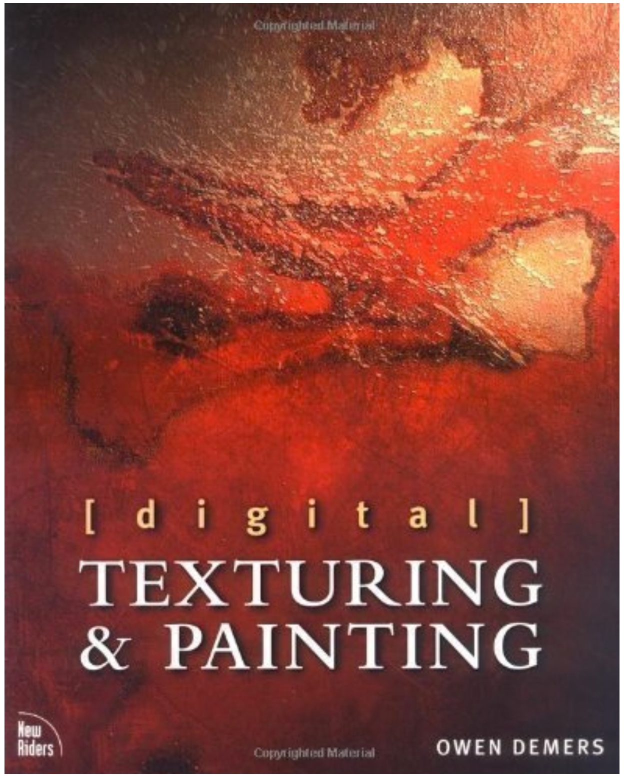
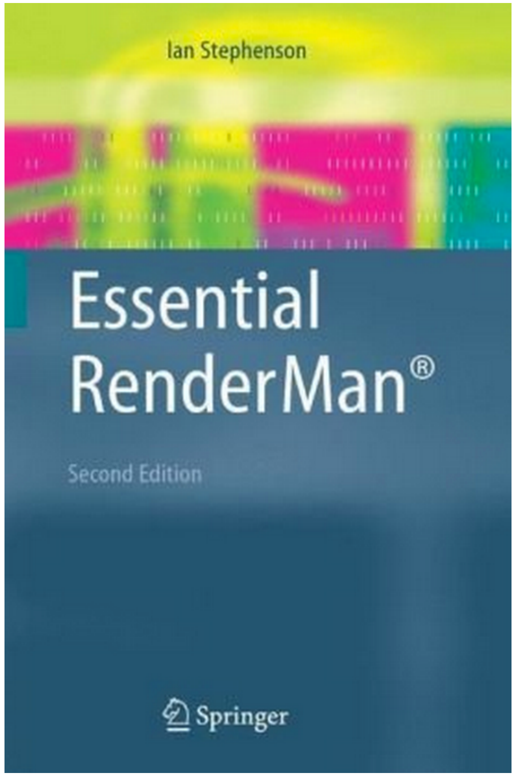
RECOMMENDED READING:



For the principles of texturing.



For developing custom shaders.

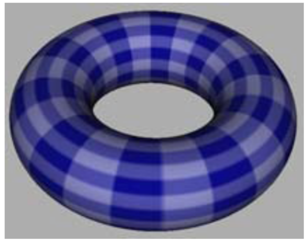
For the principles of matte painting.

Texturing is a process for **colouring** and **shading geometry.** It can be utilised to create a **variety of aesthetics** from **'toon-shading'** to **photo-realism**.

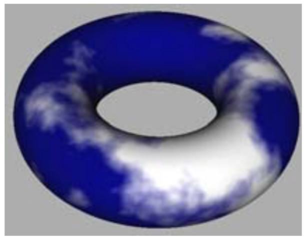
Iron Giant (1999)

Star Wars Ep. VII (2015)

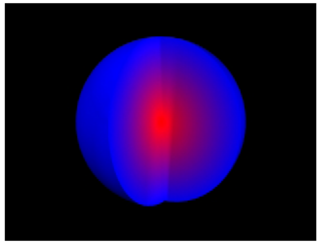
Texturing can be classified into **Procedural Textures**, or **Painted Textures**. A **Procedural Texture** is where **procedural mathematical code** is used to **generate patterns** or **noise**. A **Painted Texture** is **generated** from **an image**, for surface properties that are not easily generated procedurally.



Procedural Gingham

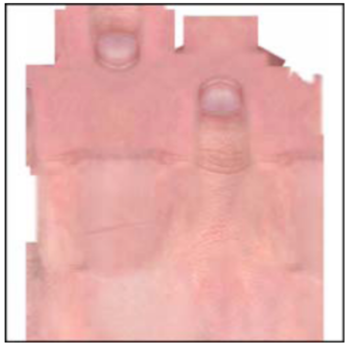


Procedural Noise



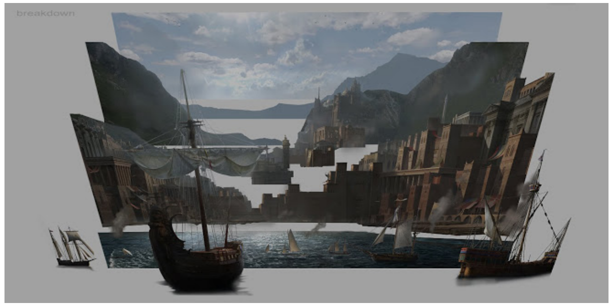
Procedural 3D centre core

Painted Texture Map applied to geometry



Painted Texture Map

Complex landscapes can also be generated using **geometry planes** with **painted textures** assigned to them. This is known as **Matte Painting**.



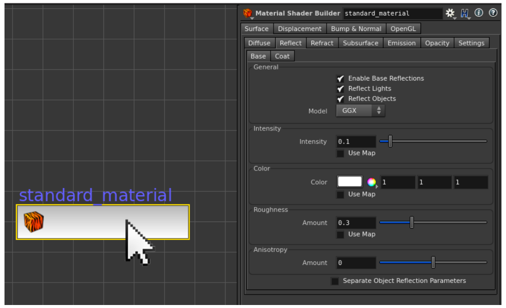
(Balan, 2013)

Underlying each texture is a **Surface Lighting Model**. A Surface Lighting Model **defines all of the properties of the surface** and tells the **Renderer** how the surface should **interact with scene lighting**. The **combination** of a (Painted or Procedural) **Texture** and a **Surface Lighting Model** equates to a **Material.**

USAGE: Mattes & Flat Shading	USAGE: X-Ray / Electron Microscope (when inverted)	USAGE: Carpets, Walls, Ceilings, Clay, Paper	USAGE: Physically Accurate Surfaces	USAGE: High Gloss Plastics	USAGE: Low Gloss Plastics, Skin	USAGE: Metals, Varnished Wood, Ceramics
Scene Lighting Interaction: NO	Scene Lighting Interaction: NO	Scene Lighting Interaction: YES	Scene Lighting Interaction: YES	Scene Lighting Interaction: YES	Scene Lighting Interaction: YES	Scene Lighting Interaction: YES

MATERIAL PROPERTIES

At a **top level**, a standard **Material** has a **number of artist-friendly parameters** that can be **adjusted** to modify a material aesthetic. These include **Surface parameters** for controlling **Diffuse**, **Reflect**, **Refract**, **Subsurface**, **Emission**, and **Opacity**; as well as **Displacement parameters** for creating **bumpy surface aesthetics**. A **standard material** can be **configured** to create **many different aesthetics**, simply by **modifying its Material Properties**. For example, a basic **chrome material** can be created by **reducing** the **Diffuse** component, and **increasing** the **Reflection** component.

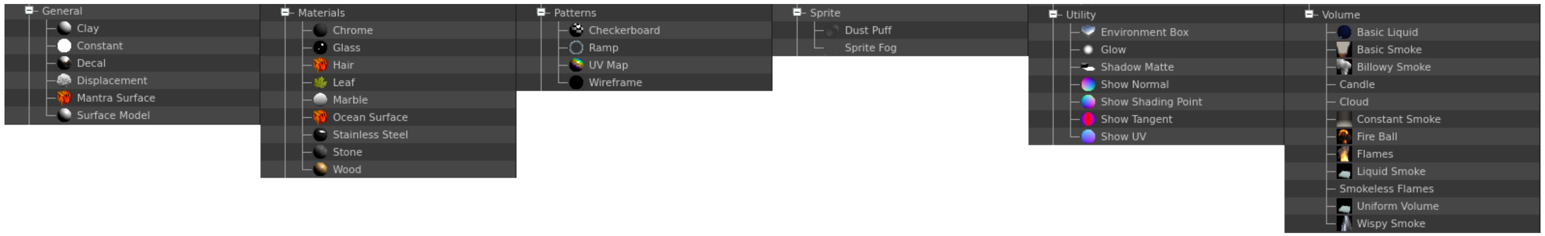


Standard Material
Diffuse Intensity 0.5
+
Reflect Intensity 0.1

Standard Material
Diffuse Intensity 0.1
+
Reflect Intensity 0.1

Standard Material
Diffuse Intensity 0.1
+
Reflect Intensity 1.0

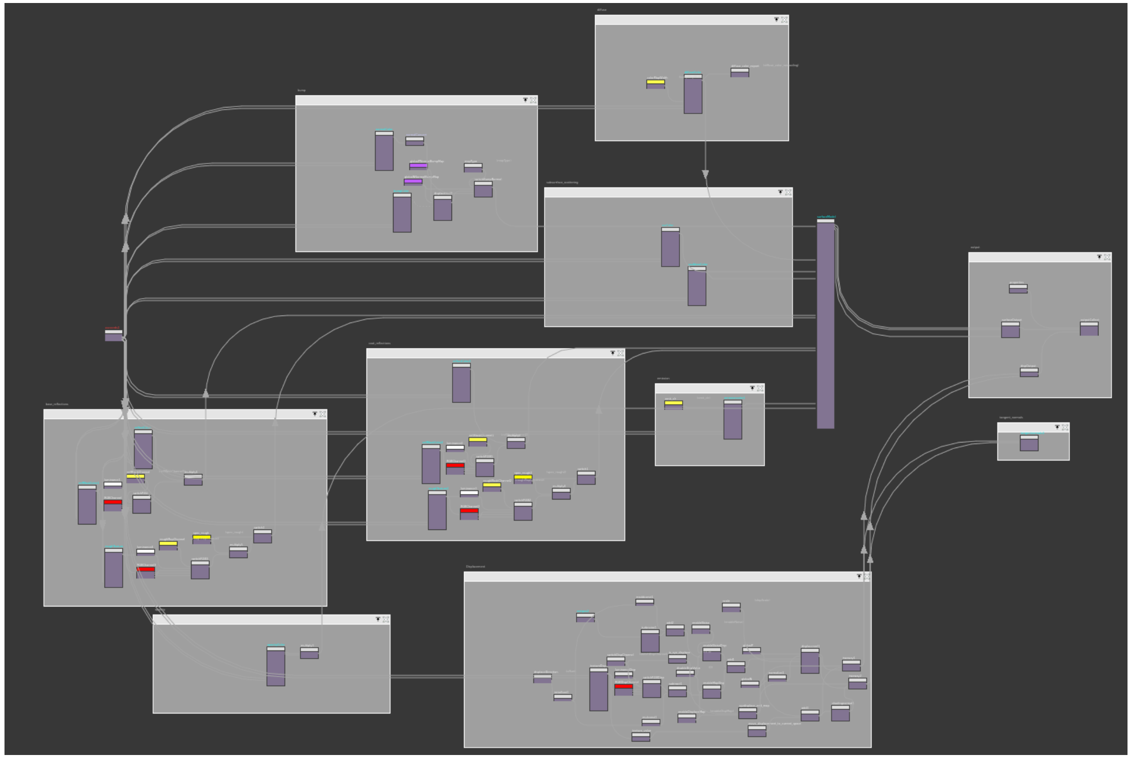
3D Animation Software Packages will come with a **Library** of **pre-configured Materials** often (but not always) derived from a standard material. This is the current list of Materials from Houdini's Material Palette.



Whether or not a Library Material is derived from a Standard Material is determined by its function and the complexity of its interaction with scene lighting.

INSIDE A MATERIAL

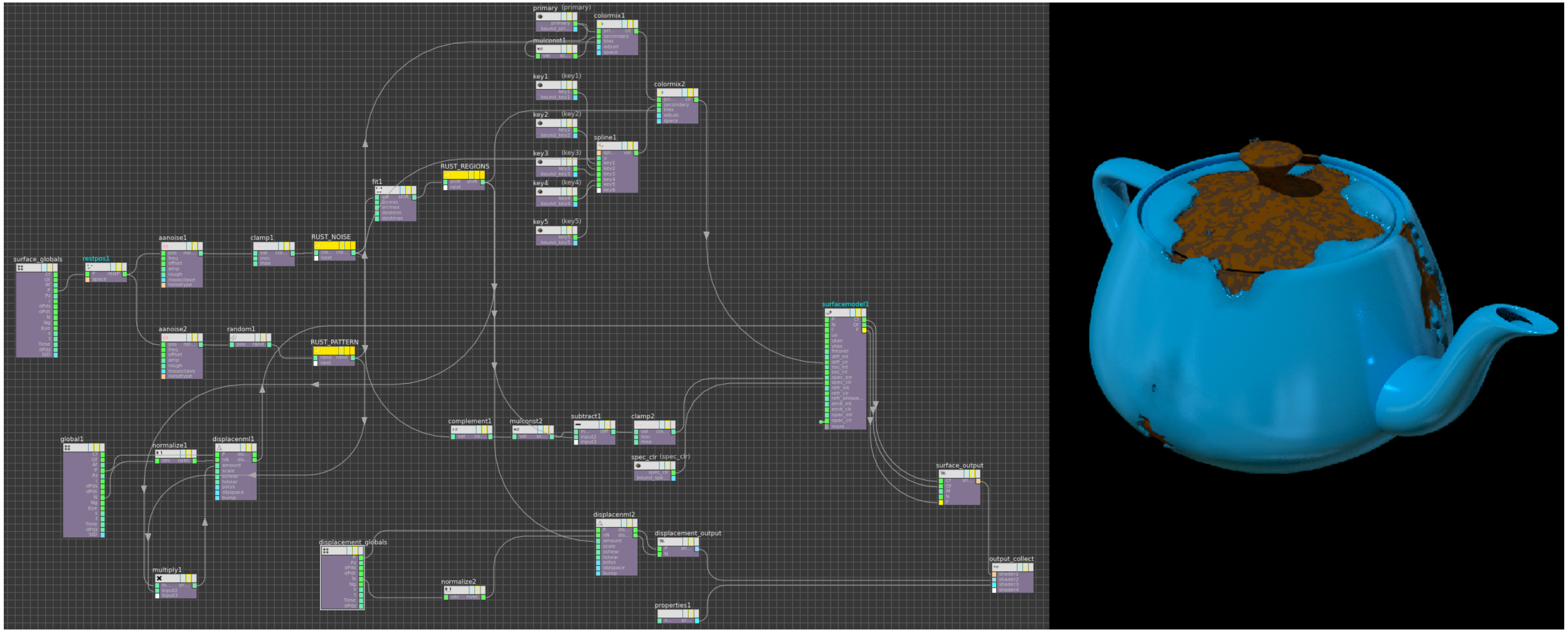
Inside a Material is the **Shader Level** - the material engine, called at render time. This is a **computer science orientated level** (containing further sub-levels) where **mathematical functions** are **combined** to give both **texture** and **surface lighting model calculations**.



In the context of a Standard Material, or a pre-configured Library Material, an end user does not generally need to access this level. The Shader Level does however also allow for bespoke custom materials to be developed.

CUSTOM SHADER LEVEL MATERIALS

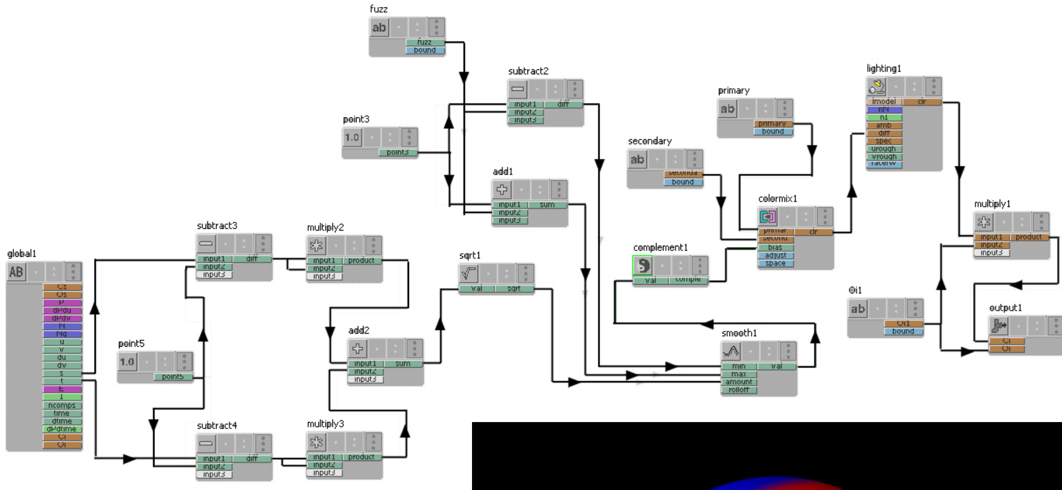
Bespoke shader level networks can be developed inside an **empty Material object**. The **principles** of **Shader Writing** are very similar to **compositing** in **Nuke**. In this example, a **custom shader network** has been developed to create a **rusty metal surface**.

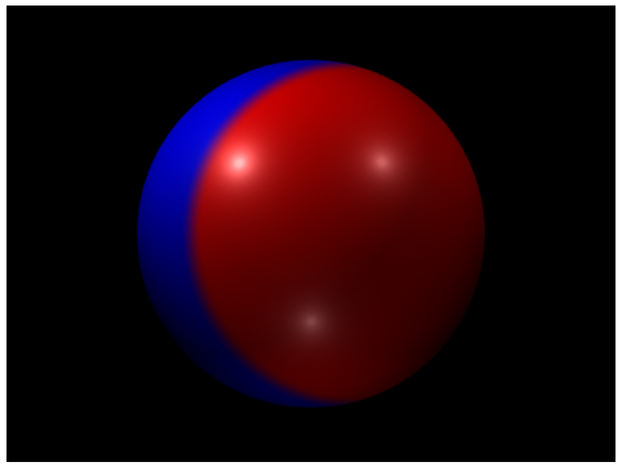


In **Houdini**, **Shader Writing** is **Renderman compliant**. This means **Renderman Shader Writing code** can be **translated** into **node networks**. Similarly, Renderman Shaders can also be developed as node networks in Houdini if Renderman is chosen as the renderer rather than Houdini's Mantra.

Custom Shaders can also be **created externally** by describing the surface through a **shading language**:

```
surface disk (
        color base = color "rgb" (1,0,0);
        color spot = color "rgb" (0,0,1);
        color spec = color "rgb" (1,1,1);
        float fuzz = 0.025;
        float Ka = 1;
        float Kd = 0.5;
        float Ks = 0.5;
        float roughness = 0.1;
        )
{
normal Nf = faceforward(normalize(N),I);
vector V = -normalize(I);
color Ct = color "rgb" 1;
float dist;
float inDisk;
dist = sqrt((s-0.5)*(s-0.5)+(t-0.5)*(t-0.5));
inDisk = 1 - smoothstep(0.3 - fuzz, 0.3 + fuzz, dist);
Ct = mix(base,spot,inDisk);
Oi = Os;
Ci = Oi * (Ct * (Ka * ambient() + Kd * diffuse(Nf)) + spec * Ks * specular(Nf,V,roughness));
}
```

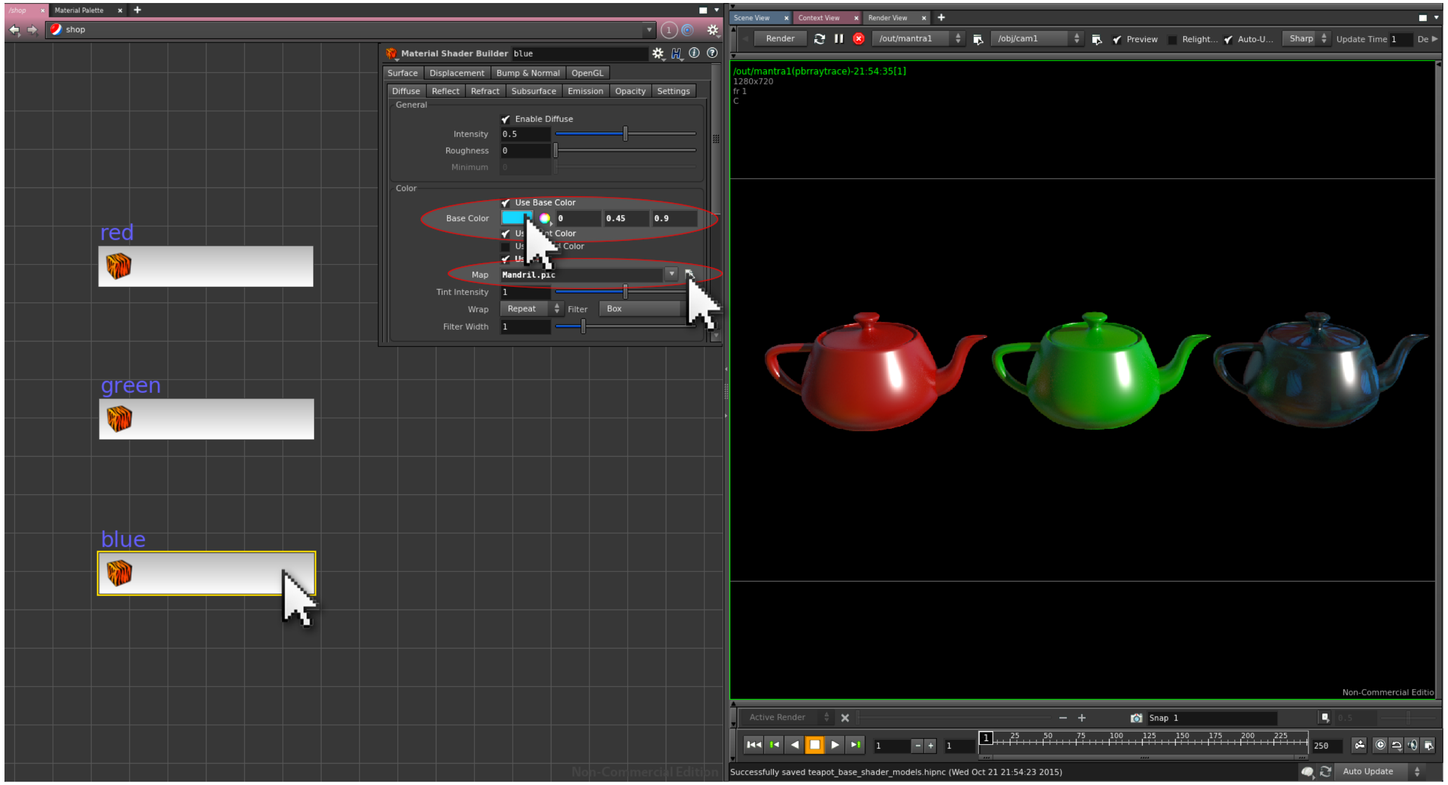




In this example, both the code and the network describe a simple procedural (Renderman) shader for creating a coloured circle on a coloured background. This shader was written by the NCCA's Dr Ian Stephenson as an example for his book 'Essential Renderman'.

