Official Autodesk Training Guide

Learning Autodesk® Maya®

2008

The Special Effects Handbook
A hands-on introduction to key tools and techniques in Autodesk Maya 2008 based on the LAIKA short film Moongirl™
# Table of Contents

How to use this book .................................................. 10

## Project 01

<table>
<thead>
<tr>
<th>Lesson</th>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Materials</td>
<td>15</td>
</tr>
<tr>
<td>02</td>
<td>Textures</td>
<td>45</td>
</tr>
</tbody>
</table>

## Project 02

<table>
<thead>
<tr>
<th>Lesson</th>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>03</td>
<td>Lighting</td>
<td>77</td>
</tr>
<tr>
<td>04</td>
<td>Shadows</td>
<td>97</td>
</tr>
<tr>
<td>05</td>
<td>Cameras</td>
<td>131</td>
</tr>
<tr>
<td>06</td>
<td>Raytracing</td>
<td>151</td>
</tr>
</tbody>
</table>

## Project 03

<table>
<thead>
<tr>
<th>Lesson</th>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>07</td>
<td>Controlling Renders</td>
<td>163</td>
</tr>
<tr>
<td>08</td>
<td>Special Effects and Compositing</td>
<td>187</td>
</tr>
<tr>
<td>09</td>
<td>Hardware Rendering</td>
<td>209</td>
</tr>
<tr>
<td>10</td>
<td>Vector Rendering</td>
<td>219</td>
</tr>
<tr>
<td>11</td>
<td>Maya Paint Effects</td>
<td>237</td>
</tr>
</tbody>
</table>

## Project 04

<table>
<thead>
<tr>
<th>Lesson</th>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>Caustics and Global Illumination</td>
<td>247</td>
</tr>
<tr>
<td>13</td>
<td>Final Gathering and HDRI</td>
<td>259</td>
</tr>
<tr>
<td>14</td>
<td>mental ray Shaders</td>
<td>281</td>
</tr>
</tbody>
</table>
### Project 05

<table>
<thead>
<tr>
<th>Lesson</th>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>Maya Dynamics</td>
<td>317</td>
</tr>
<tr>
<td>16</td>
<td>Rigid Body Dynamics</td>
<td>323</td>
</tr>
<tr>
<td>17</td>
<td>Rigid Body Constraints</td>
<td>337</td>
</tr>
<tr>
<td>18</td>
<td>Rigid Body Optimization</td>
<td>351</td>
</tr>
</tbody>
</table>

### Project 06

<table>
<thead>
<tr>
<th>Lesson</th>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>Introduction to Particles</td>
<td>371</td>
</tr>
<tr>
<td>20</td>
<td>Rigid Bodies and Particles</td>
<td>409</td>
</tr>
<tr>
<td>21</td>
<td>Particle Collisions</td>
<td>415</td>
</tr>
<tr>
<td>22</td>
<td>Particle Expressions</td>
<td>431</td>
</tr>
<tr>
<td>23</td>
<td>The emit Function</td>
<td>449</td>
</tr>
<tr>
<td>24</td>
<td>Advanced Particle Expressions</td>
<td>467</td>
</tr>
<tr>
<td>25</td>
<td>Goals</td>
<td>481</td>
</tr>
<tr>
<td>26</td>
<td>Particle Instancing</td>
<td>495</td>
</tr>
</tbody>
</table>

### Project 07

<table>
<thead>
<tr>
<th>Lesson</th>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>Rendering Particles</td>
<td>519</td>
</tr>
<tr>
<td>28</td>
<td>Compositing</td>
<td>545</td>
</tr>
</tbody>
</table>

### Project 08

<table>
<thead>
<tr>
<th>Lesson</th>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>29</td>
<td>Maya Fluids</td>
<td>557</td>
</tr>
<tr>
<td>30</td>
<td>Maya Fur</td>
<td>579</td>
</tr>
<tr>
<td>31</td>
<td>Maya Hair</td>
<td>599</td>
</tr>
<tr>
<td>32</td>
<td>Maya nCloth</td>
<td>625</td>
</tr>
</tbody>
</table>

Index 642
Lesson 08
Special Effects and Compositing

Once you have modeled, textured, and added lights to your scene, there are a number of special effects you can include to enhance the quality of your render before and after rendering.

