度**や**露出**をさまざまに変えて試すことも可能です。

これらの編集は **lightmixer** ノードに保持され、元のライトは変更されません。**lightmixer** では、マルチショットセットアップを使用しながら、既存のライトを微調整できます。



08 ネットワークビューで、**SHOT_01**、**karmarendersettings**、**usdrender_rop** ノードを **Alt** ドラッグします。**lightmixer** ノードをこのチェーンに接続します。新しい **karmarendersettings** ノードを選択し、**Camera** が **/camera2** に設定されていることを確認します。**Valid Frame Range** を **Render Frame Range** に、**Output Picture** を **\$HIP/render/anim/soccerball_anim_\$.F2.exr** に設定します。**\$.F2** によって、レンダリングにフレーム番号と 2 のパディングが追加され、**/anim/** によって、これらのフレームを保持するためのディレクトリが作成されます。

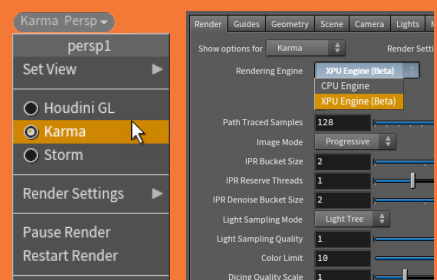
usdrender_rop ノードで、**Render to Disk** をクリックします。完了したら、**Render > Mplay > Load Disk Files** を選択し、レンダリングした画像を開いて最終的なシーケンスを確認します。作業内容を**保存**します。



KARMA レンダラ

Karma は、Solaris/LOP コンテキストで **USD** ファイルを使用できるよう設計された物理ベースの **HYDRA** レンダラです。ビューポートで使用してインタラクティブに更新したり、**Karma** ノードを使用してディスクにレンダリングできます。

注: Houdini 19 は、**Karma XPU** レンダリングエンジンのプレビューに対応しています。このハイブリッドな GPU/CPU レンダラは**アルファ**品質で、多くの機能はまだ開発中であるため、**テスト目的でのみ**使用します。**XPU** は、Scene View の **Display Options** または **Karma** ノードで選択できます。



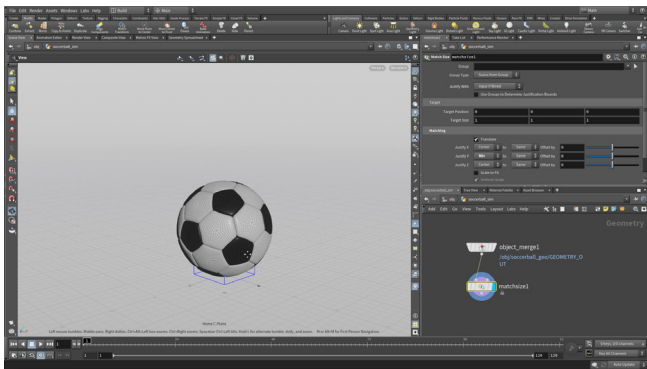
ビューポートメニュー

Display Options

パート 10

リジッドボディシミュレーションのセットアップ

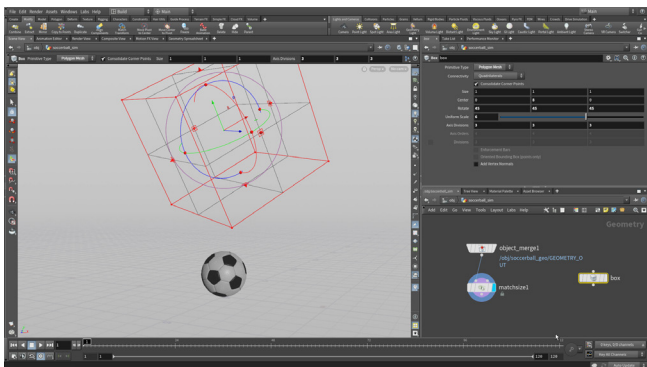
1個のサッカーボールをアニメートするには、従来のアニメーション方法が適していますが、たくさんのサッカーボールをアニメートしたい場合は、ダイナミクスの使用がおすすめです。ダイナミクスを使うには、ソルバがフレームごとに各オブジェクトの相互作用を判断できるように、シミュレーションを行う必要があります。パックドジオメトリを使用すると、効率よくシミュレーションの結果を得られます。



01 Build デスクトップに戻り、オブジェクトレベルに移動します。すべてのアニメーションリグノードと **extract_object** ノードを、Display フラグをオフにして非表示にします。**soccerball_geo** の表示をオンにします。

soccerball_geo ノードを選択してから、**Modify** シェルフの **Extract** ツールをクリックします。これにより、サッカーボールのオブジェクトが結合された、新しいオブジェクトが作成されます。1つ上のレベルに移動し、**extract_object** の名前を **soccerball_sim** に変更します。**soccerball_geo** オブジェクトを非表示にします。

soccerball_sim オブジェクトの中に戻り、ジオメトリを作業していきます。**Match Size** ノードを追加して、原点を基準とした中央にボールを配置します。