MATERIAL SURFACE PROPERTIES – DIFFUSE

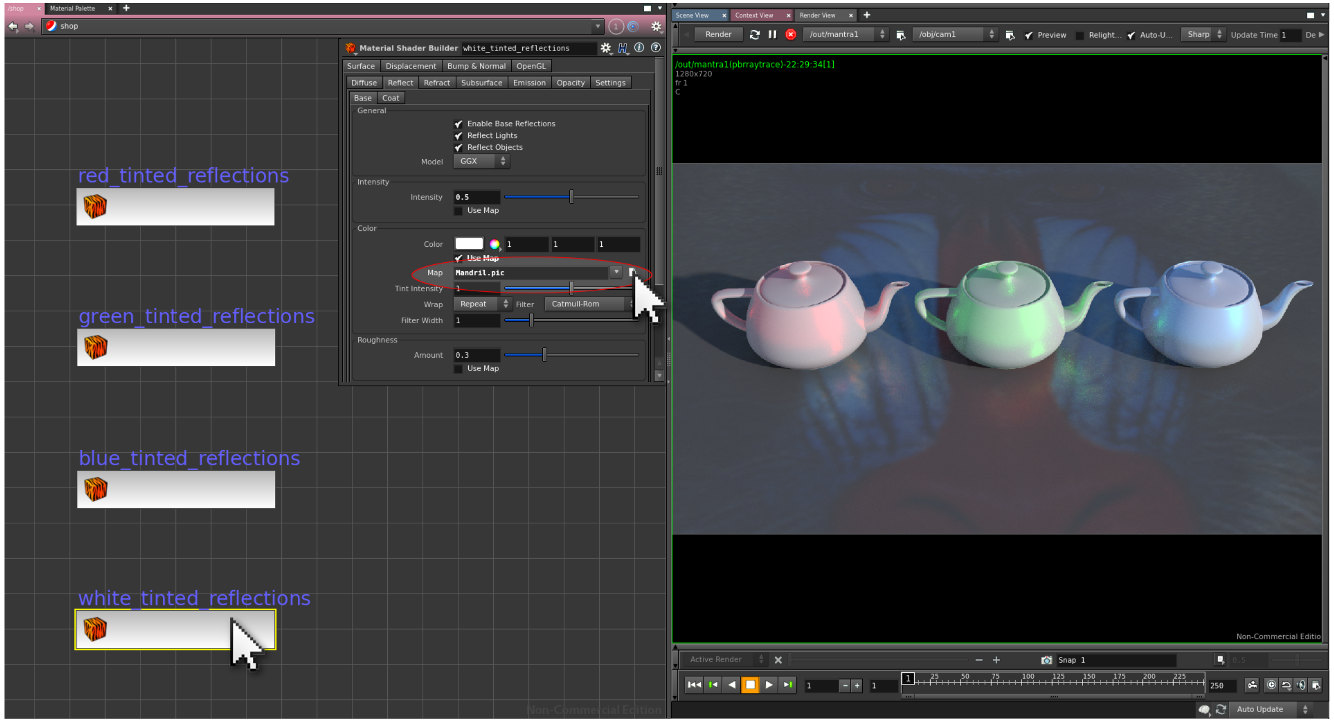
Material Surfaces can be set to return **different colours**, **images** or **both**, as part of their Surface > **Diffuse parameters.** If **both** an **image** and a **surface colour** are specified, the **colour** and the **image** are **multiplied together**.



In this example, **three standard materials** have been assigned **red**, **green** and **blue** as their **Diffuse Base Colours**. The **blue material** also has the Houdini **Mandril test image** assigned as a Diffuse **Color Map**, resulting in the **blue surface colour** and the **Mandril image** being **multiplied together** in the render.

MATERIAL SURFACE PROPERTIES – REFLECT

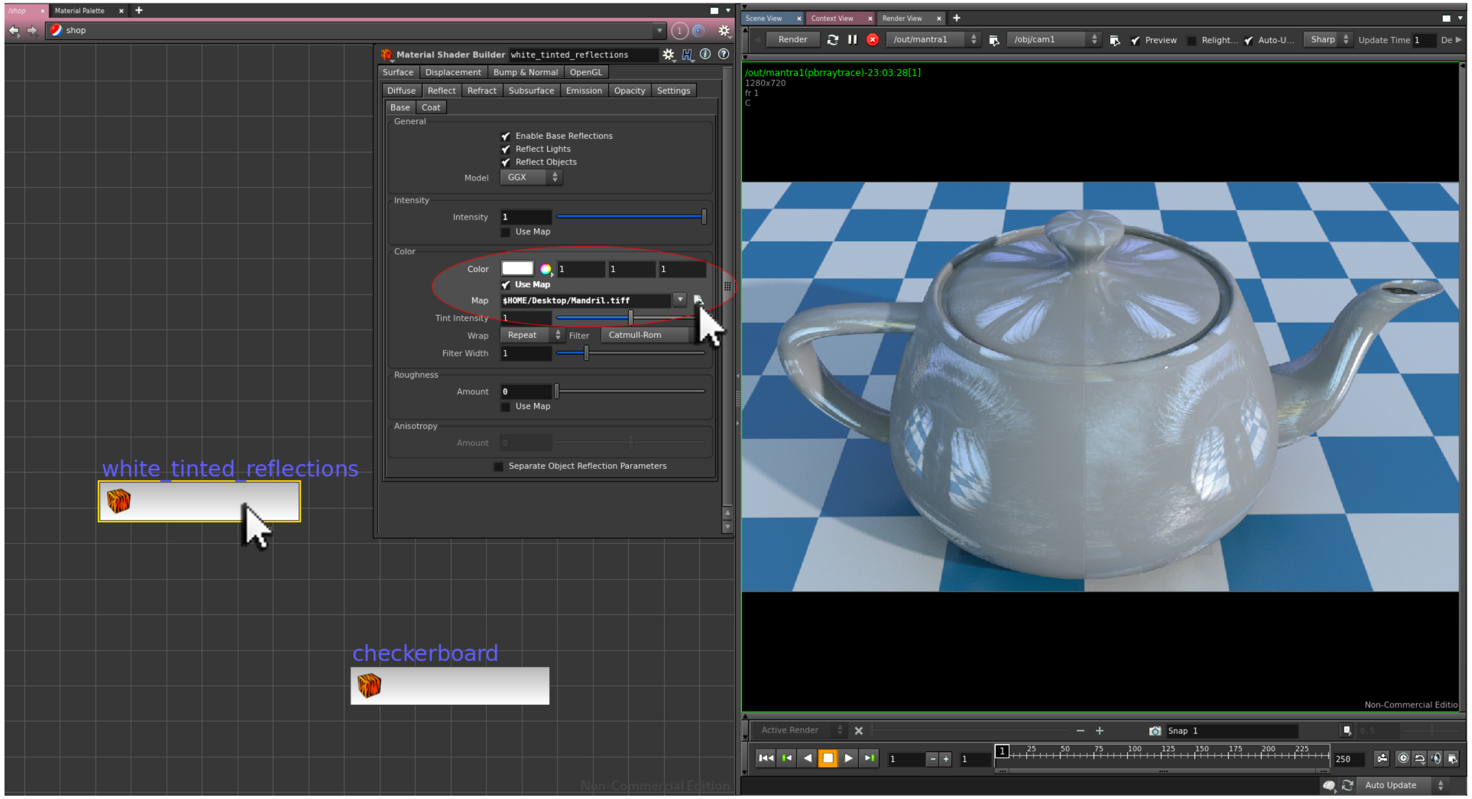
Material Surfaces can be set to return **different reflection colours**, **images** or **both**, as part of their Surface > **Reflect parameters**. If **both** an **image** and a **reflection colour** are specified, the **colour** and the **image** are **multiplied together**.

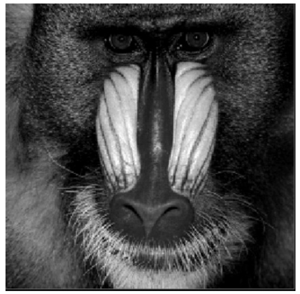


In this example, **three standard materials** have been assigned **red**, **green** and **blue** as their **Reflect Colours** and assigned to three teapots. The **ground material** also has the Houdini **Mandril test image** assigned as a **Reflect Color Map**, resulting in its **white reflection colour** and the **Mandril image** being **multiplied together** in the render.

MATERIAL SURFACE PROPERTIES – REFLECT

If a **black and white** (greyscale) **image** is used to **control reflection colour or intensity**, **black areas** of the image will **have no reflectivity** resulting in a **diffuse surface**; whilst **white areas** of the image **have maximum reflectivity**, becoming **mirrors** of other scene objects and lighting.

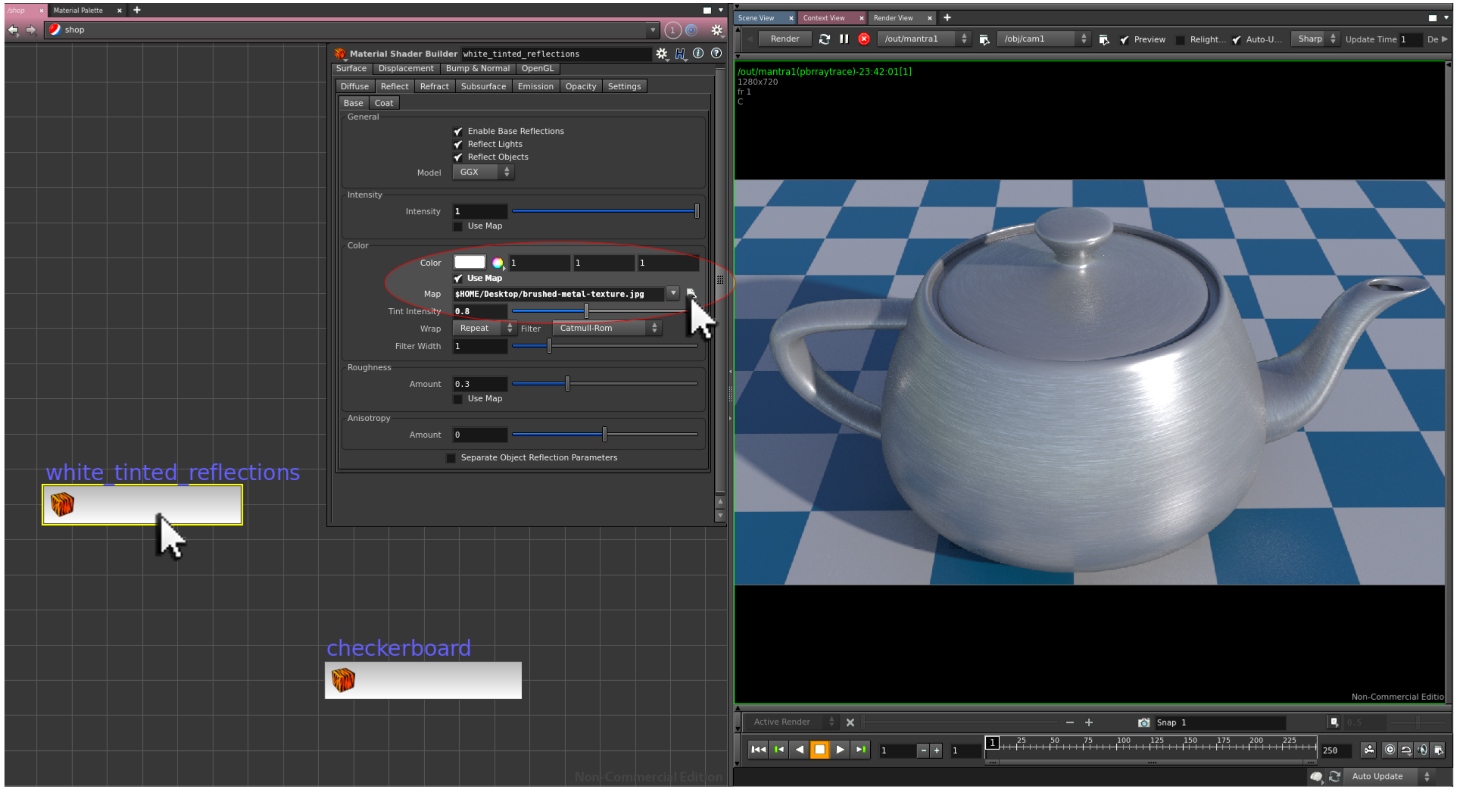


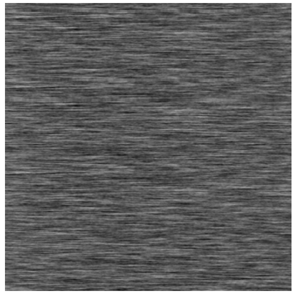


In this example, the greyscale Mandril image acts as a reveal mask controlling areas of the teapot that are mirrored, and areas of the teapot that are diffuse.

MATERIAL SURFACE PROPERTIES – REFLECT

The principle of having areas of a surface that return **high reflectivity** and other areas that **return diffuse** through the use of a **greyscale Reflection Map**, can result in **brushed metal** or **satin finishes** for surfaces.





In this example, a greyscale brushed metal image acts as a reveal mask controlling areas of the teapot that are mirrored, and areas of the teapot that are diffuse.

MATERIAL SURFACE PROPERTIES – REFLECT

The amount of **surface scattering** of **reflected light** can also be **controlled** by **adjusting** the **Reflect > Roughness parameter**.

Roughness 0

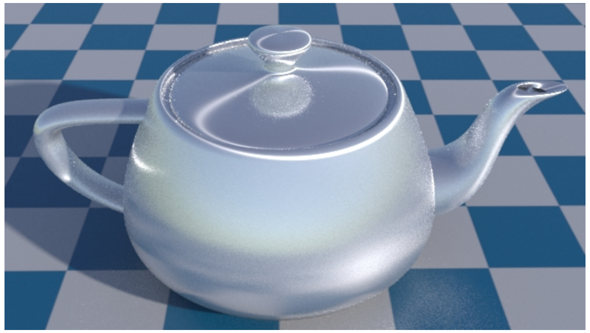
Roughness 0.5

Roughness 1

The **vertical** and **horizontal directions** of **reflected light** across a surface can also be **controlled** by **adjusting** the **Reflect > Anisotropy parameter**. This effect can further **enhance brushed metal** and **satin surfaces**.

Anisotropy -1

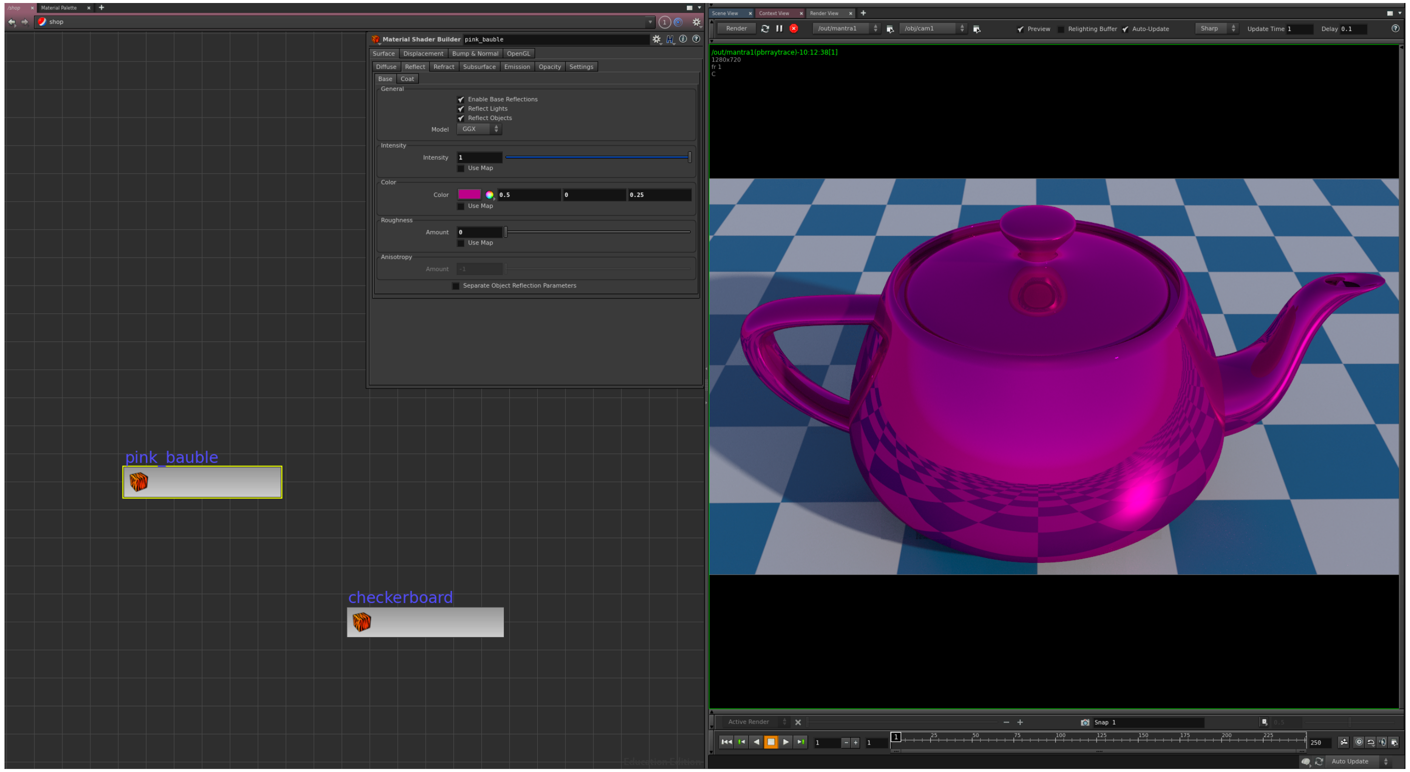
Anisotropy 0



Anisotropy 1

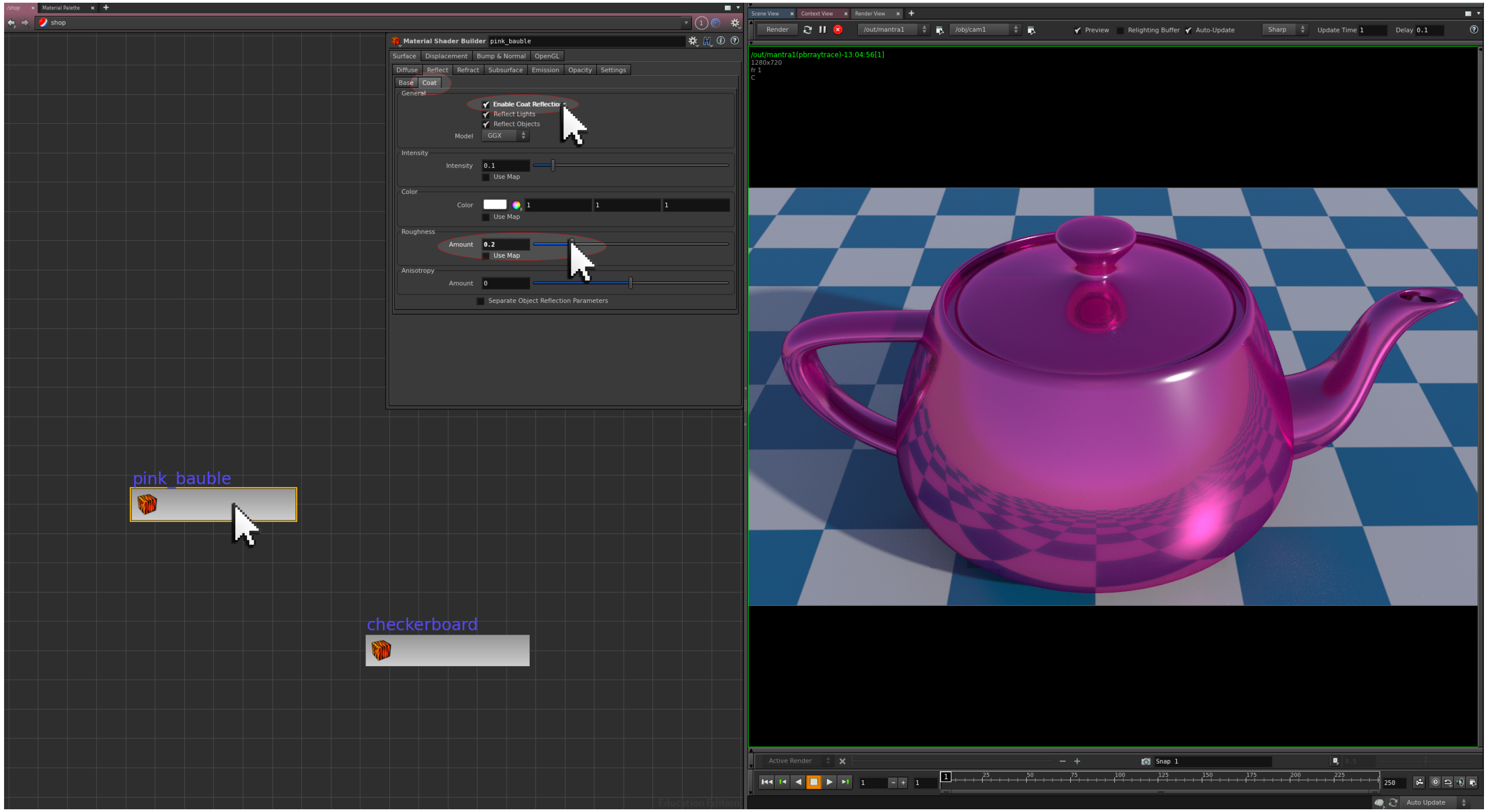
MATERIAL SURFACE PROPERTIES – REFLECT

Reflections can also be **layered**. In **Houdini** for example, **both** a **Base Layer** and a **Coat Layer** of **reflections** can be activated; **each with differing properties**.