In this lesson you will learn the following:

• How to control glow;
• How to use motion blur;
• How to use mental ray motion blur;
• How to use depth of field;
• How to render for compositing.
Special effects

OpticalFX lets you add glows, halos, and lens flares to lights. Those effects can be used to simply brighten up a light source or to create explosions, rocket thrusters, and other special effects. Shader glow can be used to brighten up a material with a luminous radiance. It can be used to create lava, neon lights, and other glow effects.

Light glow

In the real world, when light shines directly into an observer’s eye or into a camera’s lens, the light source may appear to glow. If the light passes through a mesh (for example, a star filter on a camera) or through hair or eyelashes, the light will refract, producing a star-like glow. In some cases, the light may reflect off the surface of a camera’s compound lens and produce a lens flare. These are all examples of optical light effects.

When lights appear to glow, it is purely a retinal effect in the eye. To see this, look up at a light source such as a street light and squint. You will see a glow around the light. Now use your finger to block only the light source; the glow disappears. Notice also that if you cover only part of the light source, the glow is still visible and will in fact appear in front of your finger. The light glow in Maya simulates this real world effect of blocking the light source, called occlusion.

Light glow occlusion

The most common issue that arises when working with light glow in Maya is the need to control Light Source Occlusion. Often people will animate the position of objects that pass in front of glowing lights and will find that the glow shows right through the objects. This is because the light needs you to specify how big or small the light source actually is in order to know when it is completely covered by an object.

Note: The light glow feature is only supported by the Maya software renderer.

1 Create a light glow

- Open a point light’s Attribute Editor.
- In the Light Effects section, click on the map button beside the Light Glow.

An opticalFX node is automatically created, connected to the light node, and displayed in the Attribute Editor. Also, a new icon has appeared, surrounding the light source in the views.
2 Set the size of the light source
Now that you have created a glow effect, you need to consider how you want this glow to behave. Recall that the light glow is only going to shut off completely if the entire light source is occluded. If the light is going to pass behind an object, the size of this *sphereShape* icon, relative to the size of the object, will determine whether you see the glow though the object or not.

- Select the light glow icon in one of the views.

  *Notice that a new tab appears in the Attribute Editor, called *sphereShape*.*

- Click on the *sphereShape* tab.
- Select **Render Sphere Attributes → Radius**.
- Use this **Radius** attribute to adjust the size of the icon in the scene.

**Note:** Adjusting the **Radius** attribute will not affect the appearance of the light glow. It is only used to determine occlusion.

In the images shown above, a glowing point light moves from left to right behind the columns. In the middle images, the glow is partially dimmed with a radius of **1.0**, increasingly dimmed with a radius of **0.3**, and fully occluded with a radius of **0.1**.
Shader glow

Unlike light glow, shader glow in a scene is controlled by a single shaderGlow node. This node can be found in the Hypershade window, under the Materials tab.

1  Scene file
   • Open the scene file called 08-glow_01.ma.
     
This scene contains a big star fish with pre-created shading network and a textured background.

2  Add a glow effect to the fish
   • Select the fish geometry.
   • In the Hypershade, click the Graph materials on selected objects button.
     
The fish already has a shading network that mimics stars on its skin.
   • Select the fish:lambert2 material.
   • In the Attribute Editor, scroll to the Special Effects section and set Glow Intensitity to 1.0.
   • Render the scene.

You should see a glow where the material is not black. The glow is bluish because it inherits of the color of the stars. Also, the stars are connected to the incondescence of the material, so they will be visible even if there are no lights in the scene.

The star material with glow
3 Hide the source object

- Open the Special Effects section and set Hide Source to On.

  The Hide Source attribute will render the glow without the geometry, giving an interesting effect.

- Render the scene.

4 Shader glow

By modifying a scene's shader glow, you will affect how the glow renders for your entire scene. You can achieve some very interesting effects by tweaking this specialized shader.

- In the Hypershade, select the Materials tab.
- Open the Attribute Editor for the shaderGlow1 node.
- Try to change the Glow Type and Halo Type to different values, and then render your scene.
Creating a neon effect

Neon tubes are the quintessential shader glow example. Try this to create a realistic neon effect.

1 Create the shader
   - Create a **Surface Shader** material and **assign** it to any object.
   - Set the **Out Color** attribute to a bright color.
   - Set the **Out Glow Color** to a darker complementary color.

   *Notice how you are able to set the glow color directly.*

2 Render the effect
With other material types, there is no attribute to control the shader glow color directly. It is derived from the glow color on the shaderGlow node and the color of the material. With the surface shader material, it is possible to experiment with different combinations of glow color and surface color. Also, because the surface shader has no sense of a shading model, it renders as though it is self illuminating—perfect for neon tubes, L.E.D. displays, etc.