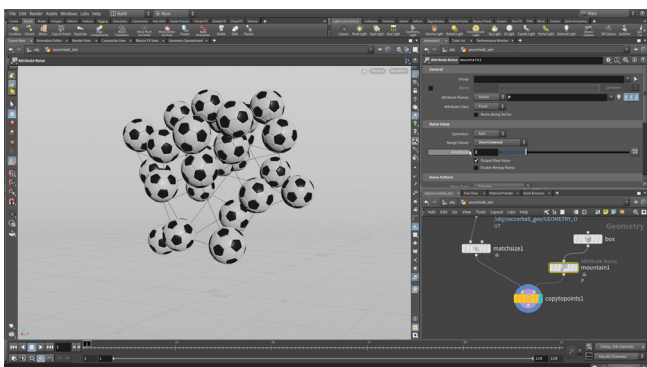


02 ネットワークビューで、**Tab > Box** を押し、それを **matchsize** ノードの右側に配置します。

box ノードで次のように設定します。

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

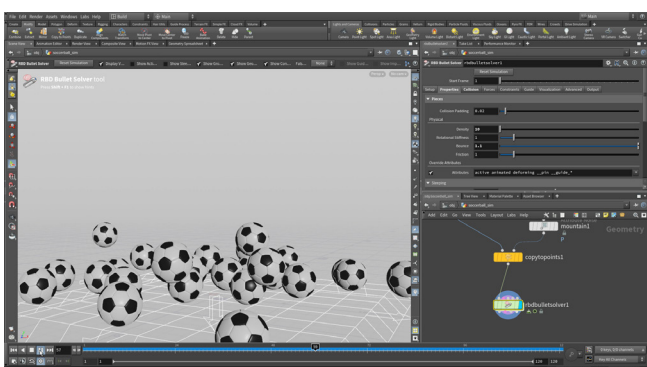
これで、シミュレーションでの位置が適切になります。



03 ネットワークビューで、その他のノードのすぐ下に **Copy to Points** ノードを追加します。**matchsize** ノードを1つ目の入力に接続し、**box** ノードを2つ目の入力に接続します。

Pack and Instance オプションを**オン**にします。これにより、ジオメトリが立方体のポイントにインスタンス化されるので、シミュレーションが高速化します。**copytopoints** ノードに **Display フラグ**を設定します。

ネットワークビューで、**Tab > Mountain** を押して、**box** と **copytopoint** ノードの間にそのノードを配置します。**Noise Along Vector** オプションを**オフ**にしてから、**Amplitude** を **2** に、**Range Values** を **Zero Centered** に設定します。これで、ボックスのポイントが微震するようになります。



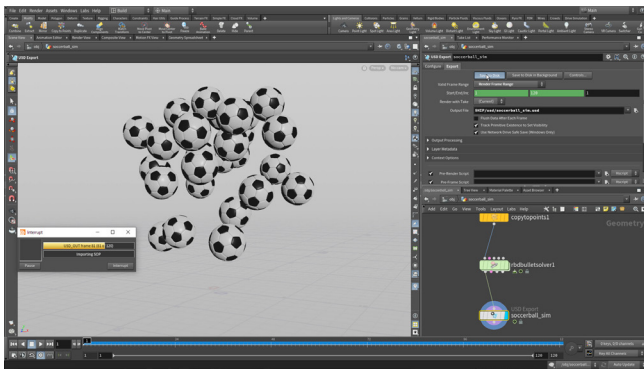
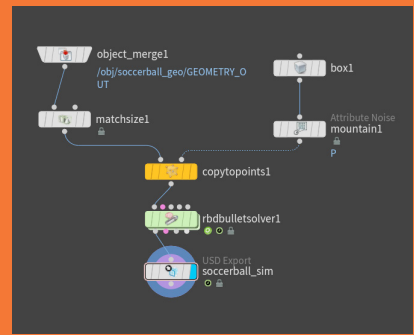
04 フレーム **1** にいることを確認します。**copytopoints** ノードの後に **RBD Bullet Solver** ノードを追加します。**Collision タブ** をクリックし、**Ground Collision** までスクロールダウンして、**Ground Type** を **Ground Plane** に設定します。**Play** を押して、シミュレーションをテストします。シミュレーションはキャッシュ化され、タイムラインをスクラブして結果を確認できるようになります。

Collisions タブで、**Bounce** を **0.8** に設定します。**Properties** タブで、**Density** を **10** に、**Bounce** を **1.1** に設定します。このノードのパラメータエディタの上部で、**Reset Simulation** ボタンをクリックしてから **Play** を押し、再度シミュレーションを実行します。スクラブして確認します。