In this example, **both** the **Diffuse Colour** and the (Base) **Reflection Colour** have been set to the same pink, and the **Reflection Roughness** reduced to **0**; creating a Christmas Bauble effect.

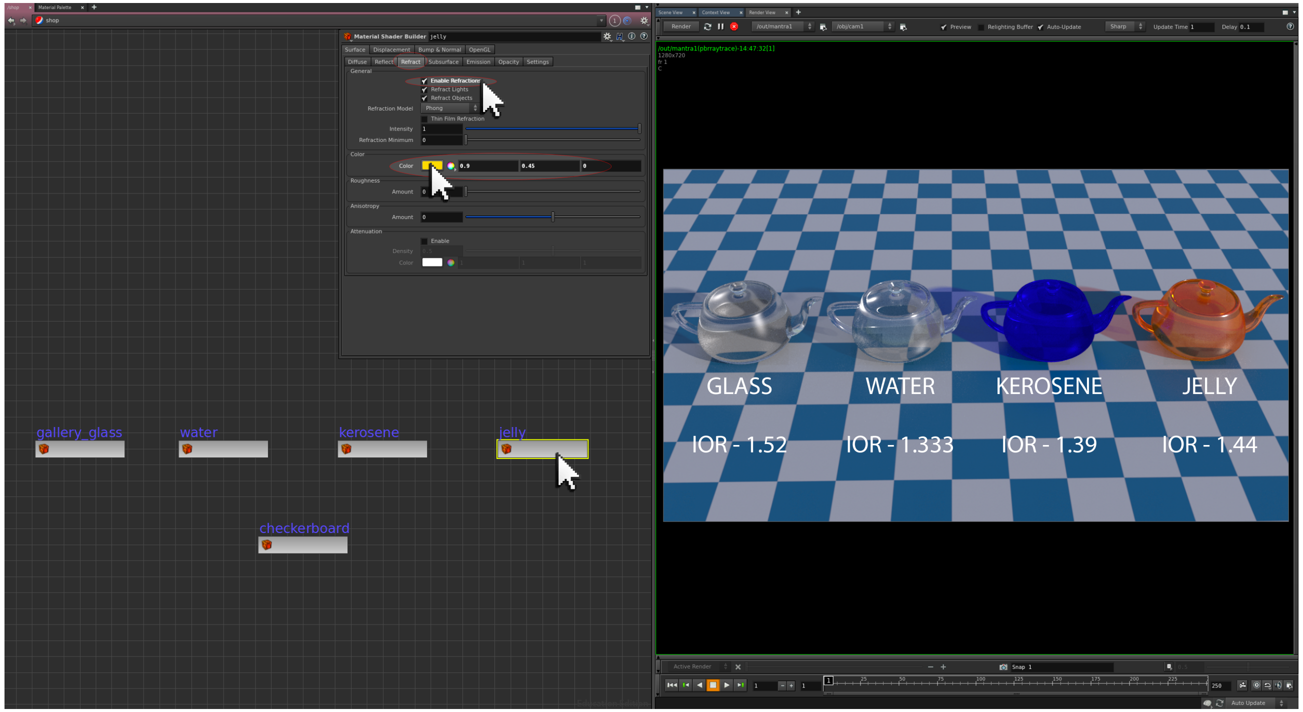
A **Coat Reflection Layer** is then **activated above** the Base Reflection Layer in order **to give white specular highlights** to the render.



The ability to activate both Base and Coat layered reflections can be **useful** for **high gloss coloured metals** or **glass**.

MATERIAL SURFACE PROPERTIES – REFRACT

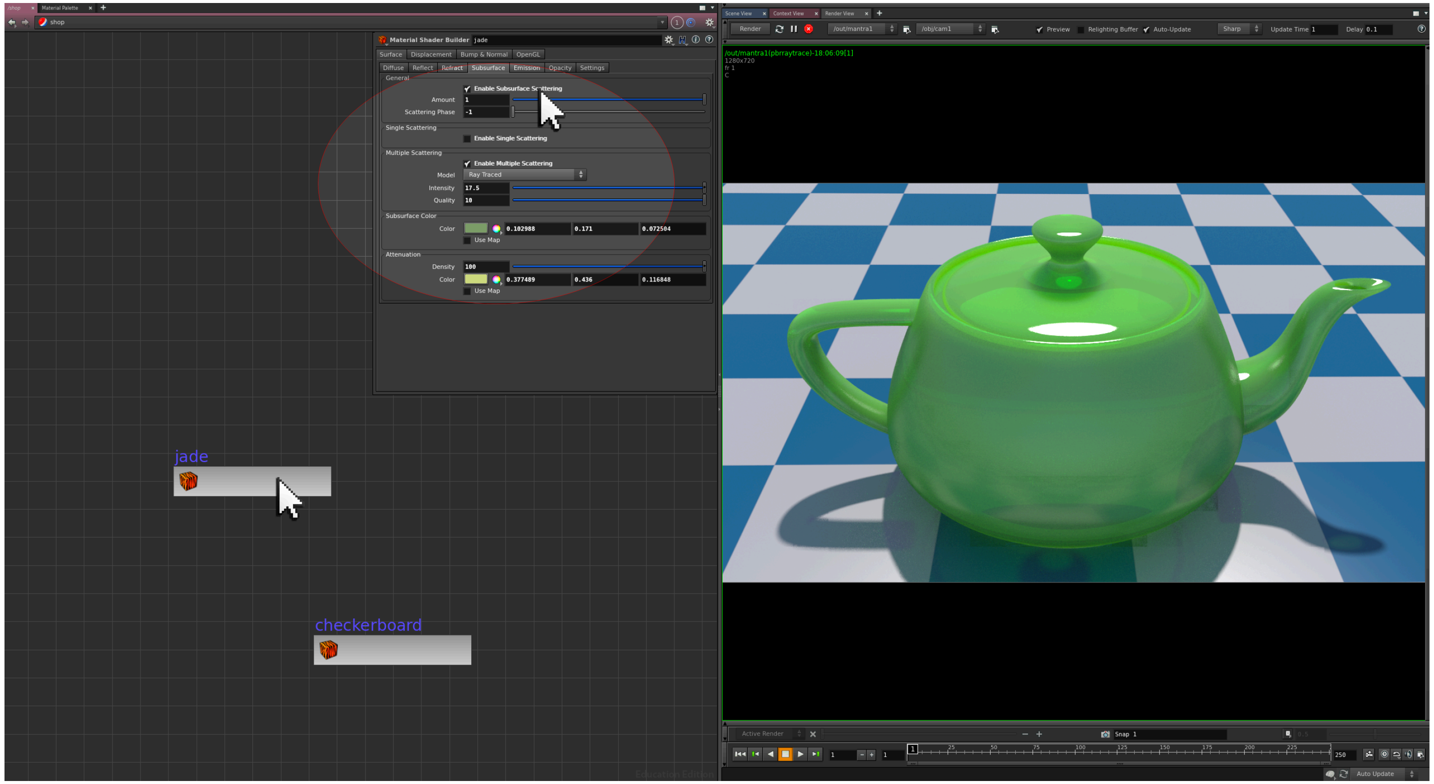
3D Animation Systems come with a **pre-configured Glass Material**; however, a **Standard Material** can also have **Refraction activated** to **bend light** through geometry. **Refraction** can be **activated alongside Reflection** to **create Glass, Water** or **Kerosene**; or with additional **Diffuse** to create **Jelly**.



As with other Reflection Surface Properties, **bespoke Refraction Colour**, **Roughness**, **Anisotropy** can be activated. **Attenuation** and **Faux Caustics** can also be activated. **All Refractive Materials** are able to specify an **Index of Refraction (IOR) value**, which varies slightly depending upon each material. **NOTE: IOR** and **Faux Caustics parameters** are hidden in the **Settings section** of the **parameters** in **Houdini's Standard Material**.

MATERIAL SURFACE PROPERTIES – SUBSURFACE

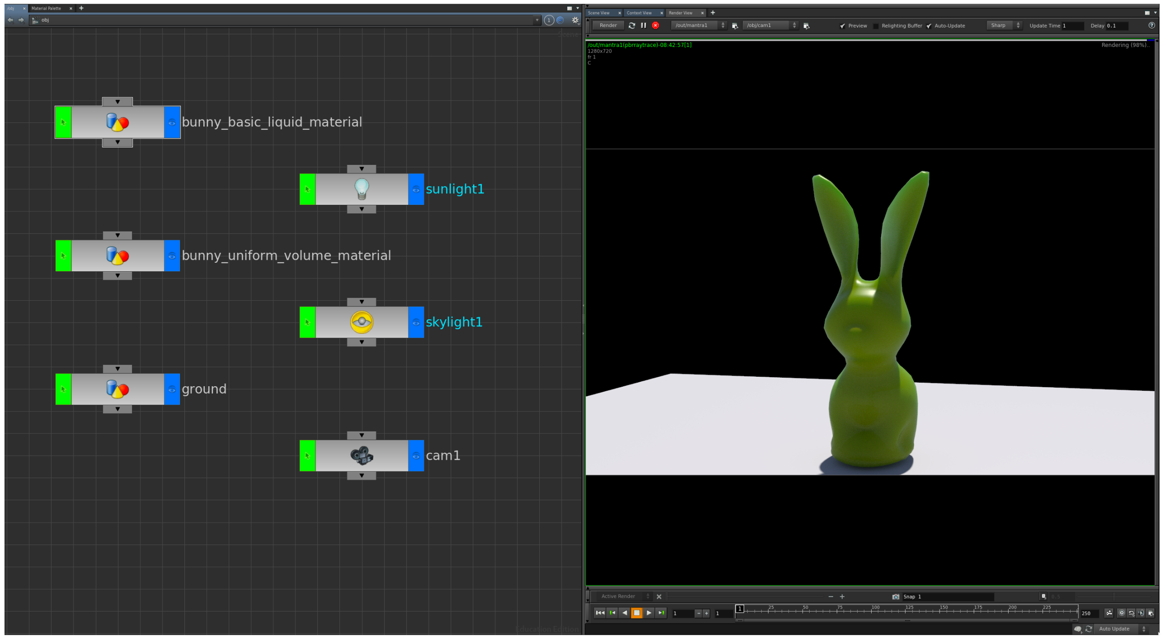
Refractive Materials such as **Water**, **Glass**, and **Jelly** can be **enhanced further** by the **activating Sub-Surface Scattering.** This is a Photon-Mapping technique where **light particles are scattered inside an object** to create a **cloudy internal aesthetic**. Sub-Surface Scattering can be **useful for creating** semi-translucent effects for **Milk**, **Skin** or **Jade Materials**.



The effect of **Sub-Surface Scattering** is **best observed** when **light is shining** directly **behind an object**.

SUBSURFACE SCATTERING ALTERNATIVE

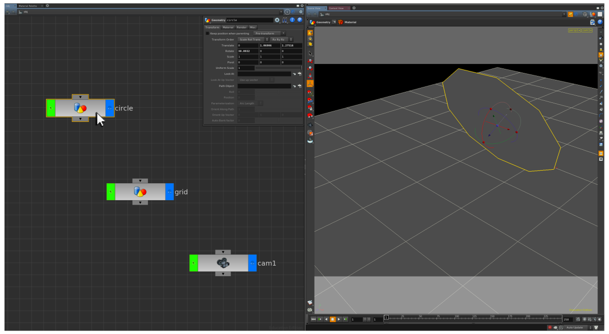
Activating **Material based Subsurface Scattering** can be a **costly render** expense. An **alternate method** of generating a subsurface scattering effect is to have **two instances of the same object**; **one rendered** with a **Volume Material** (to create the **internal subsurface density**) and the **second rendered** as the **external surface**.

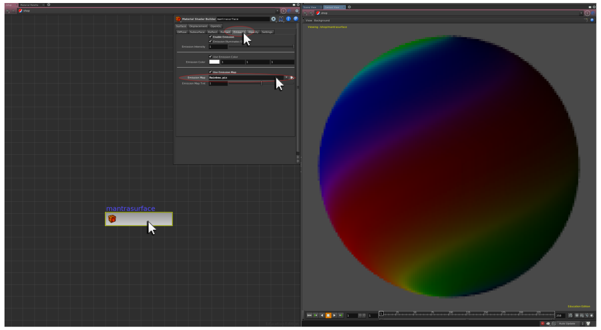


This alternate subsurface scattering method is formally used by Houdini when creating large volumes of water, but can also create **Jade** and **Ice** effects.

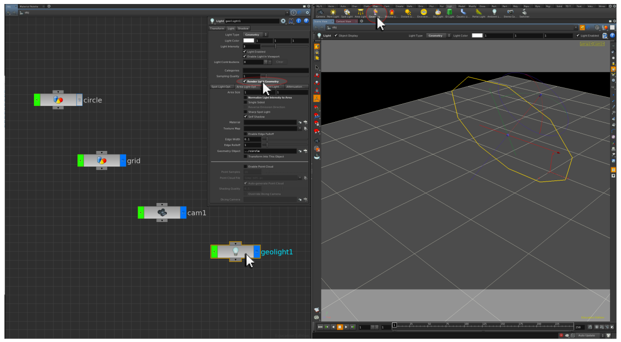
MATERIAL SURFACE PROPERTIES – EMISSION

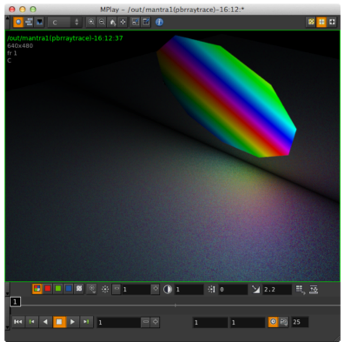
Emission is a property of a material that allows for texture information to be used for lighting effects. In this example, a simple scene is create with a grid and a circle angled above it.





At **SHOP Level** a Material is created with **Emission enabled** but all other material properties deactivated. A Rainbow image is assigned as an Emission Map.

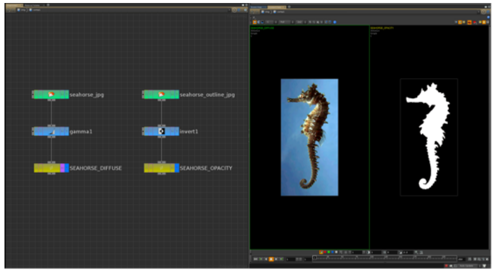


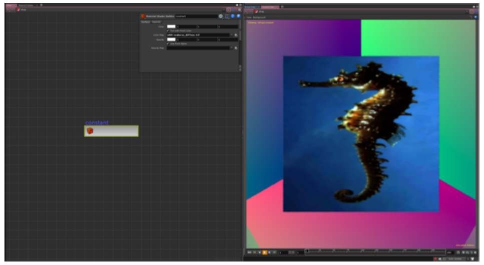


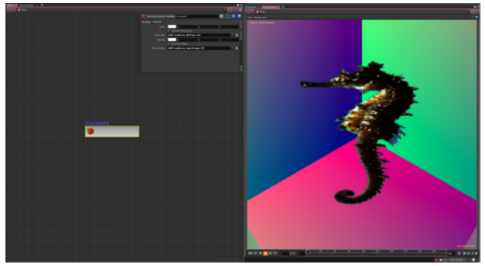
Back at **Object Level**, the **Emission Material** is assigned to the **circle**, and the **circle activated** as a **Geometry Light**. When the scene is **rendered**, the rainbow **texture emits light** and **illuminates the grid**.

MATERIAL SURFACE PROPERTIES – OPACITY

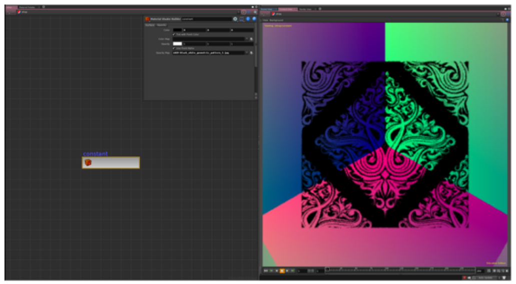
Opacity Maps are useful for **cutting out geometry** using a **black and white image**. In this example, an **Opacity Map** is created for a **seahorse image**, where **black** represents the **areas to be cut away** at render time, and **white** representing **the areas to keep** at render time. When initially the seahorse diffuse texture image is assigned to a constant shader, its blue background is still visible. When the black and white opacity map is added, the blue background is removed.

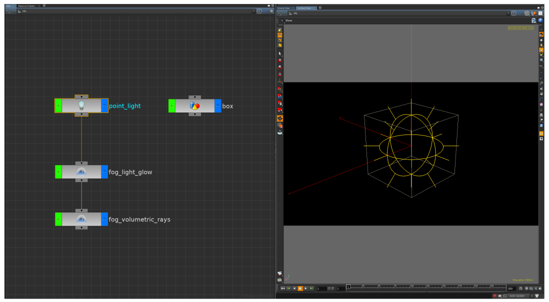


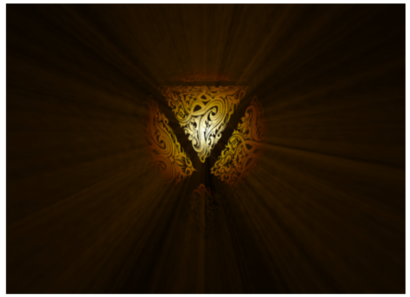




Opacity Maps can also be used to **cut out complex patterns** from geometry. A **decorative pattern** is **assigned** as an **Opacity Map**, then applied to a simple box with a light inside it (with fog light glow and volumetrics also added).

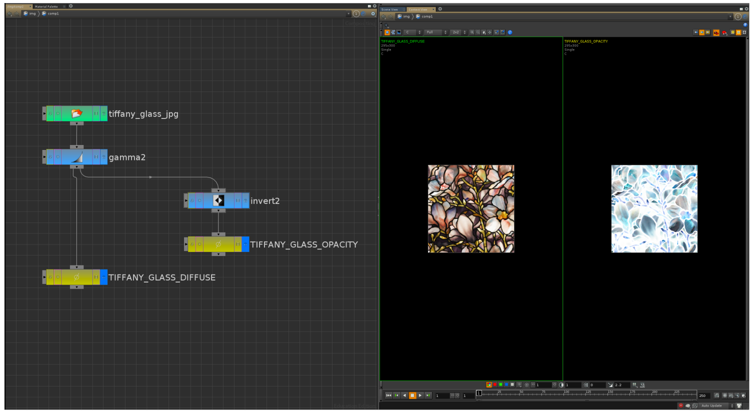


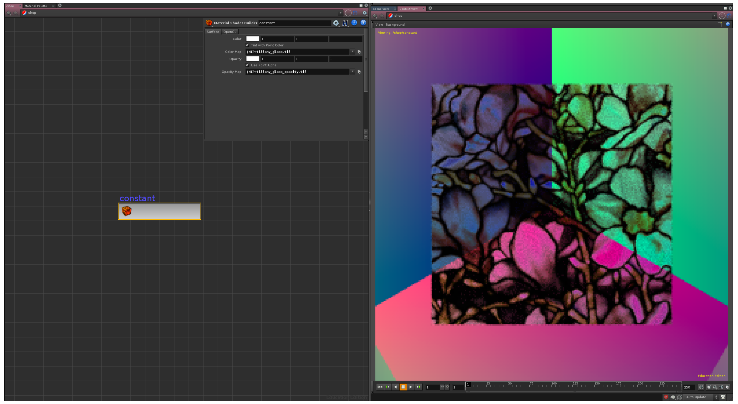


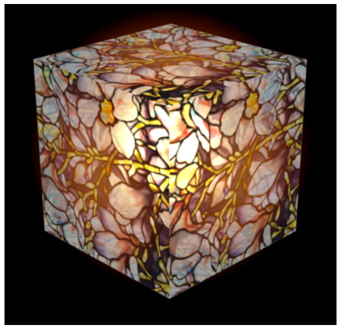


MATERIAL SURFACE PROPERTIES – OPACITY

Opacity Maps also **work with colour information**. In this example a **Tiffany Glass pattern** texture map is **inverted** to create an **Opacity Map**. Both images are then **assigned** as **Diffuse** and **Opacity** inputs to a **Constant Material**. The material can be assigned to a simple box with a light glow inside it.