Motion blur

Motion blur simulates how a real camera works if objects are moving while the camera’s shutter is still open. This technique is very common in the entertainment industry for creating photorealistic images and animation involving quick motions.

With the Maya software renderer, there are two types of motion blur: 2D and 3D. The shutter angle determines the blur length, but this can be overridden in the Render Settings. This matter will be discussed later in the lesson.

Understand the shutter angle

Whether using 2D or 3D motion blur, it is important to understand the shutter angle. The motion blur algorithm uses a shutter open, shutter mid and shutter close sample for every frame to determine the change in position of a given triangle.

Note: Triangle refers to a tessellation triangle on a surface.

The shutter angle that you specify for motion blur will determine the resulting amount of blur to be calculated. Following is how the motion blur is calculated, taking into account the shutter angle:

Take the Shutter Angle value (the default is 144), and divide it by 360-degrees. For example, \( \frac{144}{360} = 0.4 \).

0.4 represents the interval in time between the shutter open and shutter close samples. Shutter mid is always the frame time itself. For example, for motion blur at frame 1, shutter open would be at frame 0.8 and shutter closed would be at frame 1.2. However, when we calculate motion blur for mental ray, we calculate forward only.

By this, you can see that a shutter angle of 360-degrees would give shutter open and close samples that are exactly one frame apart, i.e. \( \frac{360}{360} = 1 \).

You will notice that by setting the shutter angle to 360, we increased the amount of motion blur. This is because the longer the shutter is open (i.e. the further apart the shutter open and shutter close samples are taken), the blurrier a moving object will appear to be.
How to change the shutter angle
- Open the Attribute Editor for the camera.
- Open the Special Effects section.
- Adjust the Shutter Angle attribute.

mental ray motion blur

In mental ray, there are two types of motion blur: Linear (transformation) and Exact (deformation). Motion blur in mental ray blurs everything: shaders, textures, lights, shadows, reflections, refractions, and caustics. The shutter angle determines the blur path length, but this can be adjusted by the mental ray motion blur attributes in the Render Settings.

To turn On motion blur in mental ray, go to the mental ray Render Settings and open up the Motion Blur section. From there, you can select one of the options from the calculation drop-down menu.

Note: Motion blur in mental ray is calculated forward only.

Linear vs. exact motion blur

As it is with the software renderer’s motion blur, the decision whether to use linear (2D) or exact (3D) mental ray motion blur depends on the type of motion of your object, as well as the time available to render the animation. Linear motion blur is faster to calculate than exact motion blur.

Linear motion blur only takes into account an object’s transformation, rotation, and scale. The object’s deformation will not be considered. For example, if you have blend shapes or a skeleton that deforms a piece of geometry, the resulting motion wouldn’t be considered when calculating this type of motion blur.

Exact takes into account all the transformations as well as the object’s deformations. This type of blur is more expensive to render.

Note: An object’s motion blur can be turned off in its Render Stats section of the Attribute Editor.
Editing mental ray motion blur

**Motion Blur By** is a multiplier for the **Shutter Angle**. The larger this value, the longer the shutter remains open, resulting in more blur.

**Shutter** represents the length of time the camera’s shutter is open. The longer a shutter is open, the more blurry an object will be. However, unlike a real camera, the shutter value does not affect the brightness of an image. If the shutter is set to 0, there will be no motion blur. Larger values increase the length of the blur.

**Shutter Delay** represents the normalized time that a shutter remains closed before opening. For instance, if the shutter delay is set to 0, the shutter opens at the beginning of the frame. If the shutter delay is set to .5, it opens halfway through the frame.

There are four separate controls for **Time Contrast: Red, Green, Blue** and **Alpha**. If you have a fast-moving object, these values can usually be set high. Motion blur tends to make sampling artifacts less noticeable, so you can get away with higher contrast values (in other words, lower quality settings). However, if you find that your motion blur is grainy, you can smooth it by decreasing your time contrast values. The lower the time contrast values, the greater your render times.

**Tip:** *Always try fixing the quality of motion blur by decreasing Time Contrast values first and Number of Samples last. This way, you can increase render performance while not compromising non-blurred anti-aliasing.*

**Motion Steps** can create motion paths from motion transforms. The image on the left represents a value of 1 for motion steps. The image on the right represents a value of 8 for motion steps. Notice the rounder blur on the outer edges of the blade on the right.