SOP 内に隠れた DOP

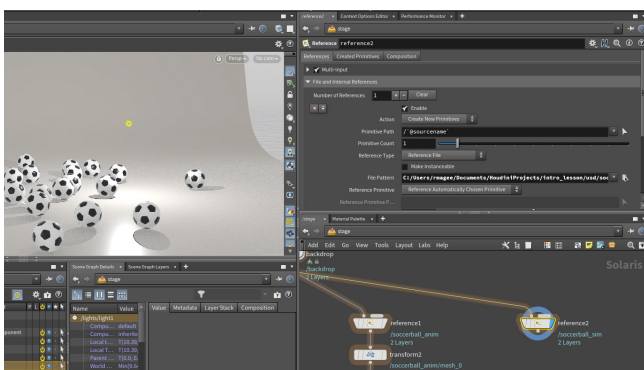
Houdini では、シミュレーションは **Dynamic Operators**、つまり **DOP** を使用して処理されます。**Geometry/SOP** コンテキストの **RBD Bullet Solver** ノードでは、内部にダイナミクスネットワークが埋め込まれたノードを使用することになります。これにより、すべての DOP ノードを接続して準備を整え、かつビューに表示されない状態を簡単にジオメトリレベルでセットアップできるようになりました。シンプルなセットアップでは、ジオメトリレベルで作業することで適切なシミュレーションが得られます。さまざまなソルバをより詳細に制御する必要がある場合は、DOP で直接作業する必要があります。



05 チェーンの終端に **USD Export** ノードを追加し、**Display フラグ**を設定したら、それを **soccerball_sim** という名前に変更します。

Valid Frame Range を **Render Frame Range** に、**Output File** を **\$HIP/geo/soccerball_sim.usd** に設定します。

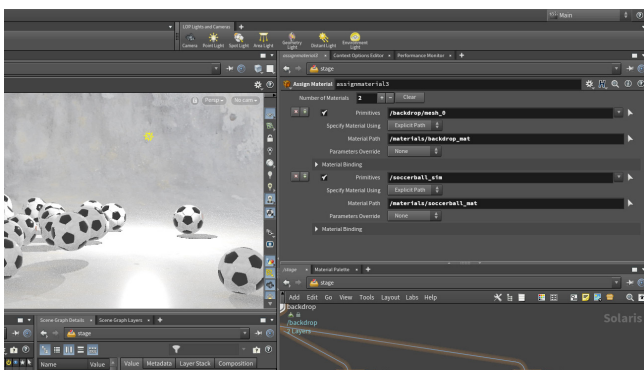
Save to Disk ボタンをクリックして、USD ファイルを geo ディレクトリに保存します。このキャッシュ化されたアセットは、3つ目のショットとして Solaris セットアップで参照されます。



06 デスクトップを **Solaris** に戻し、パスを **/stage** に設定します。**Persp** メニューで **Houdini GL** が選択されていることを確認します。

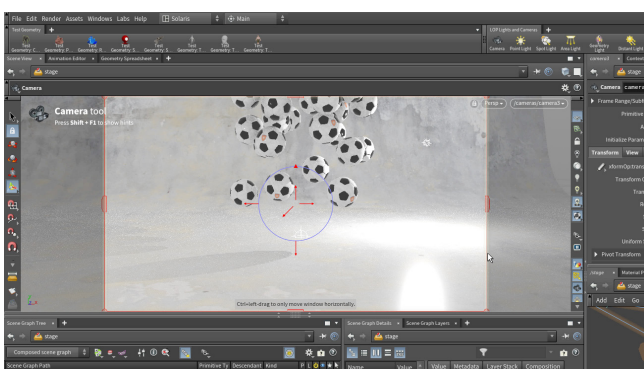
soccerball_anim **Reference** ノードを **Alt** ドラッグし、その **Display フラグ** を設定します。**File Pattern** を **\$HIP/geo/soccerball_sim.usd** に設定します。

このノードを **soccerball_anim** という名前に変更します。



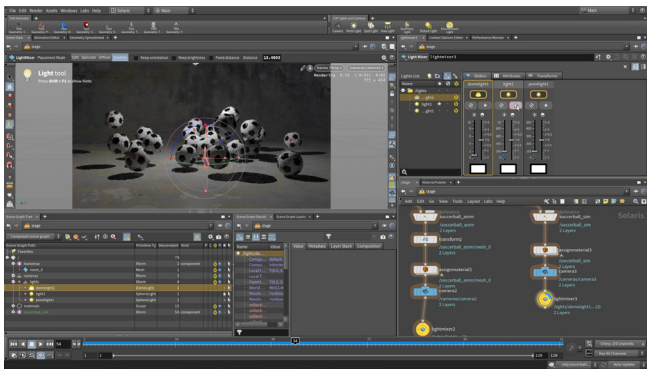
07 ネットワークビューで、SHOT 2 ネットワークから **assignmaterial** ノードを選択し、**Alt** ドラッグしてこのノードのコピーを作成します。**soccerball_anim** ノードを **assignmaterial** ノードに接続したら、その **Display フラグ** を設定します。これにより材料が背景に割り当てられますが、サッカーボールのプリミティブが変更されているため、材料を割り当て直す必要があります。