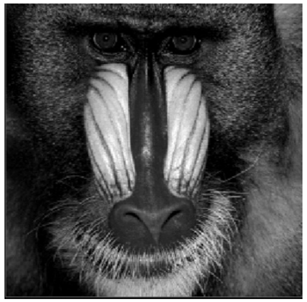


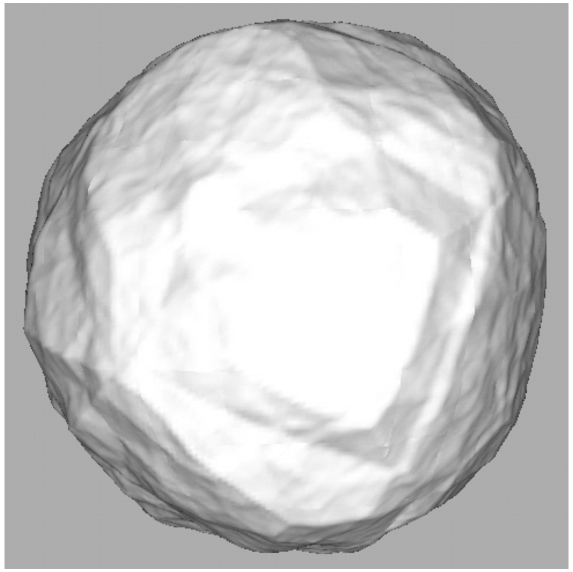


NOTE: Other types of Materials other than Constant can be used for Opacity Mapping...

MATERIAL DISPLACEMENT PROPERTIES

Rather than modelling specific details of a surface, these **details can be added to geometry** at **render time** using a process known as **displacement**. This is where the points of the surface are moved in accordance to a **procedural noise**, **pattern**, or a **painted image**. The **Displacement parameters** can control the **height of displacement** assigned to the geometry, **and whether white, black or grey displacement** information is used as the basis for pushing the geometry out or in.



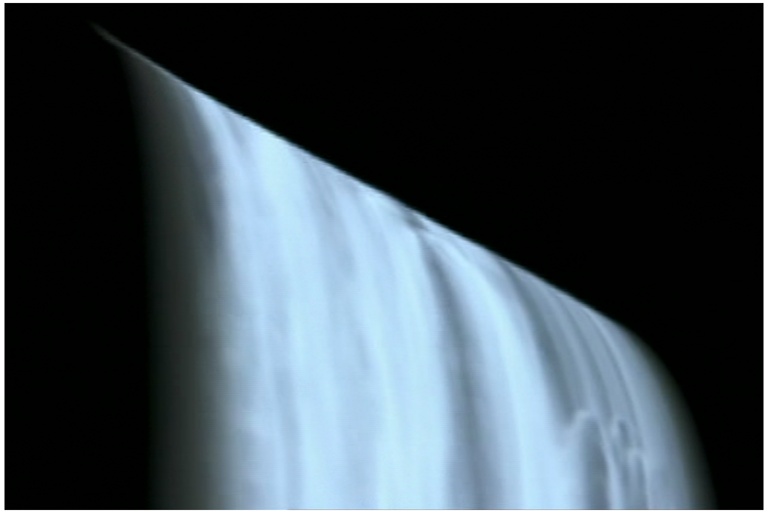


In the **first example**, a **greyscale Mandril image** is used to **create displacement** on a **grid** at render time. In the second example, procedural Displacement noise has been assigned to a sphere using a Standard Material, creating an undulating terrain effect at render time.

NOTE: At render time, **Displacement** will be **calculated first**, **before** the formal **shading** the the surface takes place.

MATERIAL DISPLACEMENT PROPERTIES – PRODUCTION EXAMPLE

In **Star Wars Episode 1** – **The Deleted Waterfall scene** reveals the use of **Displacement**. For **creating** the **waterfall effect**, **Acquisition Footage** of **falling salt filmed against a black screen** was **layered together** as an **animated texture**. This was then **UV projected onto a grid**, and **deformed** to **create the lip of the waterfall**.

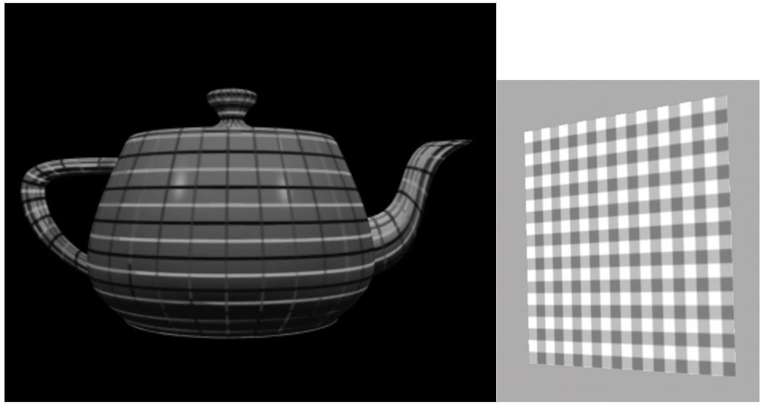


The **same footage** was then **processed** to become an **animated Displacement map**, **creating** the **shape and form** of the waterfall. This **helped bring additional realism** to the waterfall effect.

MATERIAL DISPLACEMENT PROPERTIES – BUMP MAPPING

Displacement Materials can also be **set** to only **return the displaced surface normals**. **Surface Normals control** a **surface response** to **light**. When a **surface is displaced**, **both** its **geometry point position** and **surface normals** are **recalculated**. In **Bump Mapping only** the **surface normals** are **recalculated**. This **creates** the **illusion of a bumpy surface** without directly altering the shape of the geometry.

Bump Mapping is useful **for** any **surface detail** that does not **overtly disrupt** the surface topology. As with Displacement, both images and procedural patterns can be used to drive a Bump effect.

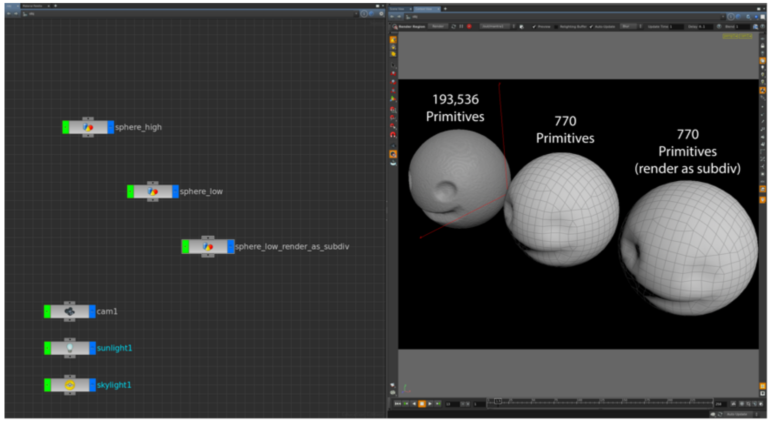


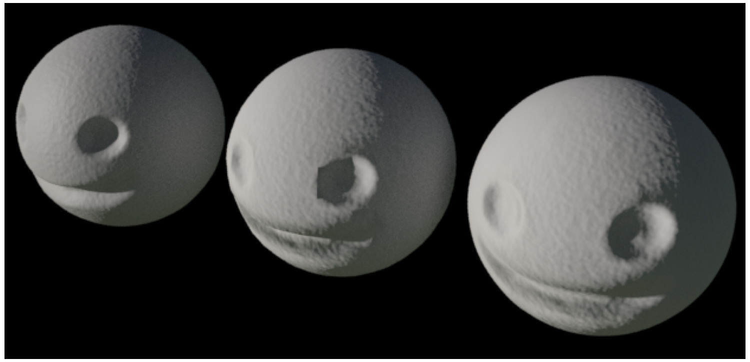
When the profile of a decorative real world clay vase is examined, the decoration does not interfere with its shape. Recreating this object in Computer Graphics could result in Bump Mapping being used over Displacement due to this surface phenomena. The teapot bump mapping effect exampled uses a Gingham Check Procedural Pattern to disrupt its Surface Normals without changing the shape of the teapot itself.

MATERIAL DISPLACEMENT PROPERTIES – NORMAL MAPPING

A variant on Bump Mapping is **Normal Mapping.** This is a technique widely utilised in Computer Games. This is a technique where **surface normal information** is **calculated** for a **high resolution mesh**, **saved** out **as an image**, and **re-applied** over a **low resolution mesh**.

In this example, a **high resolution sphere** was sculpted in **ZBrush**, and **exported** alongside with a **Normal Map and** a **low resolution** version of the **sculpt**. In **Houdini** the **Normal Map** was **assigned** to the **low resolution** version of the sculpt.





Normal Map from ZBrush

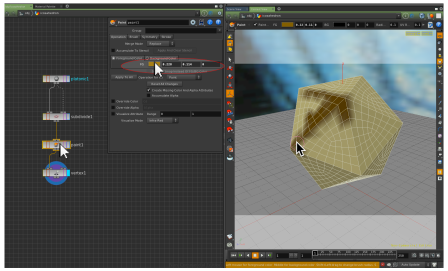
Assigned to a low res version of the sculpt model

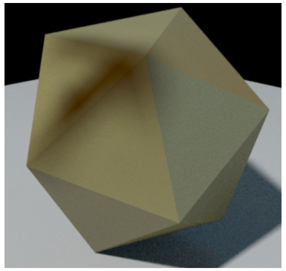
Rendered as a Subdivision Surface to smooth low res detail

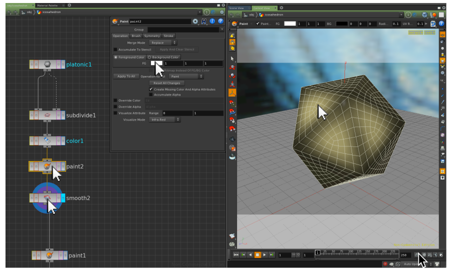
When rendered, the visual surface detail of a high resolution sculpt can be seen on the low res sculpt, but without the detail modelled into the geometry. The low res sphere is also rendered as a subdivision surface to smooth out the low res detail.

APPLYING MATERIALS TO GEOMETRY – SHADING EFFECTS THROUGH PAINTING

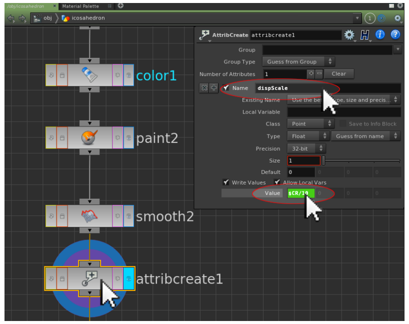
In this example, **Point Colour** has been painted onto an **icosahedron**. This **colour** is **automatically passed** into the **Material** at **Render Time**.

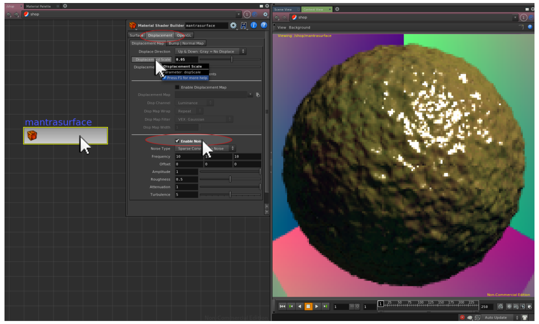


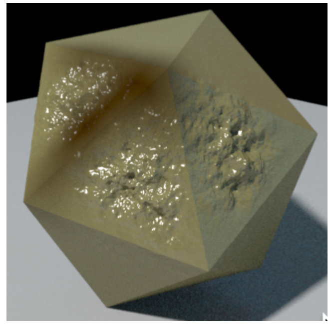




A **second layer of colour** is then **painted** onto the surface **temporarily overriding the original paintwork.** This **additional layer of colour** is then **stored** as a **Displacement Attribute** that the **Material** will **recognise**, and then the **original paintwork** is **restored. Noise based Displacement** is then **activated** on the **Material**. When the **scene** is **rendered again**, the **paintwork** from the **second layer** now **controls where** the **surface displacement takes place**.



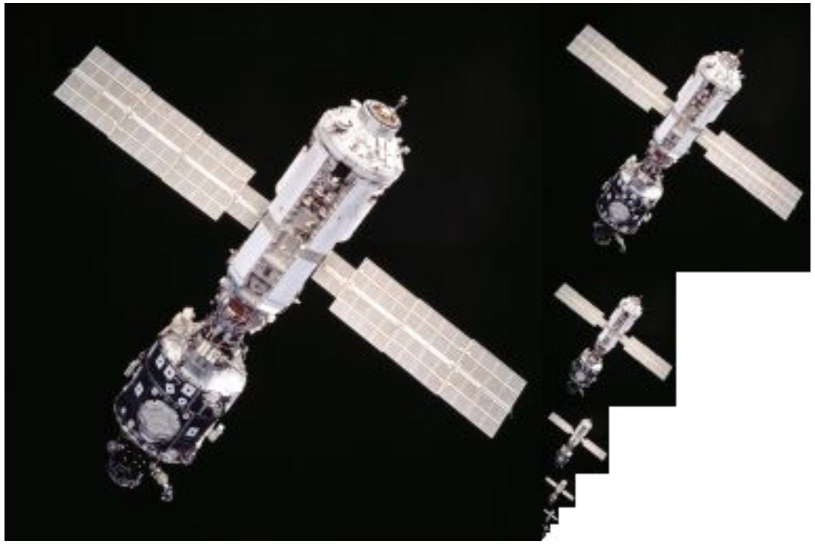




This **principle** of **painting geometry** to control **Diffuse** and **Displacement** can also be **applied** to the **Reflection**, **Refraction**, **Subsurface**, **Emission** and **Opacity properties of a Material**.

APPLYING TEXTURES TO MATERIALS - TEXTURE IMAGE FORMATS

While **3D animation systems** can handle most **standard image file formats (.tif, .jpg etc**); they will often have a **native image format** for textures. This is so textures can be internally optimised for rendering. These native image formats often include **MIP Mapping**, a technique where smaller versions of an image are stored with the full resolution image. These smaller images are called when the renderer does not need to compute higher levels of detail (for example if the object moves away from the camera).



(Tokigun, 2006)

The MIP Map format allows the renderer to access portions of the texture without having to load the whole image into memory at once.

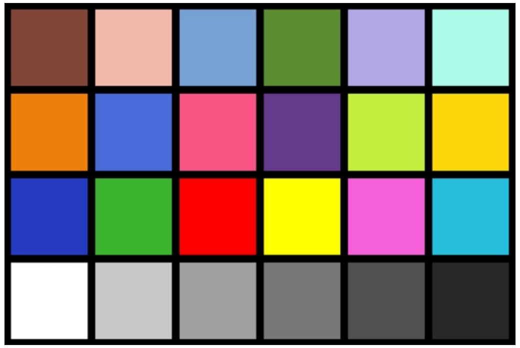
A **.rat file** (Random Access Texture) is the **native texture image format** for **Houdini** for texturing within this particular piece of software.

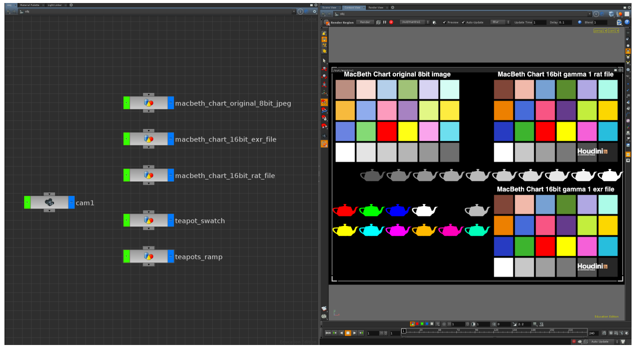
Similarly, a **.tex file** is is the **native texture image format** for **Renderman** for texturing within this particular piece of software.

APPLYING TEXTURES TO MATERIALS - GAMMA CORRECTING 8 BIT TEXTURES

http://www.cambridgeincolour.com/tutorials/gamma-correction.htm

Gamma is a method of **encoding 8 bit images** so that they look visually correct. **8 bit images** images sourced online for example have a gamma value embedded into them. As **rendering images for compositing** is done at **16 bit** (linear rendering and compositing), this can cause **colour discrepancies** when **8 bit images** are **assigned** as textures and **rendered linearly** at **16 bit**. An **8 bit Macbeth Colour Chart** can be used to test the **rendering workflow**:

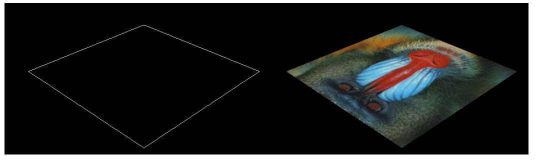


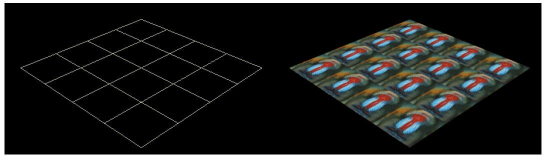


When **rendering linearly** for compositing, the **8 bit Macbeth Chart will appear washed out** rather than its proper colour values. If, however the 8 bit Macbeth Chart has its **gamma value removed**, and **re-saved as a 16-bit image**, it will appear correctly in the render. **NOTE: Houdini** now **internally resamples 8 bit images** to **16 bit images automatically**.

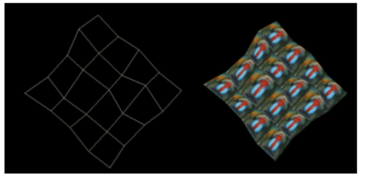
APPLYING MATERIALS TO GEOMETRY – TEXTURING SIMPLE GEOMETRY

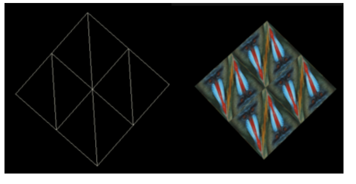
If a **single polygon** is **textured** with a **Material**, the **texture map** will be **applied** to the **whole face** in accordance with the UV Square. Similarly, if geometry consists of **multiple polygon faces**, **each individual face will be textured**.

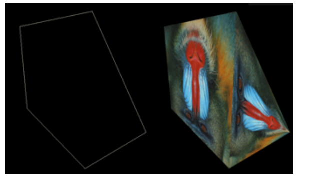


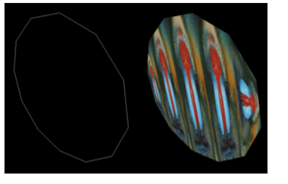


This is the **same for irregular shaped polygons**, as well as **triangles**. A **Pentagon will** however **assign the image twice**, once **as a square and** secondly as a **triangle**. A **Dodecagon** (12 sides) will be **divided into quads**, with the **image assigned five times**.

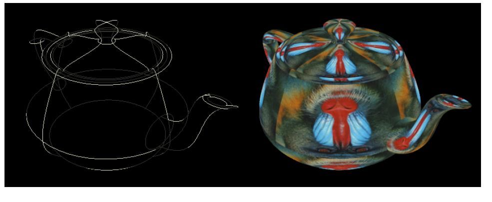


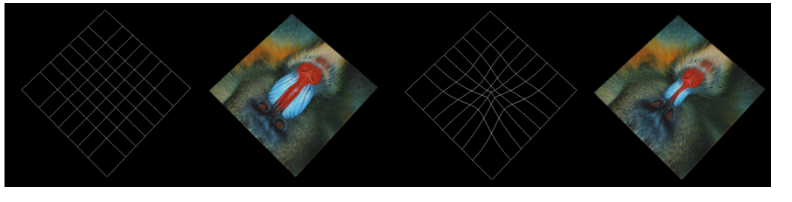






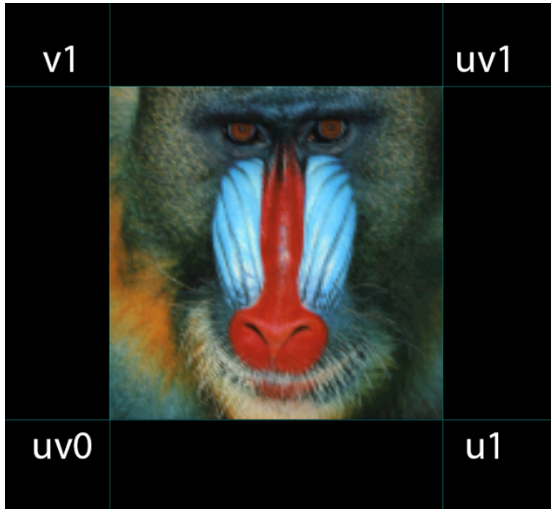
A **NURBS** or **Bezier Patch** will also be **textured** the **contents** of the **1x1 UV Square**. A **Bezier Utah Teapot** contains **33 separate patches**. **Each** will be **textured individually**. The amount of **inner geometry** of a single **NURBS** or **Bezier patch** will not affect this default texturing behavior, but **will influence where on the surface the image is drawn**. **If** for example **internal geometry** of a NURBS or Bezier patch **is modified**, this can **result in pinching**.

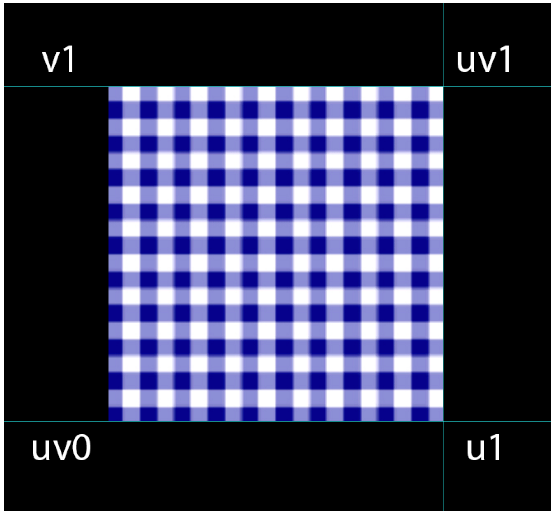




APPLYING MATERIALS TO GEOMETRY – THE UV SQUARE

All **painted** and **procedural** Materials in a 3D Animation system are generated and developed relative to a **1x1 2D square** known as **the UV Square**.