![Different amount of motion steps](image-url)
2D vs. 3D motion blur

The decision whether to use 3D or 2D motion blur is really a matter of determining which one is more appropriate for a given scene and the time available to render the animation. 3D motion blur is usually slower and more memory intensive. However, there will be times when 3D motion blur is required because of some limitations of 2D blur (discussed in the next section). In general, it is recommended that you try to use 2D motion blur because it is very fast and produces excellent results in most cases. All of the motion blur attributes, other than Shutter Angle, are found in the Render Settings under the Motion Blur section. If it is desirable for motion blur to be off for some objects, open the Attribute Editor for those objects and toggle Off the motion blur in the Render Stats section.

The following example compares the results of 2D vs. 3D motion blur.

There was quite a difference in rendering time for the above images. The 3D motion blur image took about four times as long as the 2D motion blur image to render.

Note: Motion vector files can be used by other programs to generate blur.

Limitations of 2D motion blur

2D motion blur does not work well in these situations:

Moving transparent objects with a background
The background will also be blurred in this case. The solution is to blur the transparent object separately and composite it into the rest of the scene.
Detailed background behind moving objects
Some details might be lost since the renderer has to make assumptions about the background area occluded by the moving objects. The solution is to blur the moving objects without the background and then composite the results.

Fast rotating objects
A motion vector can be thought of as the direction of a pixel in 3D. This vector does not contain any rotation values, so the rendered image will show a linear movement because it does not know about the arc motion of the pixel in between the first and last positions.

Objects entering from outside the image or leaving the image
The renderer does not know the object color outside of the image and has to make assumptions. The solution is to render a slightly larger image, which covers the original image, and then crop it to the desired size.

Volume objects (particles, fog) and image planes
Motion vectors are only calculated for moving triangles (tessellated NURBS and poly meshes).

Note: The rendered results from 3D and 2D are quite different. It is not a good idea to mix the rendered images from these two kinds of blurring operations.

Depth of field
Depth of field is a photographic effect in which objects within a certain range of distance remain sharply focused. Objects outside this range appear out of focus. You can simulate this using the camera's Depth of Field attribute. This is not a post-process effect in mental ray, but true depth of field.

1 Setting up the camera for depth of field
   - Open the file 08-depthOfField_01.ma.
   - In a four-view layout, set camera1 to replace the Perspective view.
   - In the top view, select camera1 and press t to show the camera's Manipulators. Place the Center of Interest at the location you want to remain in focus.
   - Select Window → General Editors → Connection Editor.
   - Open the Hypershade and select the camera you are using from the Cameras tab.
   - Reload this camera into both sides of the Connection Editor.
   - Connect the Center of Interest to the Focus Distance.
Note: Another alternate but equally useful workflow in setting up depth of field is to use the Distance Tool. This can be found under Create → Measure Tools → Distance Tool. This will allow you to measure the distance between the camera and the point in your scene that you want to use as the focus distance.

2 Enable depth of field
   - In the Attribute Editor for the camera, open the Depth of Field section.
   - Set the Depth of Field flag to On.
   - Adjust the F Stop to control the amount of depth of field.

   The F Stop value represents the distance in front of and behind the focus distance that will remain in focus. A low value represents a short distance that will be in focus; a very high value F-stop will result in very little blur because a deeper range is in focus. In essence, the lower the F-stop value, the smaller the region in focus will be.

   Image rendered with depth of field

Tip: It is possible to use Render Region to test render depth of field.
Limitations of depth of field

Transparent surfaces can cause problems with depth of field. The technical reason for this limitation is that the transparent surface is at a certain depth from the camera. The renderer only stores one depth per pixel, and it chooses to store the nearest point to the eye. For transparent surfaces, the depth of the transparent surface will determine the blur, so the background will show through, un-blurred. The background, when seen through the transparent object, will be blurred at the same depth as the transparent surface. This limitation is not limited to Maya and has led to the industry accepted practice of rendering components separately and compositing them.