soccerball_mat の **Primitives** の横にあるフィールドで、プリミティブの名前を **/soccerball_sim** に変更し、新しいジオメトリに材料を割り当て直します。



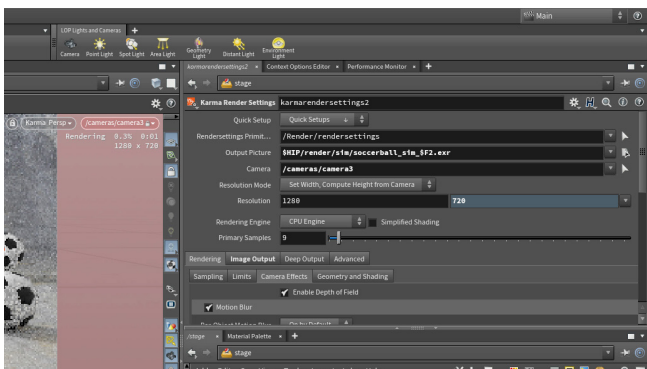
08 ボールがカメラに向かってアニメートされる様子が見えるまで、タンブルします。**LOP Lights and Cameras** シェルフで、**Camera** ツールを **Alt** クリックして、現在見ているアングルにカメラを配置します。

Lock Camera/Light to View ボタンを押し、ビュー変更に応じてカメラの位置が更新されるようにします。次にビューポートで**タンブル**、**パン**、**ドリー**してカメラを微調整し、ショットに適したフレーミングにします。タイムラインをスクラブルし、シーケンス全体でカメラが機能していることを確認します。

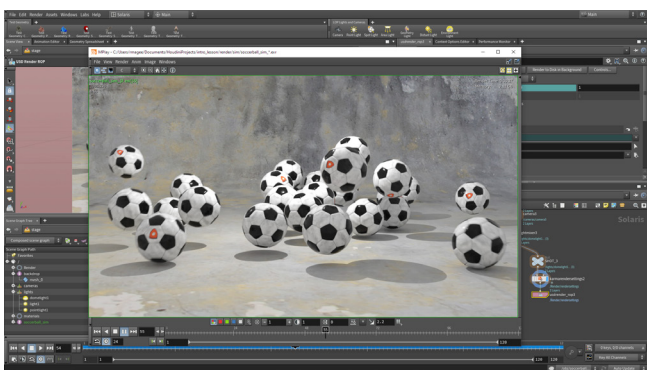


09 camera の後に **Light Mixer** ノードを追加します。**lightmixer** ノードでは、ライト上にカーソルを移動します。こうすると、前に紹介したライトハンドルを使用して、このショットのライティングを決定したり、ビューポートの **Karma** 表示を使ってセットアップを確認することができます。**lightmixer** ノードを使用すると、このショットの**強度**や**露出**をさまざまに変えて試すことも可能です。

これらの編集は **lightmixer** ノードに保持され、元のライトは変更されません。**lightmixer** では、マルチショットセットアップを使用しながら、既存のライトを微調整できます。



10 ネットワークビューで、**SHOT_02**、**karmarendersettings**、**usdrender_rop** ノードを **Alt** ドラッグします。新しい **lightmixer** ノードをこのチェーンに接続します。新しい **karmarendersettings** ノードを選択し、**Camera** が **/camera3** に設定されていることを確認します。



11 usdrender_rop ノードで、**Valid Frame Range** を **Render Frame Range** に、**Output Picture** を **\$HIP/render/sim/soccerball_sim_5F2.exr** に設定します。**Render to Disk** をクリックします。

完了したら、**Render > Mplay > Load Disk Files** を選択し、レンダリングした画像を開いて最終的なシーケンスを確認します。



まとめ

Houdini のさまざまな側面に触れながら、ゼロからシーンを構築してきました。モデリング、テクスチャのセットアップ、レンダリング、シミュレーションを行いました。その過程で、さまざまな Houdini コンテキストと、それらの切り替え方法についても学習しました。

このレッスンは、VFX の大作を作るものではありません。Houdini を深く探り、その包括的なツールセットを探求するにあたって必要となる基本スキルを紹介するものです。

SideFX の Web サイトには、スキルアップのための学習教材が豊富に用意されています。

最高の旅になりますように!