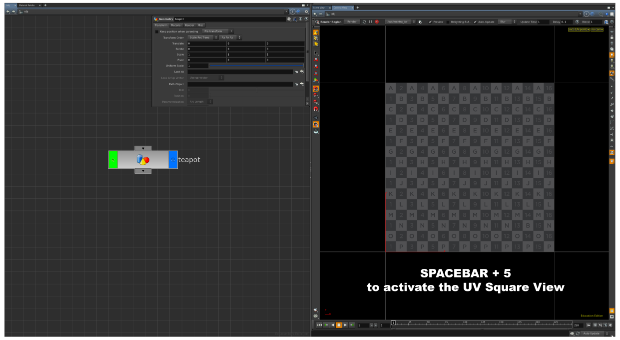


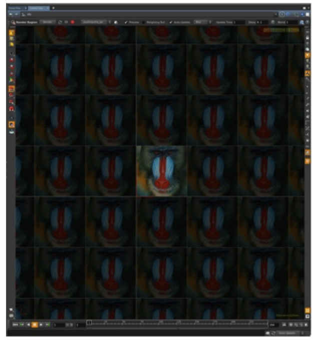


- **Images for texturing** should therefore **always** be in a **square format**, **and** be **72 Dots Per Inch** (DPI) **Resolution**; as the resolution of the image when applied to geometry is determined solely by pixel width and pixel height
- The **pixel width** and **pixel height** of a **texture image**, always be **Power of 2 numbering** (**256x256**, **512x512**, **1024x1024**, **2048x2048 etc**).
- Be aware that **the larger the pixel width and height**, the **longer** it will take **to render**.
- **Test** the **pixel width** and **height** of a **developing texture relative** to **geometry proximity to camera** to help **gauge** the **optimal texture size**.
- **Reflection Maps** and **Displacement maps** should be **greyscale** to **minimize** file size.

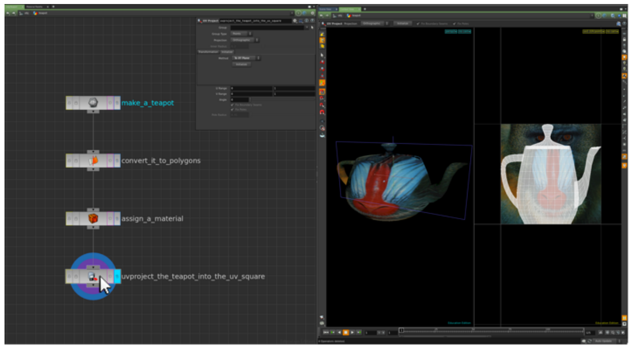
APPLYING MATERIALS TO GEOMETRY – THE UV SQUARE

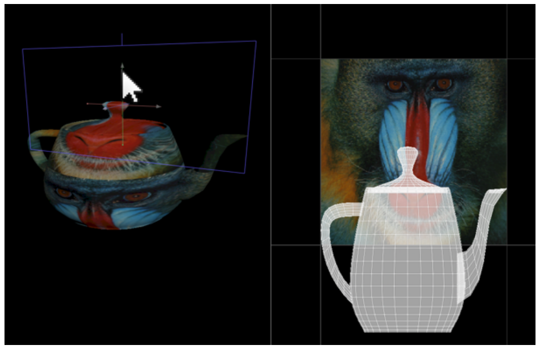
The **UV Square** sits within an '**infinite' tiled area** know as **UV Space**. **Textures** can be **loaded** into the **UV Square** in order t**o help projecting and editing UV layouts** of geometry. **Textures assigned** to the **UV Square** are also set to **tile repeat** in **accordance with UV Space**.

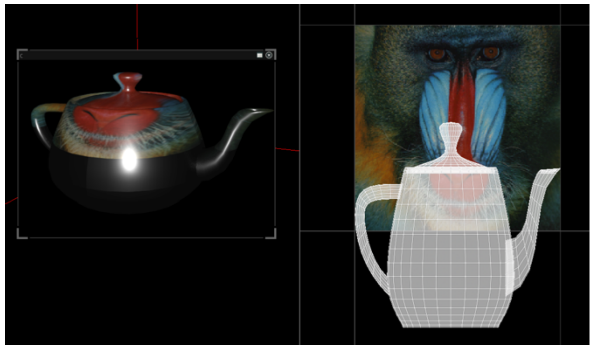




Geometry can then be **projected into the UV Square**, and whatever **part of the image** the projected **geometry aligns with gets textured** accordingly. If a **geometry projection goes outside** the **UV Square**, then **this part of the geometry** will be **textured using the next UV tile** instead **unless this** repeat **behavior** is explicitly **deactivated** in the **Material parameters**. The success of the projection determines the success of the texturing on the object.

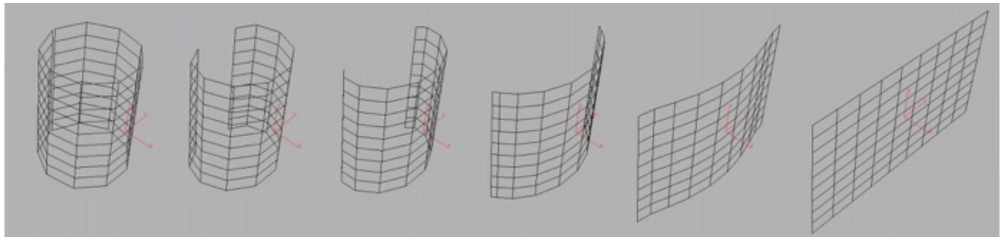




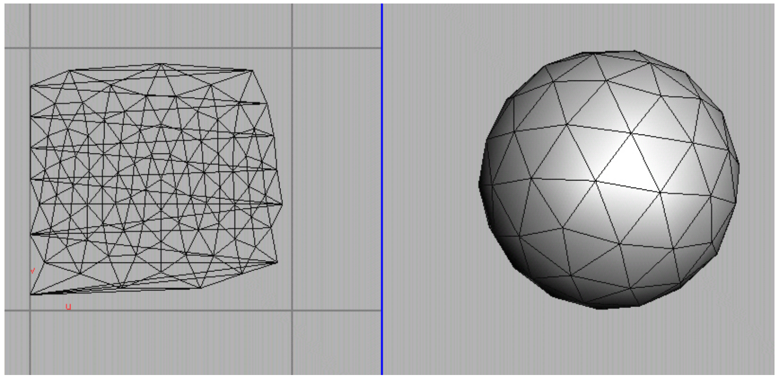


APPLYING MATERIALS TO GEOMETRY – POINTS VERSUS VERTICES

A **UV Projection** is a way of **unwrapping geometry cleanly** onto a **flat plane**. If for example a **tube** is **unwrapped** into **UV space**, a **cut line** is **determined** automatically **by the software** (**or** sometimes **manually by the end user)**, and the **geometry is flattened accordingly**.

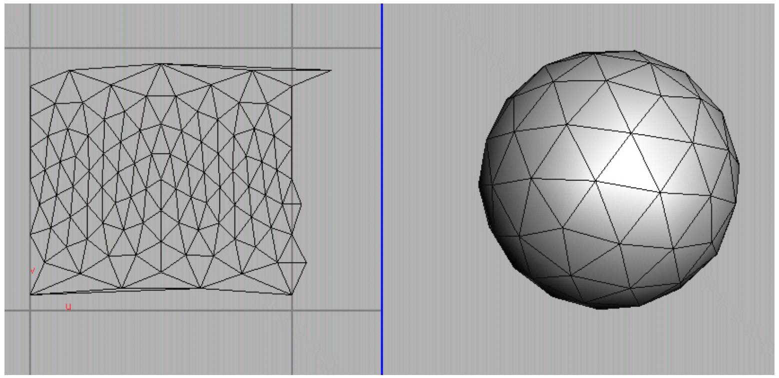


When **geometry** is **projected** into **UV Space**, a **seam along** the **geometry edges** is specified **to allow for unwrapping** of the **3D geometry** into **2D UV Space**. If the **projection geometry type** is specified as **Points**, the **edges** of the **seam** will **remain connected**. If the **projection geometry type** is specified as **Vertices**, the **edges** of the **seam** will be **treated as separate elements**.



A Sphere Projected as Points

This method will cause seam rendering artefacts

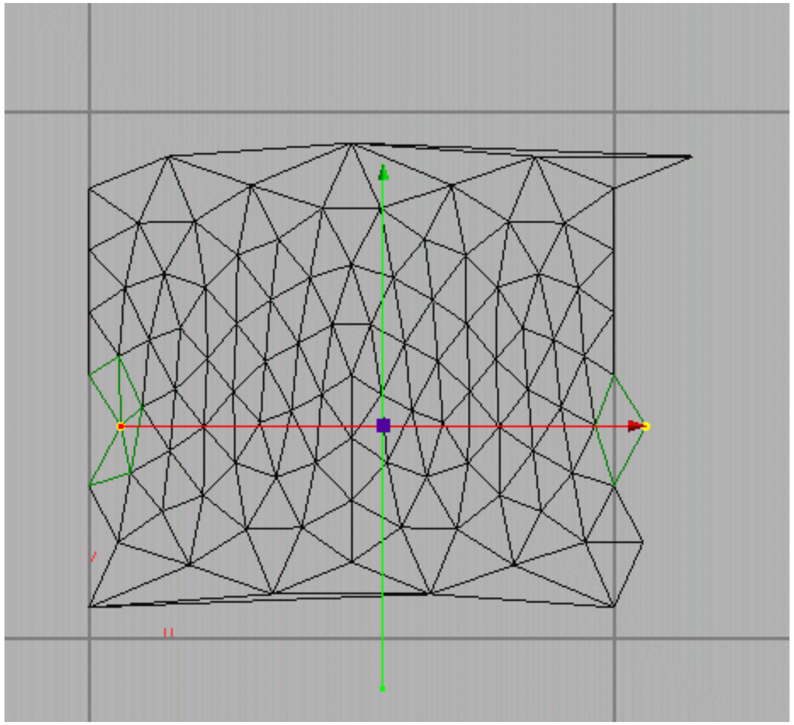


A Sphere Projected as Vertices

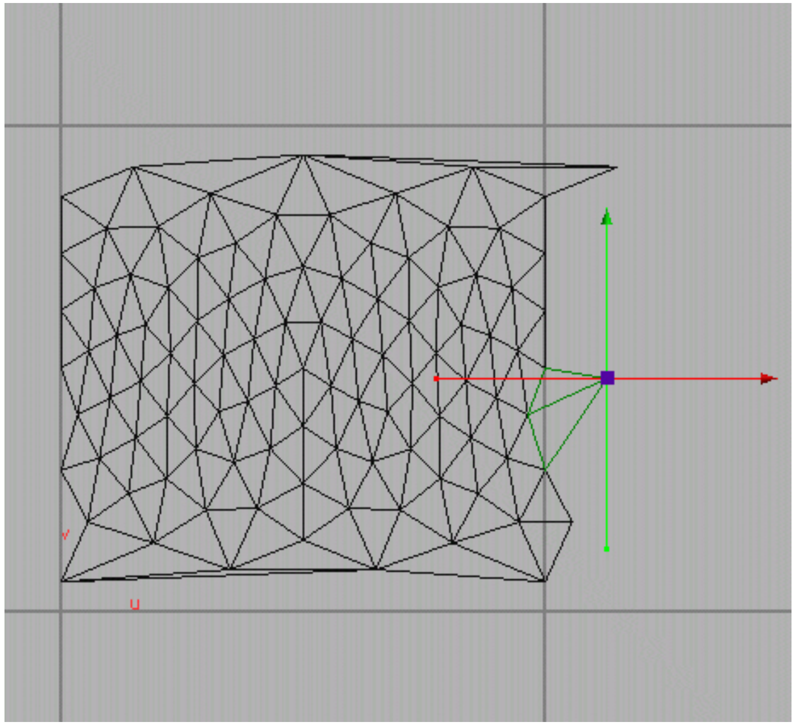
This method will create invisible seams

APPLYING MATERIALS TO GEOMETRY – POINTS VERSUS VERTICES

A **similar phenomenon** happens when **UV information** is **edited**. A **Point edit** will **result** in **all vertices contained within a point** being **moved**. **Vertices editing** will **not affect** any **other vertices shared within the point** that groups them.



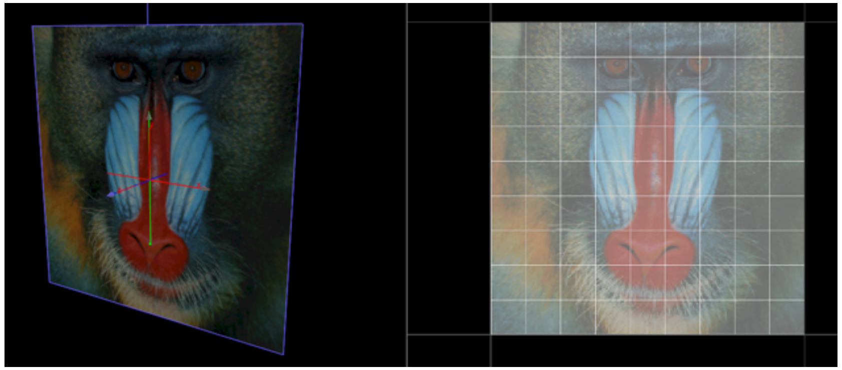
The same vertices edited as a Point



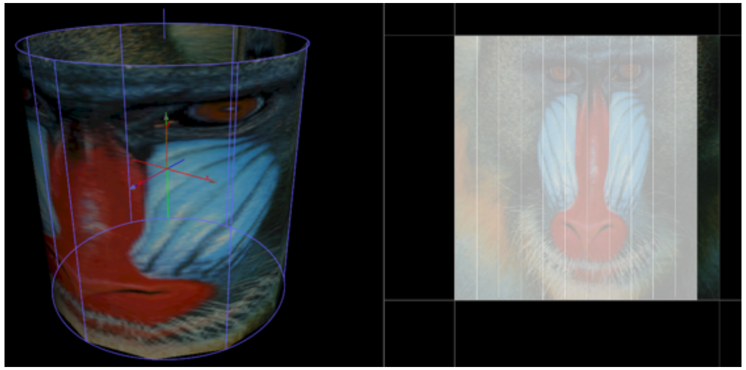
And as a single Vertex

APPLYING MATERIALS TO GEOMETRY – DIFFERENT PROJECTION TYPES

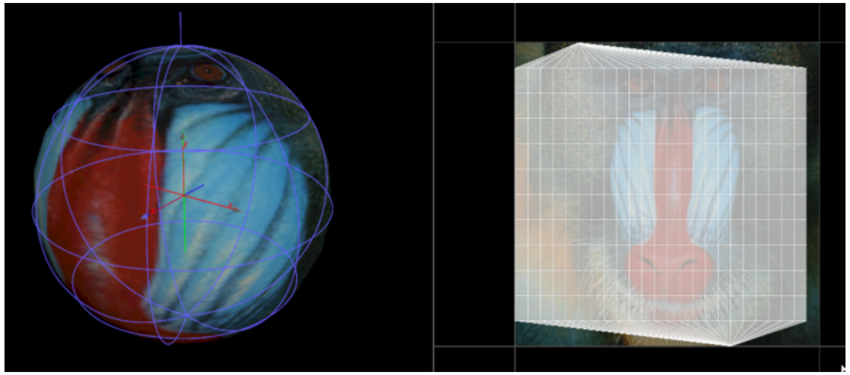
A **3D animation system** will have a **number of tools** for **projecting geometry** into the **UV Square**. **Each tool** creates a **different type** of **projection**, and the **UV Square** can hold **different projection types simultaneously** if required. The **standard projection types** are:



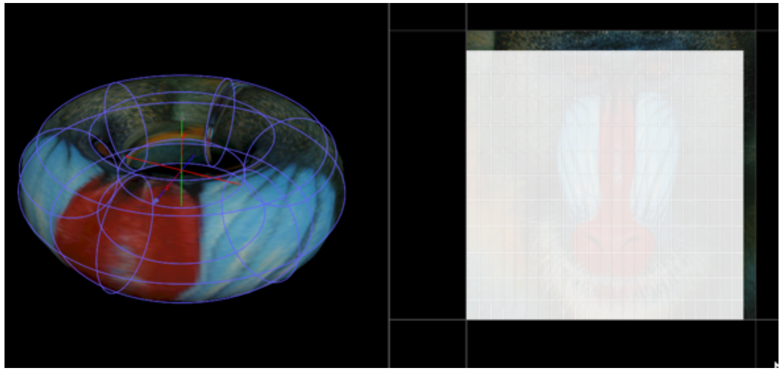
Orthographic



Cylindrical



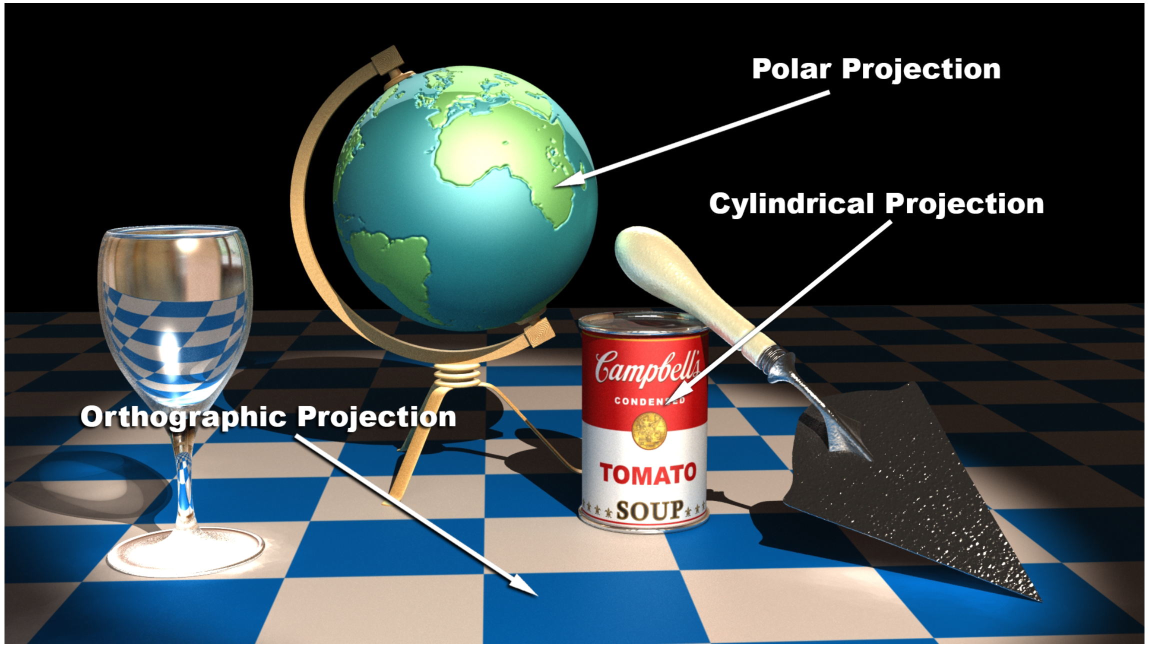
Polar



Toroidal

APPLYING MATERIALS TO GEOMETRY – DIFFERENT PROJECTION TYPES IN CONTEXT

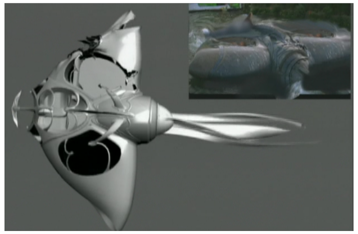
A **textured scene** might consist of a **mix** of **both Painted** and **Procedural Textures**. **In this example** the **Ground** is a checker pattern **Procedural Texture assigned** with an **Orthographic Projection**; the **Globe** is a **Painted Texture** (**including** a **Painted Displacement Texture** for the land mass outlines) **assigned** with a **Polar Projection**; and the **Soup Tin Label** is a **Painted Texture** assigned with a **Cylindrical Projection**.

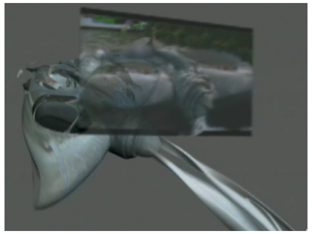


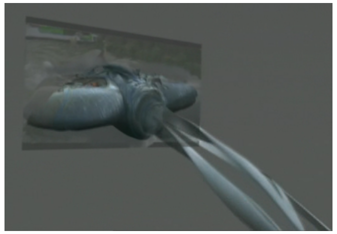
The **other scene objects** are **textured** with **Procedural Materials** that do not require **UV Projections**. The **Wine Glass** has been assigned a **Glass Material**; the **Globe Stand** has been assigned a **Wood Material**; the **Soup Tin Rim** and **Ends** have been assigned a **Chrome Material**; the **Trowel Handle** has been assigned a hand configured **Ivory Material**; and the **Trowel Thread**, **Stem** and **Faceplate** have been assigned a **Stainless Steel Material** (with **Procedural Noise Displacement** also activated).

APPLYING MATERIALS TO GEOMETRY – CAMERA PROJECTIONS

Another method of **projecting textures** onto geometry is known as **Camera Projection.** This is an **Perspective Projection method**, but **done** directly **from the view point of the camera**. This allows for **2D Textures** to be **draped over scene geometry** or **objects** (normally as a **Constant Material**).