Reasons to render for compositing

Compositing is the process of merging multiple layers of image information into one image to create a final look. A common misconception is that compositing is for large productions with many artists. However, smaller production facilities and individual artists can also benefit from the opportunities and advantages offered by compositing. For example, with compositing you can:

- Have the flexibility to re-render or color-correct individual elements without having to re-render the whole scene.
- Increase creative potential and achieve effects with the 2D compositing package that are not possible with the renderer.
- Take advantage of effects that are faster and more flexible in 2D, such as depth of field and glow, rather than rendering them in 3D.
- Combine different looks from different renderers, such as hardware and software particle effects.
- Combine 3D rendered elements with 2D live action footage.
- Save time when rendering scenes where the camera does not move; you only need to render one frame of the background to be used behind the whole animation sequence.
- Successfully render large complex scenes in layers so that you don’t exceed your hardware and software memory capabilities.

Set up a render for compositing

Rendering in layers refers to the process of separating scene elements so that different objects or sets of objects can be rendered as separate images. The first step is to determine how to divide the scene into layers. This may be very simple or incredibly complex and will depend entirely on your needs for any given project. Once you have decided how you want to separate your scene elements, there are several workflow approaches you can use to render them separately.
Rendering with render layers

A typical approach to separating your scene elements is to use Render Layers. You can assign objects to render layers using the same workflow as you would when working with display layers.

Render layers allow you to organize the objects in your scene specifically to meet your rendering needs. The most basic approach might be to separate objects into foreground, midground and background layers. Or, you may decide to divide the scene elements by specific objects or sets of objects.

If you need very precise control over the color of your rendered objects, separate from the shadows on them, you can further break down your shot by rendering separate passes within any render layer. The term render passes generally refers to the process of rendering various attributes separately such as color, shadows, specular highlights, etc. The Render Layers Editor allows you to set this up.

The following images show Leon rendered with different render passes: specular highlights (left) and diffuse (center). The last image on the right shows the resulting composited image.
The alpha channel

When rendering objects for compositing, one of the most important requirements is an *alpha channel*. The alpha channel, sometimes called a *mask* or *matte*, contains information about the coverage and opacity of objects in an image. This information is later used by the compositing application to combine the images.

In the alpha channel, opaque regions of objects are white, and fully transparent objects or empty spaces are black. The grayscale regions in the alpha channel mean semi-transparent objects.

The following image shows the alpha channel for Leon and the boat.

![Alpha and RGB channels](image)

Matte opacity

There are many cases where compositing the separate elements of even a simple scene can be tricky and require careful planning.

The following image depicts the compositing of two separately rendered objects. A problem exists where, for example, Leon stands in the boat, and his geometry is both in front of and behind some of the boat’s geometry. This is because the alpha channel does not contain any information about what part of an object goes in front or behind other objects. For this reason, the compositing application doesn’t know this information either.
The Matte Opacity feature provides one way to resolve this dilemma.

Separately rendered objects that will be difficult to composite correctly

Note: In some cases, it is also possible to affect the alpha channels later, in the compositing application, to allow images to composite correctly. A third possible approach is to render the images with a depth channel for use in compositing packages with depth compositing capabilities. However, there are limitations to depth compositing techniques, so it is a good idea to learn these other methods as well.

To ensure that the objects composite properly, you can use an attribute called Matte Opacity, found in the Attribute Editor for all materials. This allows you to manipulate the rendered alpha value on a per-material basis.

Matte opacity found in any material’s Attribute Editor
The **Matte Opacity** feature has three modes:

**Black Hole**

To solve this particular compositing problem, the **Black Hole** mode is useful. This mode will set the RGBA values to exactly (0,0,0,0), resulting in images with cutout regions that allow the objects to fit together correctly. The image below shows the alpha of Leon once the boat materials have been set to **Black Hole**.

![Black Hole used to hide parts of the objects](image)

**Opacity Gain**

This is the default mode for **Matte Opacity**. Alpha values are calculated in the normal way, and then multiplied by the **Matte Opacity** value. Because the **Matte Opacity** attribute has a default value of 1.0, the rendered alpha values remain unchanged (1.0 * x = x). However, you can adjust the matte opacity value to achieve the following effects:

- Animate the matte opacity value from 0-1 or vice versa to create fade-in or fade-out effects when composited.
- Texture map the matte opacity attribute to create interesting compositing effects, especially if you use an animated texture or sequence of images.
Solid Matte

When Matte Opacity is in Solid Matte mode, the normally calculated alpha values are ignored in favor of the matte opacity setting. The entire matte for the object is set to the value of the matte opacity attribute. This can be useful if you need an object to have a specific alpha value. For example, if you have a transparent object, the normal alpha value calculated by the renderer will be 0. Solid matte can be used to set a non-zero value for the alpha on the transparent object. If you were rendering a view through a window and wanted to composite that into another scene, setting the matte opacity value to 1.0 (in solid matte mode) on the window’s material would help you achieve this.