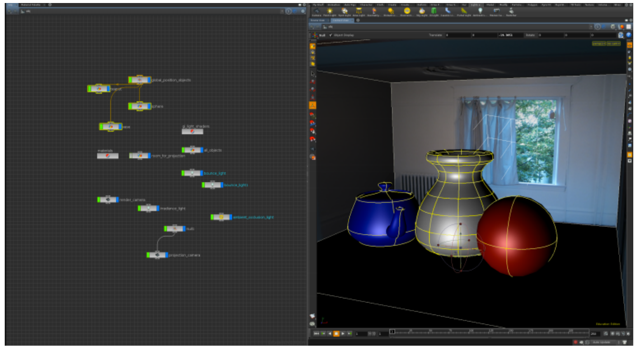


In **Star Wars Episode 1** – **The Deleted Waterfall Scene** reveals the **use** of **Camera Projections** in several ways. The **submarine** has a **location photograph** of a **real world film set camera projected onto** the **digital** version of the **object**. This **gives instant realism** and a very **fast method of texturing** and rendering a complex object.

Camera Projections were also used to build up the **Naboo City environment**. **Rendered images** of the city were **assigned** as **Constant Materials** and then **camera projected onto grids** (with **Opacity activated** to cut out the shapes). These **grids** were then **positioned in 3D space**, **layering up the city**, and **facilitating camera movement with parallax** (multi-planing). The **scene** was **further enhanced** by the **addition** of **camera projected footage of a river**, creating the **rushing water for the waterfall**. **These are the basic principles of Matte Painting.**

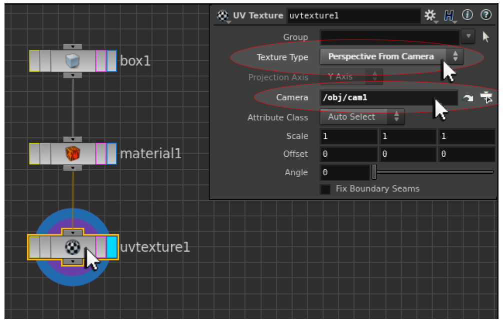
APPLYING MATERIALS TO GEOMETRY – CAMERA PROJECTIONS

Camera Projections are also useful for **rendering tricks**. A **filmed** or **photographed** background plate is then **Camera Projected** onto a **Dummy Geometry Digital Set** as a **Constant Material**. The Dummy Geometry Digital Set is then **activated** as a **Phantom Object**, so that in **only appears** in **Reflections and Refractions** of other surface geometry **at render time**. The **result** is **accurate reflections** of the room in the **surfaces** of digital objects.



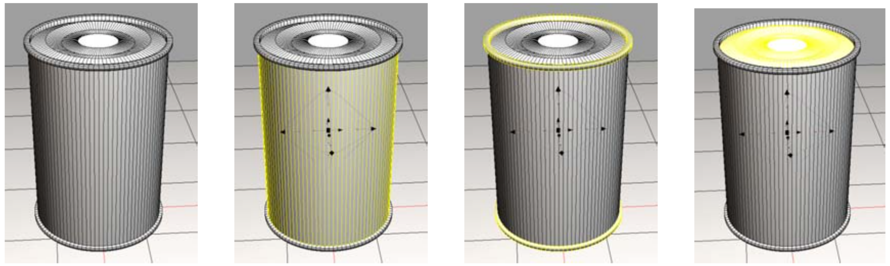
The **Dummy Geometry Digital Set** can also **be activated** as a **Geometry Light**, to create **accurate scene lighting** that **surrounds other digital objects**.

In **Houdini**, **Camera Projections** are **created** using the **UV Texture SOP**.



APPLYING MATERIALS TO GEOMETRY – GROUPING GEOMETRY REGIONS FOR DIFFERENT MATERIALS

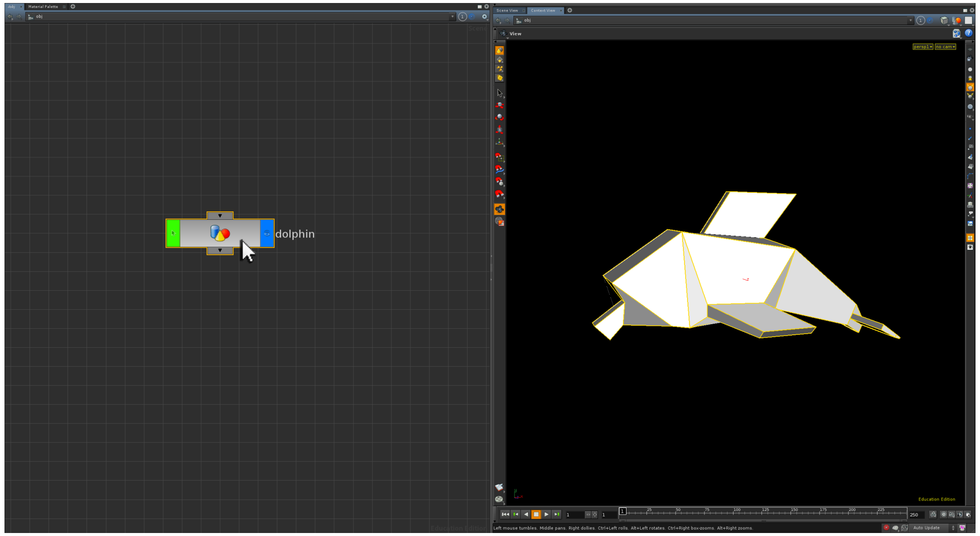
Materials can be **assigned** to **whole geometry** or **different regions** of **connected geometry**. For objects requiring more than one Material, **Primitive Groups can be made** on the geometry **denoting different Material Regions**:

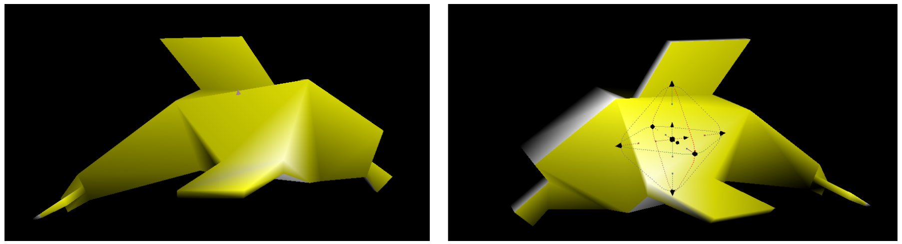


A **Tin Can** for example **may require separate Materials** for the **Label Region**, the **Rim Regions** and **End Regions** of the can. These areas can be **stored as Groups** that are then assigned to specific Materials (both Painted and Procedural).

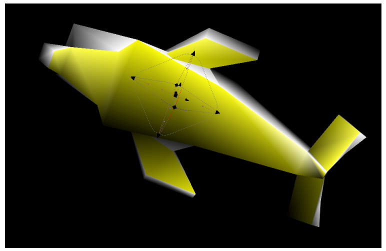
APPLYING MATERIALS TO GEOMETRY – GROUPING GEOMETRY REGIONS FOR MULTIPLE PROJECTIONS

Grouping Geometry also allows for **multiple UV Projections** to take place on a **single mesh**. In this example, a **low resolution** model of a **dolphin** has been **assigned three geometry primitive groups** (sides, under and top).

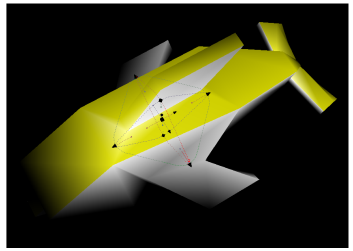




(sides group)



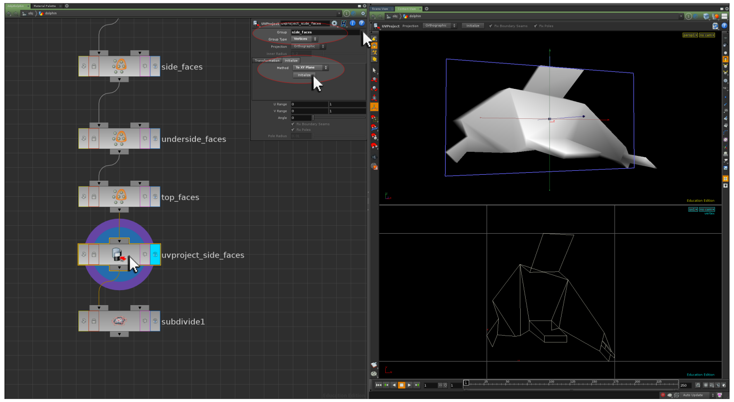
(under group)

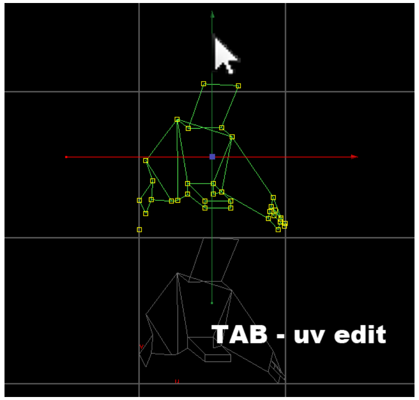


(top group)

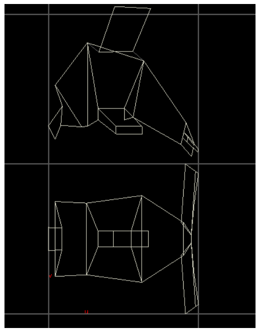
APPLYING MATERIALS TO GEOMETRY – GROUPING GEOMETRY REGIONS FOR MULTIPLE PROJECTIONS

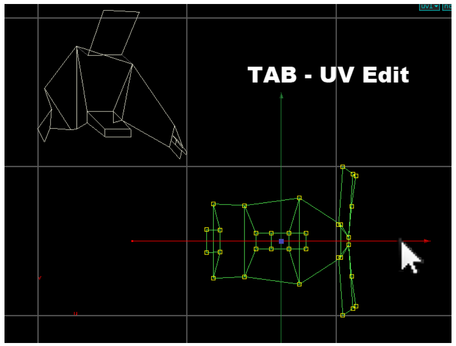
This then allows for 3 **separate orthographic projections** to take place. First, the **sides group** is **projected** into UV Space, and **UV Edited** to the square above the main UV projection square.

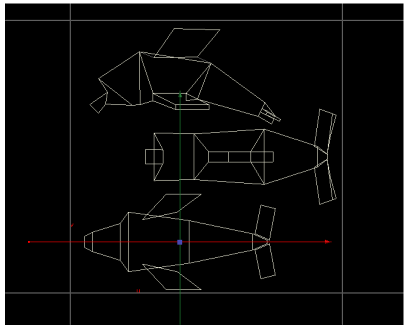




Secondly the **top group** is **projected**, and **UV Edited** to the right side square. The **under group** is then **projected** and **all three projections** are **UV Edited back** into the **main UV square** to create the **final UV layout**.

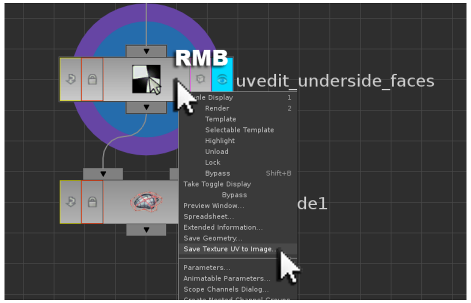


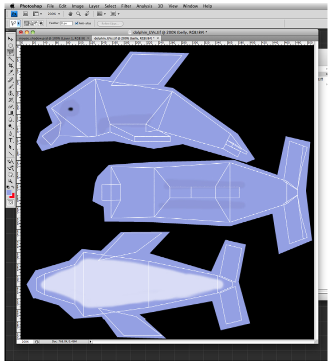


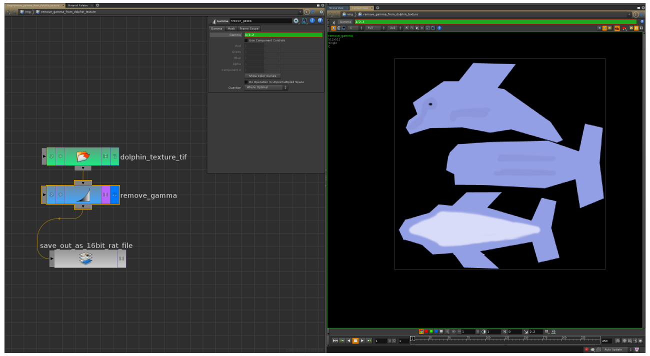


APPLYING MATERIALS TO GEOMETRY – GROUPING GEOMETRY REGIONS FOR MULTIPLE PROJECTIONS

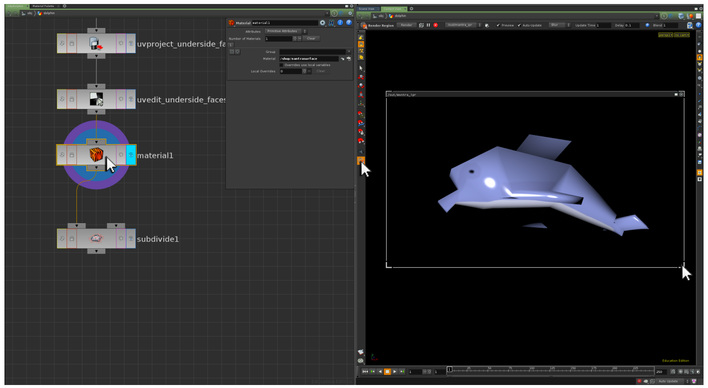
The **final UV layout** can then be **saved to disk** and used as a **layer overlay** in **Photoshop** or **Gimp** to **develop** the **texture map** for the dolphin.

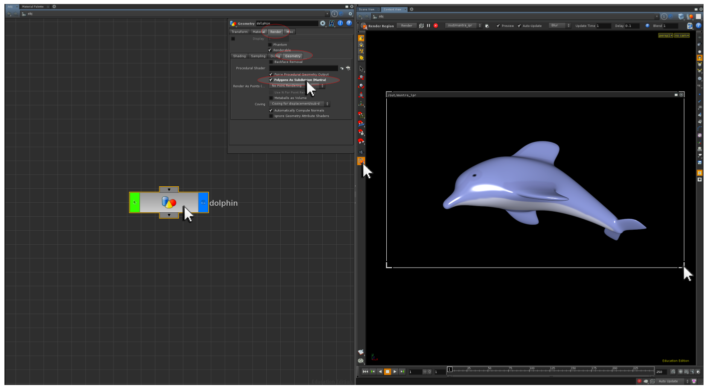






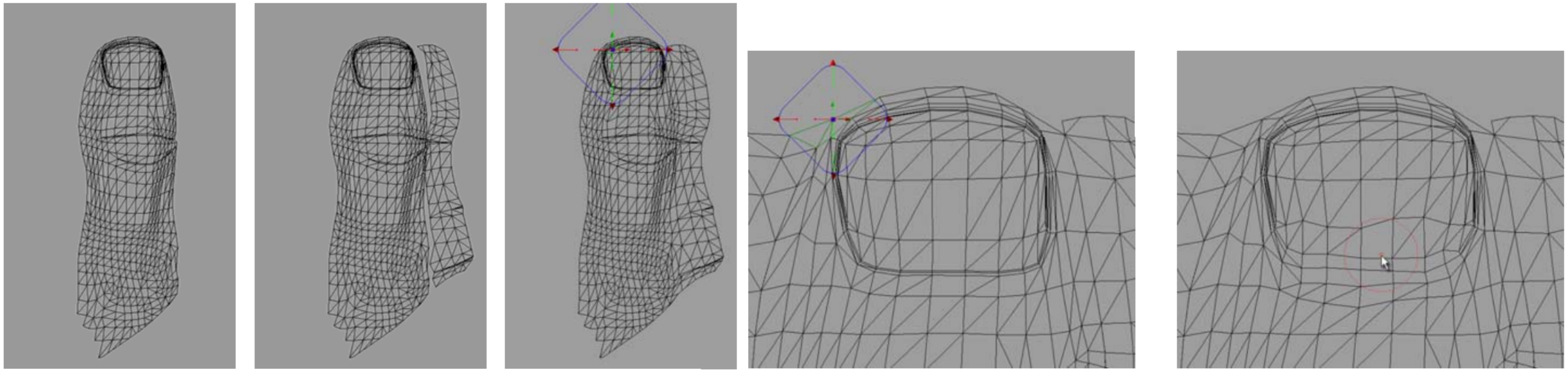
The **final texture** can then be **assigned** to a Material as a **Diffuse Map**, and then the **Material assigned** to the **low res dolphin geometry**. The **final look** of the dolphin can further **be enhanced** by **Subdividing** the **low res mesh** either before or at **render time**.





APPLYING MATERIALS TO GEOMETRY – UV EDITING MULTIPLE PROJECTIONS

The **process** of creating **multiple orthographic projections** can also be **applied** to **more complex geometry**. In this example, the **thumb** of a hand model has **received** an **orthographic projection** for the **front** and the **side** of the thumb. A **UV Edit** is then used to **move the second orthographic projection into position**, and align its vertices with the other edge of the thumb. The **two projections** can then be **UV Fused together** to create a **unified UV Layout**.

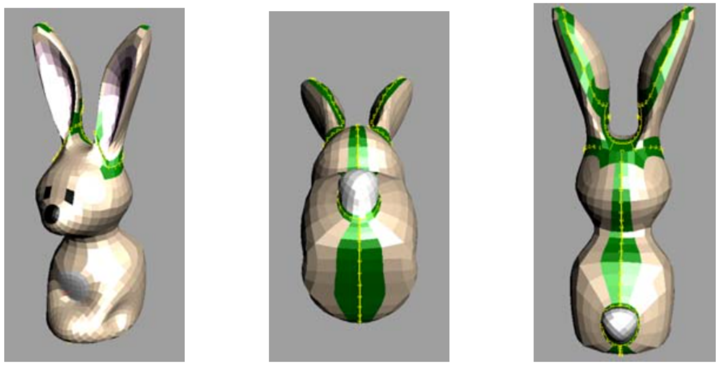


After the initial UV Layout, a **UV Brush Operation** can then **soften** and **smooth** out the proximity of the UV faces around the thumb nail region. **Smoothing UV information** helps guarantee that there are **no overlapping UVs**, which may cause problems in other texturing packages such as Mari.

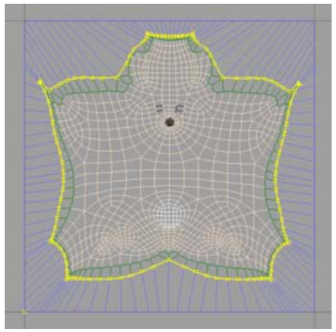
This tools are **designed** to help **create** the **cleanest UV Layout possible**.

APPLYING MATERIALS TO GEOMETRY – UV PELTING

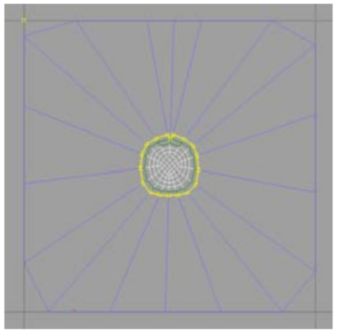
For **organic characters** and **creatures**, **a UV unwrapping process** called **Pelting** can also be activated. This is the **process** of **manually declaring seams** from which a mesh will be unwrapped in UV Space.



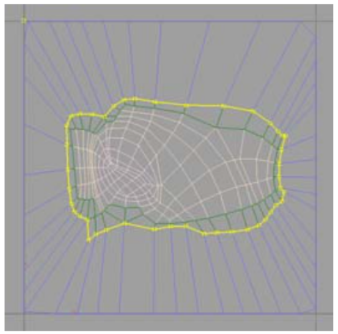
In this example, **seams** have been **declared around the ears**, **spine** and **underside** of the **bunny geometry**. **This** seam declaration **creates four UV Projections** (each ear, the torso, and the tail) that can now **be combined** to create a **final UV Layout of the Pelts**.



Torso Pelt

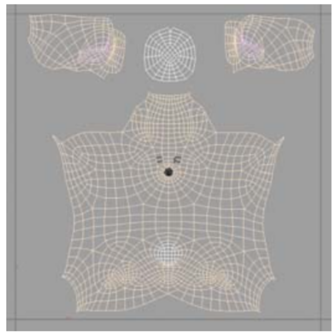


Tail Pelt



Left Ear Pelt

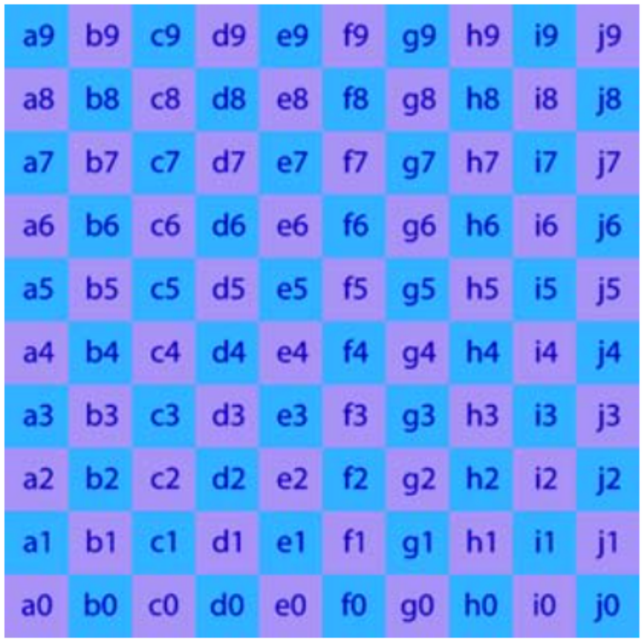
Right Ear Pelt

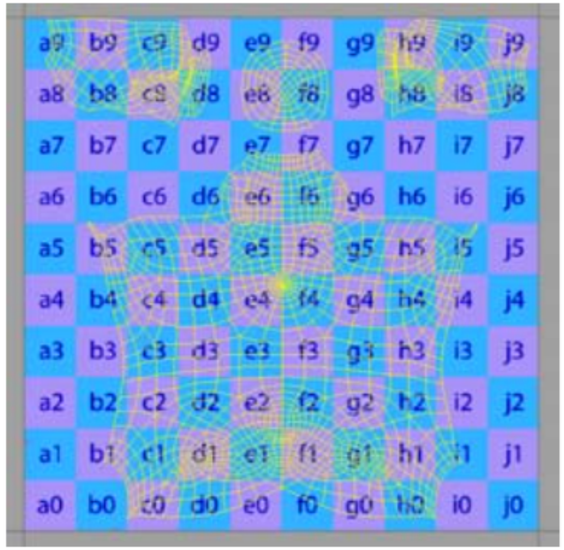


Pelts Combined

APPLYING MATERIALS TO GEOMETRY – UV GRIDS

As UV Layouts develop, a **UV Grid** can be **activated on** the **geometry** as a **Material.** This is a **grid-based image** with a **coordinate assigned to each square**. This is then applied to the **geometry** as a **Material**, and then **used to check the distribution of the UVs**. This technique is especially **useful when softening UVs**, and **ordering / un-distorting the UV projection**.

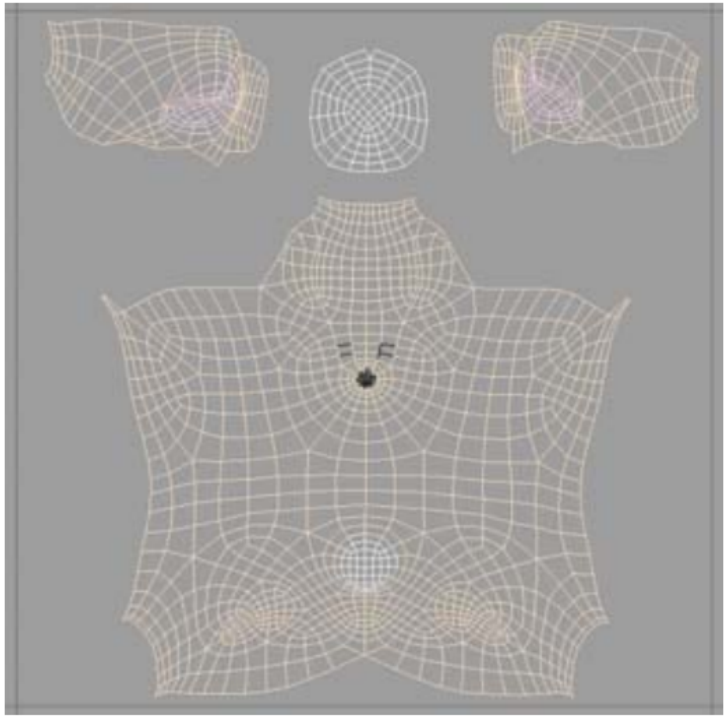




This technique is simply **designed** to help **create** the **cleanest UV Layout possible**.

APPLYING MATERIALS TO GEOMETRY – UV'S AND PAINTING DIRECTLY ONTO GEOMETRY

If **geometry** has been **assigned** a **colour attribute directly** onto its **mesh**, this **colour information** is **automatically passed onto** any **projected UV's**.



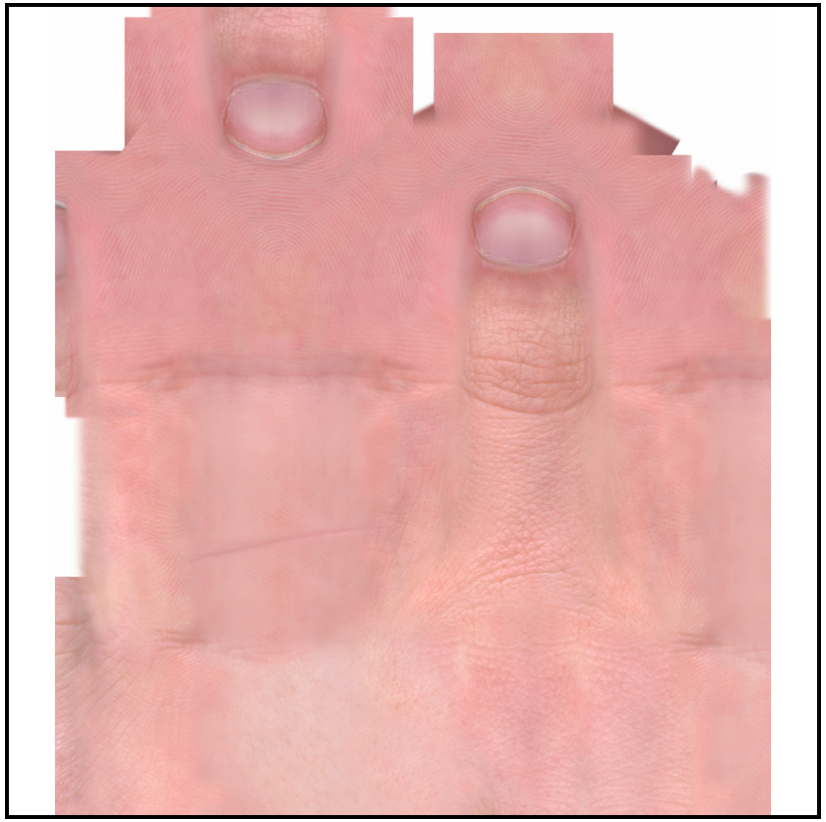
In this example, a **simple bunny mesh** has been **coloured** by assigning **different colours** to **different faces** as a **Primitive Colour Attribute**. When the UV information for this geometry is created, the colour information also viewable. This can prove to be **a useful guide** when **laying out UV information**, as well as **giving a Texture Artist useful information** about the development of a texture map for the mesh.

APPLYING MATERIALS TO GEOMETRY – UV'S AND PAINTING DIRECTLY ONTO GEOMETRY

In this example, **a simple character** head has been **sculpted** from a **sphere**, and then **colour** has been **painted** onto the surface as a **Point Colour Attribute**. As **UV's** were **Polar Projected** onto the Sphere **before** the **shaping and sculpting** of the face began, this gives a **very straightforward UV Layout** to work with. As with the Primitive Colour Attribute, **Point Colour** also **appears** on the **UV projection**.

APPLYING MATERIALS TO GEOMETRY – DEVELOPING TEXTURE SETS

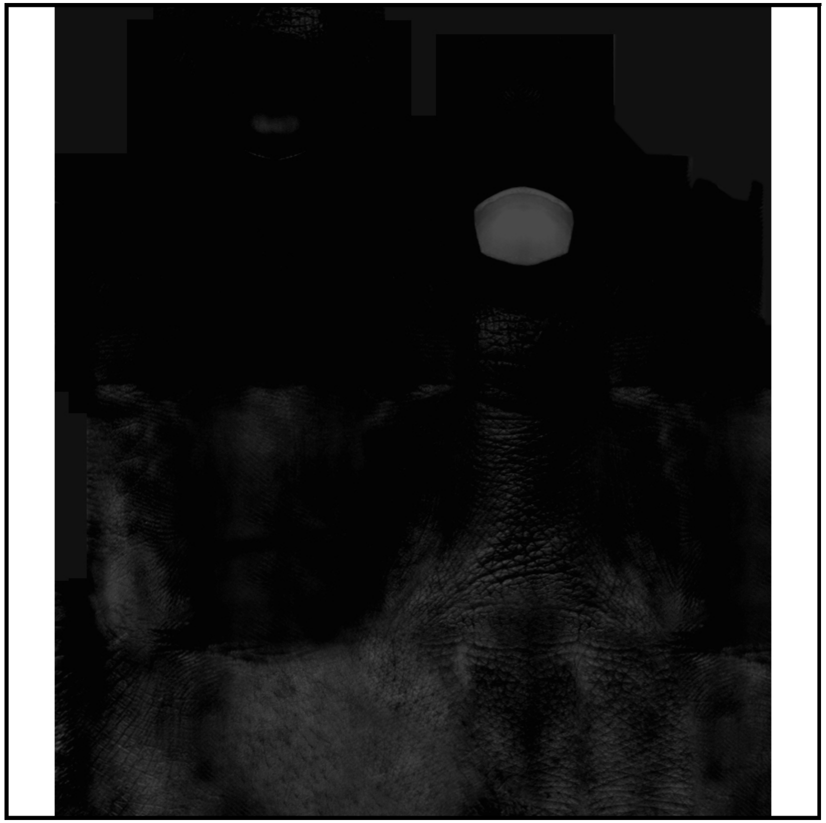
When all the **UV's** have been **successfully created**, a **set of texture maps** can be **developed** that **control different aspects** of the **Material Properties**. In this example, **specific texture maps** for a **thumb** have been created **to control the Diffuse**, **Reflection**, **Emission** (**Ambient**), and **Displacement Components** of a **Material**. The Diffuse Map is the starting point, from which all other texture maps in a set can be developed.



Base thumb + diffuse map

APPLYING MATERIALS TO GEOMETRY – DEVELOPING TEXTURE SETS

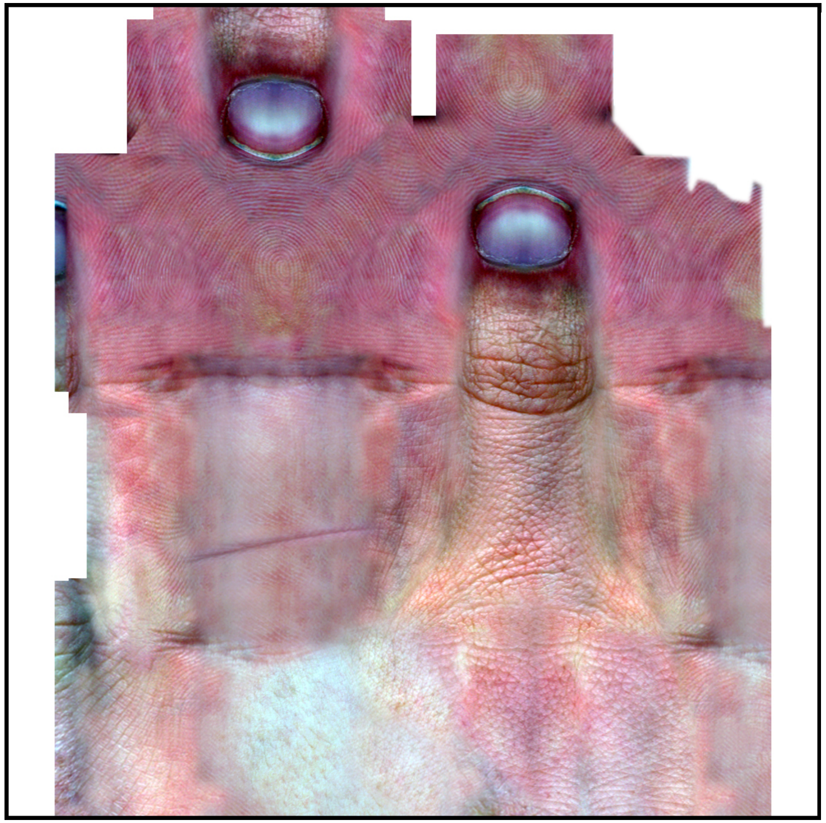
After the application of the Diffuse Map, **a greyscale Reflection map** can be added to **control the reflectivity** of the **thumb surface.** The **thumbnail** is the **whitest region** of the Reflection Map, **allowing it to subtly reflect in the scene lighting**. The majority of the rest of the Reflection Map is darker, but has slightly lighter skin regions that will help give a bit of sheen to the skin surface under scene lighting.



Base thumb + diffuse + specular map

APPLYING MATERIALS TO GEOMETRY – DEVELOPING TEXTURE SETS

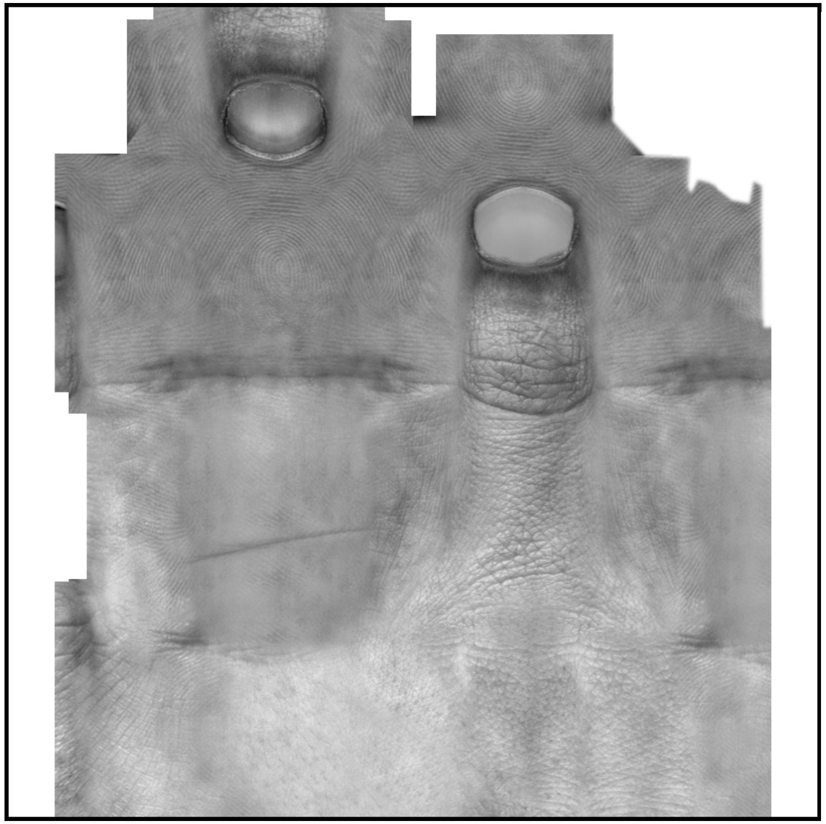
After the application of the Diffuse and Reflection Maps, **a colour Emission (or Ambient) Map can be added.** This has a **subtle effect** of **gently boosting** the **overall glow of the skin** to help give a greater sense of life force.



Base thumb + diffuse + specular + ambient map

APPLYING MATERIALS TO GEOMETRY – DEVELOPING TEXTURE SETS

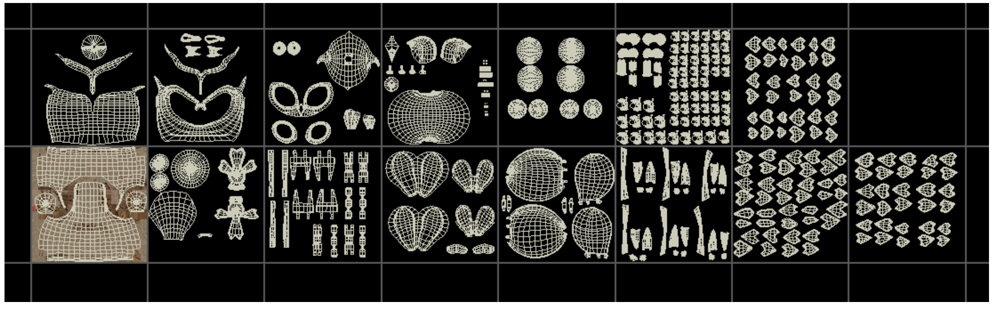
After the application of the Diffuse, Reflection and Emission (Ambient) Maps, **a greyscale Displacement Map can be added.** This helps give a bit of subtle roughness to the thumb geometry so that it is never renders completely smooth.



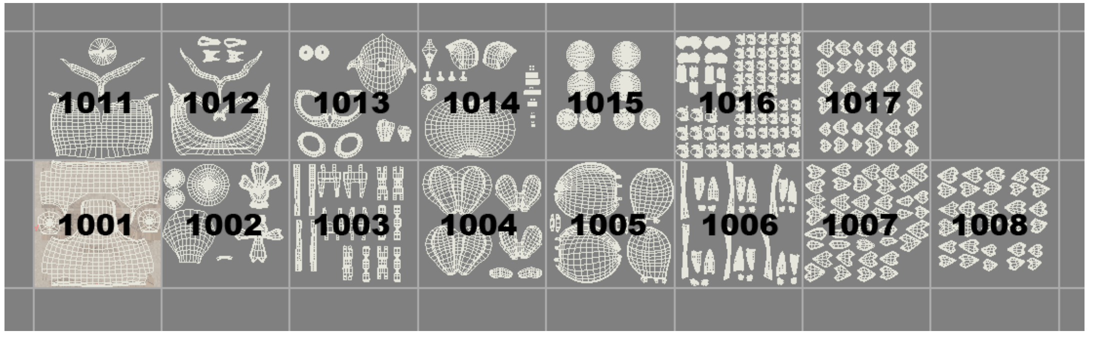
Base thumb + diffuse + specular + ambient
+ displacement map

APPLYING MATERIALS TO GEOMETRY – UV LAYOUT FOR MARI

Mari is a **Texturing Software** that can work with **either** a **single uv square**, or a **grid system of uv's**. Each **UV projection** or **material regions** should ideally be **contained on its own uv grid square**. Doing this allows for **greater resolution** of the resulting **painted texture maps**, than if all the uv's are placed within the standard uv square.



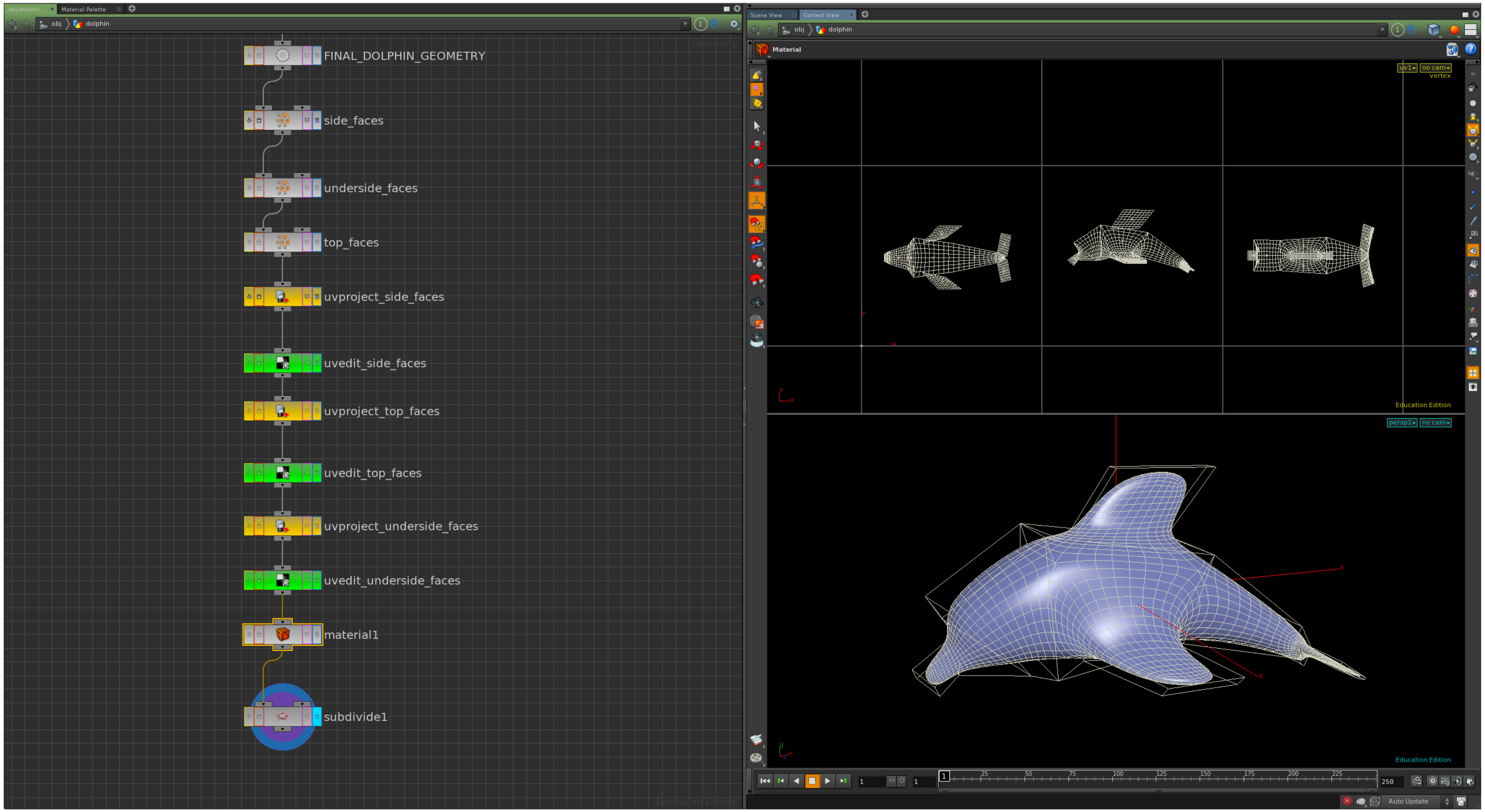
Each **UV Square** is hypothetically given its own **unique UDIM number** so that any **texture maps** generated in Mari can be assigned with the **correct UVs**.