Note: Opacity gain and solid matte modes will not change the RGB component of your image. They will only change the alpha value generated by the shader.

Altering the mattes in a compositing application

Depending on what effects will be used at the compositing stage, it is sometimes important to render the whole object rather than having parts cut away with black hole. This gives you greater flexibility for effects such as blur, or overcoming moiré patterns on edges. Under these circumstances, you would need to use techniques in the compositing application in order to composite the elements correctly. This can involve manipulating a combination of the alpha values themselves, or creating custom masks to reveal/conceal objects as they are layered together.
mental ray blurred reflections and refractions

Typically, raytraced reflections and refractions exhibit very sharp definition. In reality, there are always inaccuracies in surface finishes and impurities in material structures that cause light rays to be reflected and refracted slightly off the original ray direction.

![mental ray reflection and refraction blur of a material](image)

You can adjust mental ray reflection and refraction blur in the mental ray section of the material node.

**Reflection and refraction blur**

These attributes determine the amount of reflection and refraction blur. A good starting point is between 0.1 and 0.3.

**Reflection and refraction rays**

When reflection and refraction blur have been enabled, ray direction is not exactly determined by the raytracing algorithm. Reflection and refraction rays will randomly deviate as specified by the blur attributes. This attribute is used to control the amount of *supersampling* required by the random deviation of the ray direction. Generally, higher reflection and refraction rays are required with more blur.

**Use Background Shader**

You can also use the Use Background Shader to make 3D geometry look like it is part of a real image. For example, if you want to place Leon over a background shot of a desert, you will need his shadows to be on the sand. Doing so would greatly help to enhance the integration of CG elements into the background image.

1. **Scene file**
   - Open the file called 08-waterfall_01.ma.
2 Image plane
  • Create an Image Plane for the camera.
  • Browse for the image called waterfall.tif from the sourceimages directory.

3 Environment
  • Hide the set.
  • Model a stand-in geometry that represents the content of the image plane.
    *If you model a geometry similar to the content of the reference image, your shadows and reflections will perfectly match upon compositing.*
  • Assign a Use Background Shader to the stand-in geometry.
    *This will make the geometry disappear seamlessly into the background image, but it will catch shadows and reflections.*

4 Lights
  • Create lighting similar to the one in the reference image.
  • Turn On shadows on the lights.
  • Render the scene.
    *The stand-in geometry will receive shadows, creating the illusion that Leon is actually part of the image.*
Tip: The same approach can be used to make a 2D image on an image plane look like it is part of a 3D scene. Use the same technique for modeling stand-in geometry: Assign a Use Background Shader to the stand-ins. With the stand-in geometry casting shadows and raytraced reflections of other geometry in the 3D scene, it is very convincing.

Camera projection

The Use Background technique described above reaches its limit in a case where, for example, you decide you want to be able to animate something that is getting its color from part of a 2D image plane. This might be a case of making a dog talk or a cat’s eyes bulge open, where the dog and cat exist in a live shot behind stand-in geometry. In this case, you can use the As Projection method of texture mapping to project the 2D image onto the stand-in, making sure that the Projection Type on the projection node is set to perspective and the Link to Camera attribute is set to the appropriate camera. Then you would do a Convert Solid Texture to create parametric texture maps on the surfaces. Once this is done, you can animate the stand-in geometry and render it so that it can be composited with the original images.

Composite rendering

If you find yourself in a situation where you are rendering an object over a background that is any color other than completely black (0, 0, 0), you should set Premultiply to Off in the Render Settings under the Render Options section.

What this feature does is prevent the edges of geometry from being anti-aliased against the background color. For this reason, the RGB component of the image will look badly aliased. However, the mask channel is perfectly anti-aliased. The mask channel is what is used to blend the rendered element into the background of choice at the compositing stage. Because the composite rendering flag prevented the edges from including any of the rendering background color, you will not get an unsightly rim showing in the rendering background color after compositing.

Premultiply Threshold is mainly a games feature. This is a normalized [0, 1] alpha threshold; the foreground is registered only if the alpha value is above the composite threshold.

Conclusion

Adding effects enhances a scene’s quality and produces some interesting results. Compositing involves rendering a scene in separate components and then merging those components together.

In the next lesson, you will review hardware rendering.