http://bneall.blogspot.co.nz/p/udim-guide.html

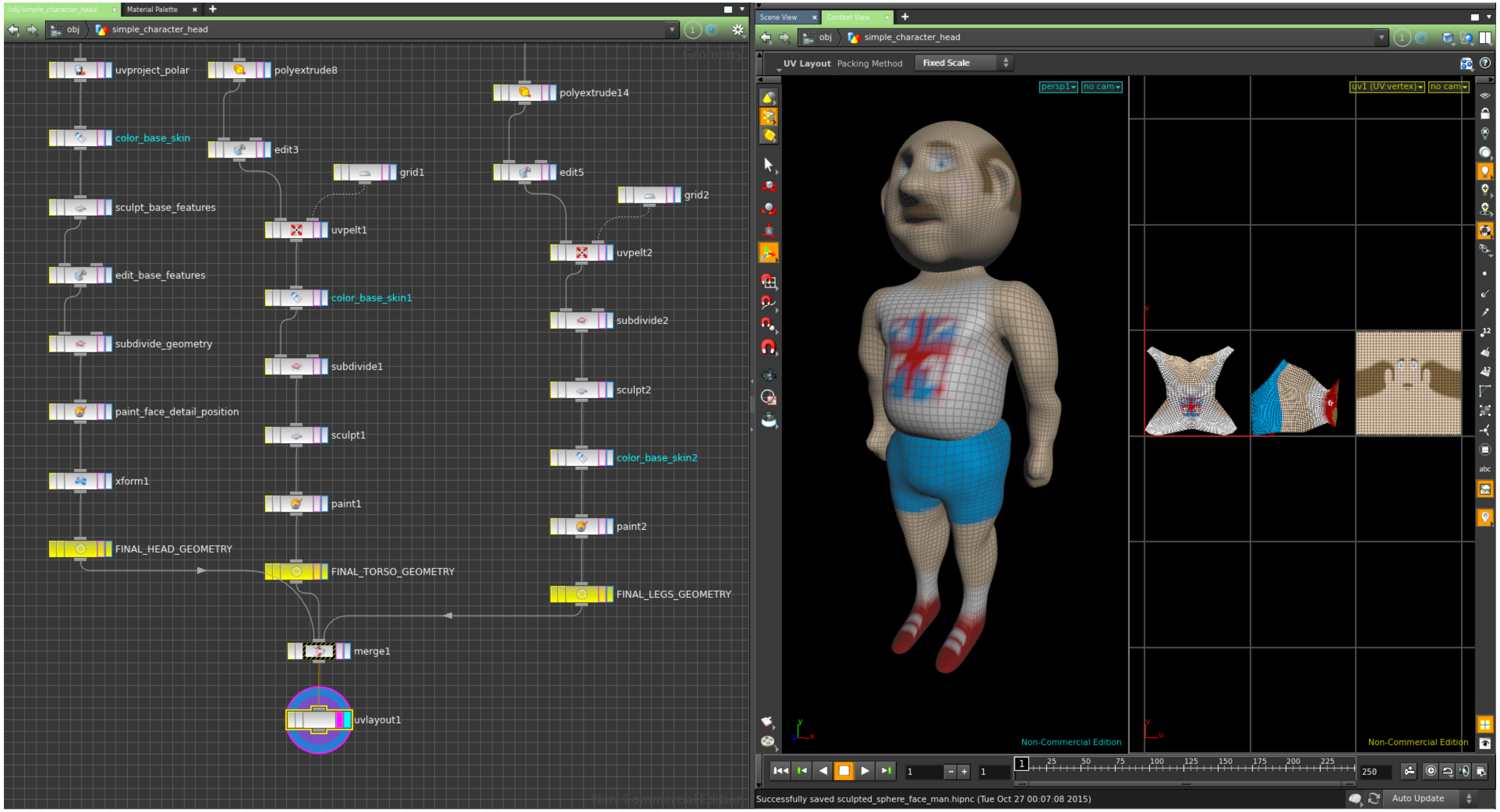
APPLYING MATERIALS TO GEOMETRY – UV LAYOUT FOR MARI

The **Layout of UV's** in this **format** can either be done **manually** using a **UV Edit** or a **UV Transform**.



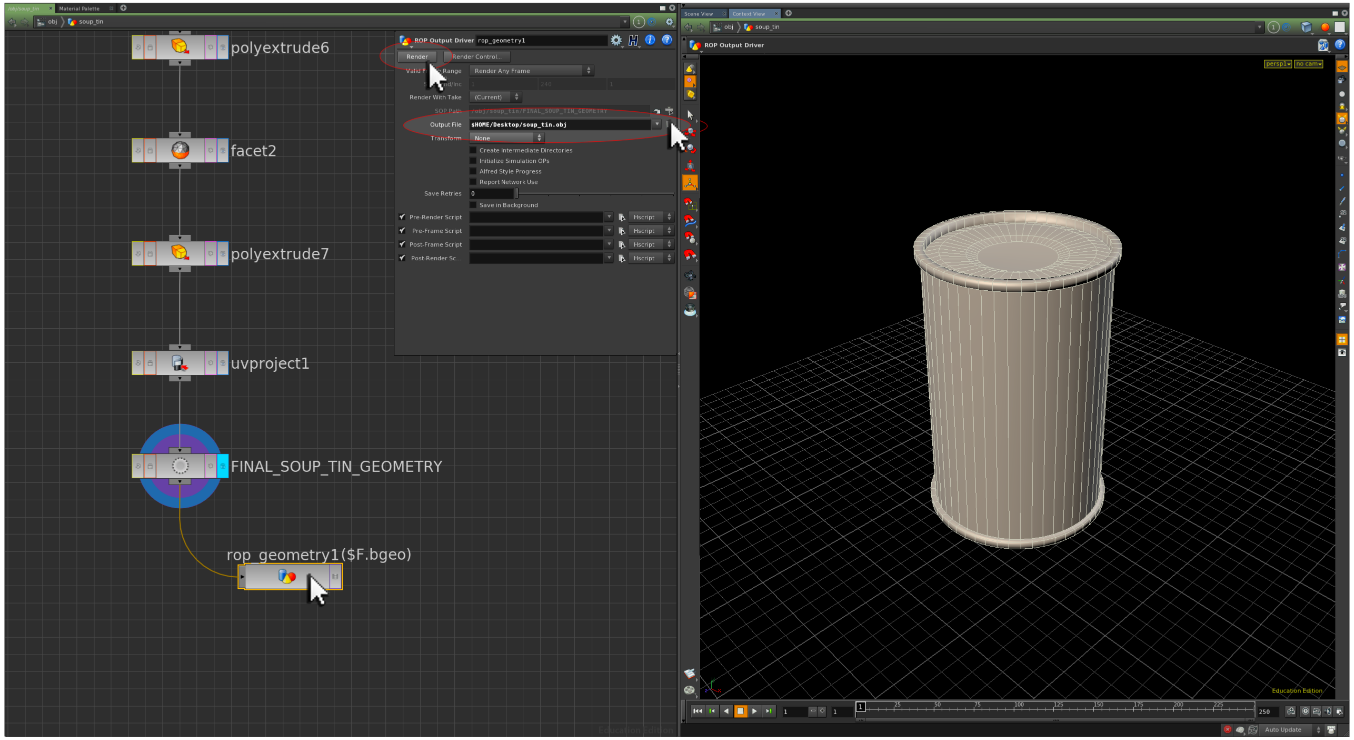
APPLYING MATERIALS TO GEOMETRY – UV LAYOUT FOR MARI

Alternatively in **Houdini** the **UV Layout SOP** can be used to create **automatic UV Layouts in this format** (when each network chain of the model has its own UV Projection assigned to the default UV Square).



APPLYING MATERIALS TO GEOMETRY – GEOMETRY EXPORT FOR MARI

Geometry (with assigned uv's) can then be **exported from Houdini to Mari** by appending a **ROP Output Driver** to the end of the modeling network.

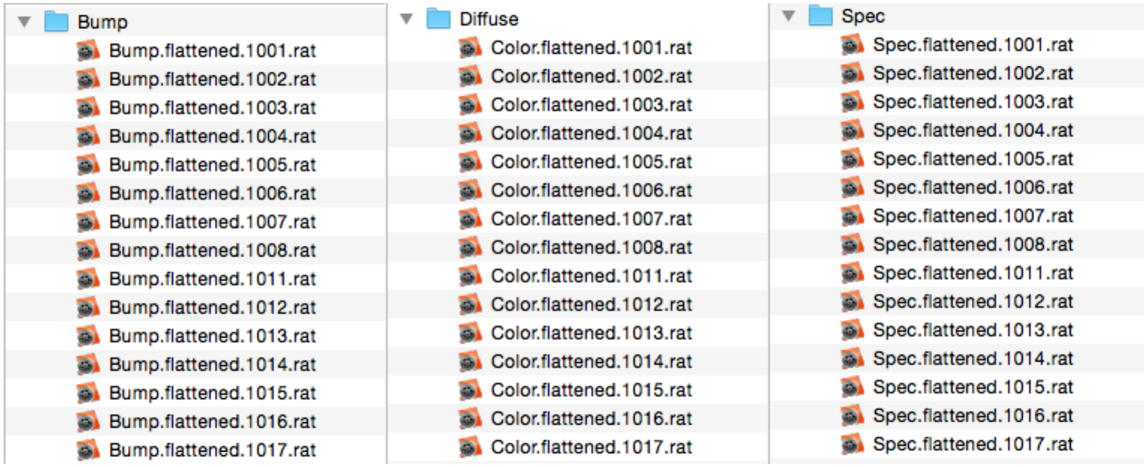


A **save location** and **.obj format** can be specified (for example **$HIP/soup_tin.obj**). When the **Render button** is pressed, the geometry will be written out to disk as a .obj file. When imported **into Mari**, the **.obj file** can be **painted on** to create the final texture maps.

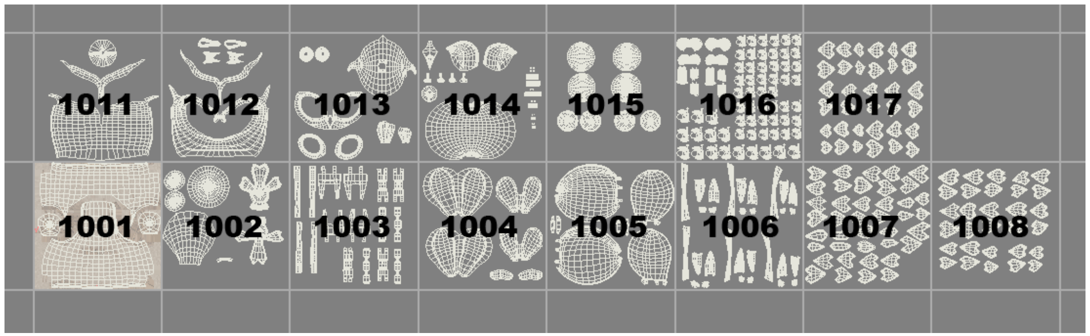
https://www.thefoundry.co.uk/products/mari/

APPLYING MATERIALS TO GEOMETRY – TEXTURE MAPS FROM MARI

When **texturing** in **Mari** has been **completed**, the **resulting texture maps** can be **converted to .rat files** (using **mplay**) for optimal rendering efficiency.



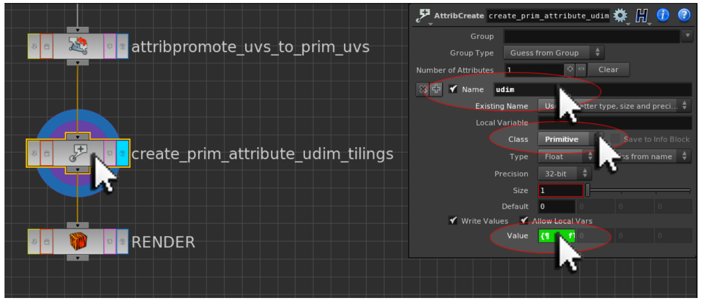
The **Texture Maps** generated by **Mari** will correspond with the **UDIM numbering system**.



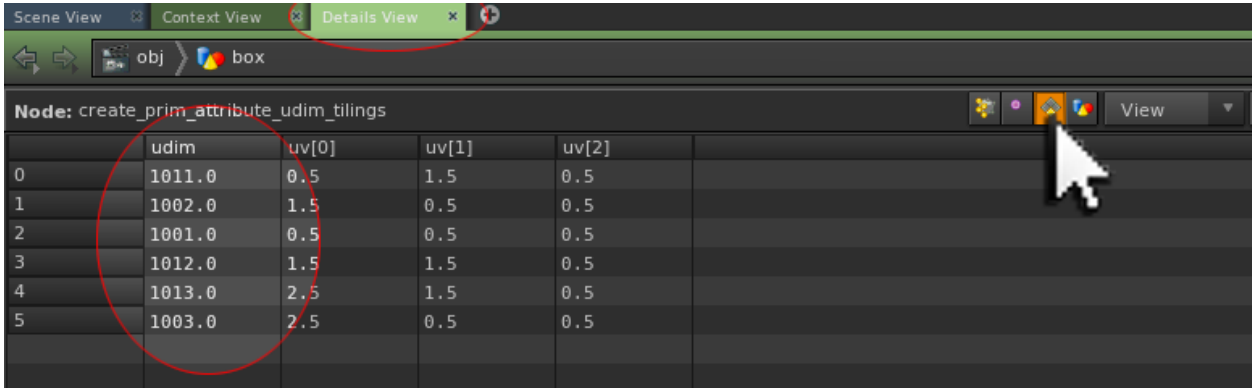
APPLYING MATERIALS TO GEOMETRY – ASSIGNING THE CORRECT UDIM NUMBER TO EACH PRIMITIVE AFTER MARI

The **UV Layout** on the **original geometry** is **duplicated** and **converted** into **a Primitive** (rather than Vertex) **Attribute,** which can then be **assigned** a **custom Primitive Attribute** that creates the correct **UDIM number** using the Expression:

Value **((floor($MAPU) % 10) + 1) + (floor($MAPV) * 10) + 1000**

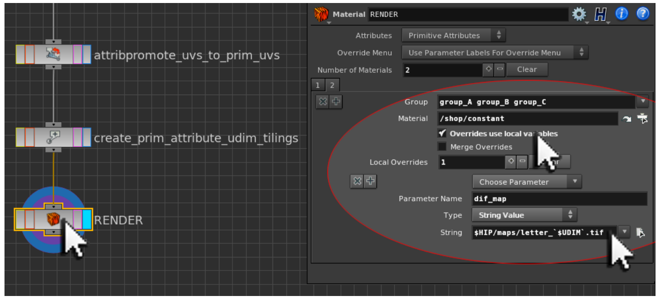


Each Primitive will now be **assigned** the **correct UDIM number**. This can be verified in the Geometry Spreadsheet.

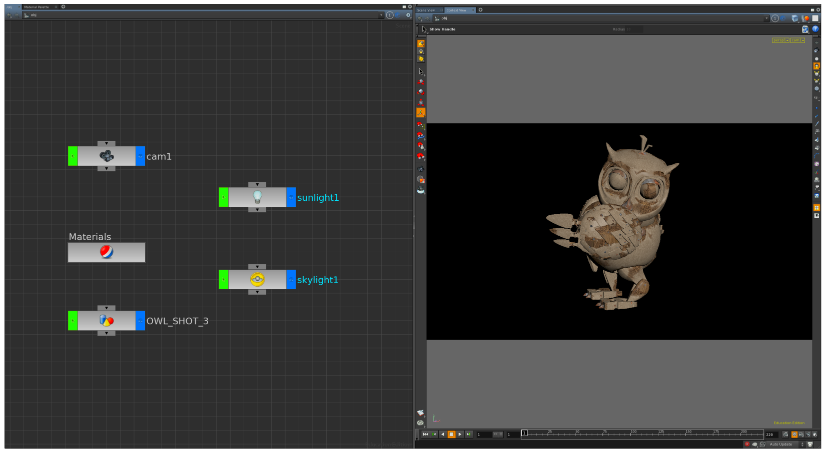


APPLYING MATERIALS TO GEOMETRY – ASSIGNING THE CORRECT UDIM NUMBER TO EACH MATERIAL AFTER MARI

The **UDIM Attribute** can then be procedurally called in the **Material SOP**, so that **at Render Time**, the **correct texture** is **assigned** to the **correct geometry primitive**. This is achieved using a **Material Override** that will bypass the Material's existing parameters. The geometry can then be rendered.



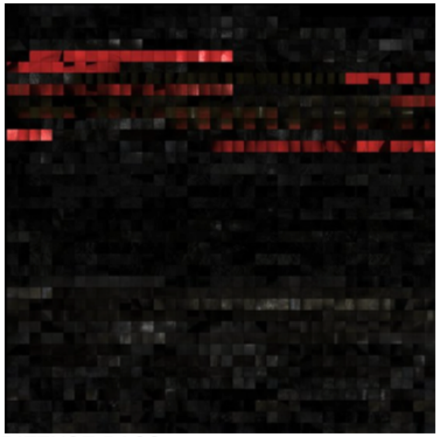
Mari Texturing by Eliana Subero Rivas (NCCA, MADE1415).



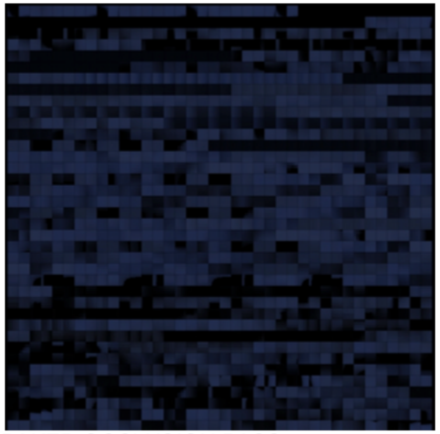
APPLYING MATERIALS TO GEOMETRY – PTEX

PTEX is a **texturing system** developed by **Disney** that allows for the **development of Texture Sets without** the need for **pre-creating UV Layouts**.

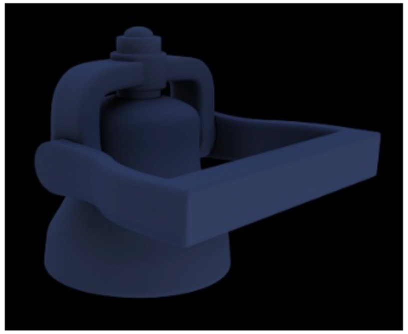
It works by creating an **array of** primitive **faces** in the **UV Square**, each **coloured** by the **paintwork** done any **texture paint software** that supports the **PTEX format** - such as **3D Coat** (http://www.3d-coat.com/) and **Mari** (http://www.thefoundry.co.uk/products/mari/).



DIFFUSE PASS

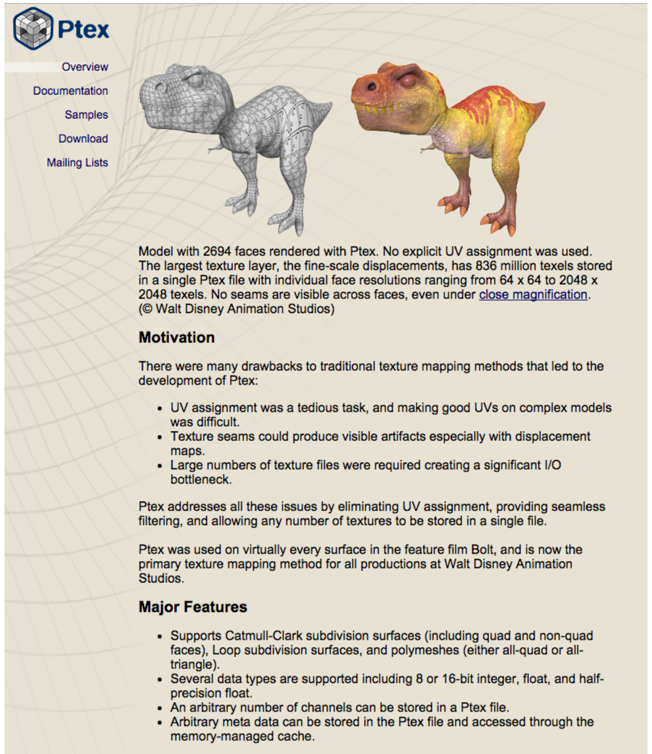


BAKED OCCLUSION PASS



This **array of faces** is in essence the **default behaviour of** a **single polygon**, where **an entire image** is **assigned** to a **single polygon face**; however this principle is used to create a **resolution independent array** of **sub-images** as the **final texture map**.

APPLYING MATERIALS TO GEOMETRY – PTEX


Ptex

Overview
Documentation
Samples
Download
Mailing Lists

Model with 2694 faces rendered with Ptex. No explicit UV assignment was used. The largest texture layer, the fine-scale displacements, has 836 million texels stored in a single Ptex file with individual face resolutions ranging from 64 x 64 to 2048 x 2048 texels. No seams are visible across faces, even under close magnification.
(© Walt Disney Animation Studios)

Motivation

There were many drawbacks to traditional texture mapping methods that led to the development of Ptex:

- UV assignment was a tedious task, and making good UVs on complex models was difficult.
- Texture seams could produce visible artifacts especially with displacement maps.
- Large numbers of texture files were required creating a significant I/O bottleneck.

Ptex addresses all these issues by eliminating UV assignment, providing seamless filtering, and allowing any number of textures to be stored in a single file.

Ptex was used on virtually every surface in the feature film Bolt, and is now the primary texture mapping method for all productions at Walt Disney Animation Studios.

Major Features

- Supports Catmull-Clark subdivision surfaces (including quad and non-quad faces), Loop subdivision surfaces, and polymeshes (either all-quad or all-triangle).
- Several data types are supported including 8 or 16-bit integer, float, and half-precision float.
- An arbitrary number of channels can be stored in a Ptex file.
- Arbitrary meta data can be stored in the Ptex file and accessed through the memory-managed cache.

http://ptex.us/overview.html

http://arstechnica.com/information-technology/2010/07/ptex-3d-texturing-becomes-a-reality-at-siggraph/

http://www.youtube.com/watch?v=GxNIAlOuQQQ

http://www.youtube.com/watch?v=1m05YAAo1j4