Exploiting the Power of Softimage's Render Tree

By Lucy Burton

Shaders are in effect little computer programs that drive how Mental Ray® rendering software reacts during the rendering or post-processing of a scene. The *Render Tree* in Autodesk® Softimage® allows you to create dynamic property assignments by connecting shader nodes, and then bundle them into presets and shader compounds via this visual interface, thereby creating an entire library of custom material shader assets that can be traded within your production house as presets that are readily applicable across many scenes. As a result, your ability to develop complex worlds efficiently, increases exponentially.

The nodal system lets developers combine not only shader and texture properties, but also includes tool nodes that perform additional functions such as mathematical calculations & pattern generation, as well as mixing properties, creating custom blends, & atmospheric effects, drive lighting, plug in camera lens properties, map displacement effects, create non-photoreal effects like toon shading, and hyper-realistic lighting via environment shaders using HDRI. It truly is one of the most versatile toolsets in the world of 3D, and a game-changing development that has had profound ripple-effects into every other toolset on the market.

Lighting With Paint

Lights and material surfaces are irrevocably connected. Just as a Bauhaus artist like Laszlo Moholy-Nagy or a director of photography like John Alton "paints with light", Impressionists like Monet, Degas, & Cassatt, and set designers like Josef Svoboda "light with paint" (not to mention a bevy of other materials). Figure 1 provides and example of lighting with paint.

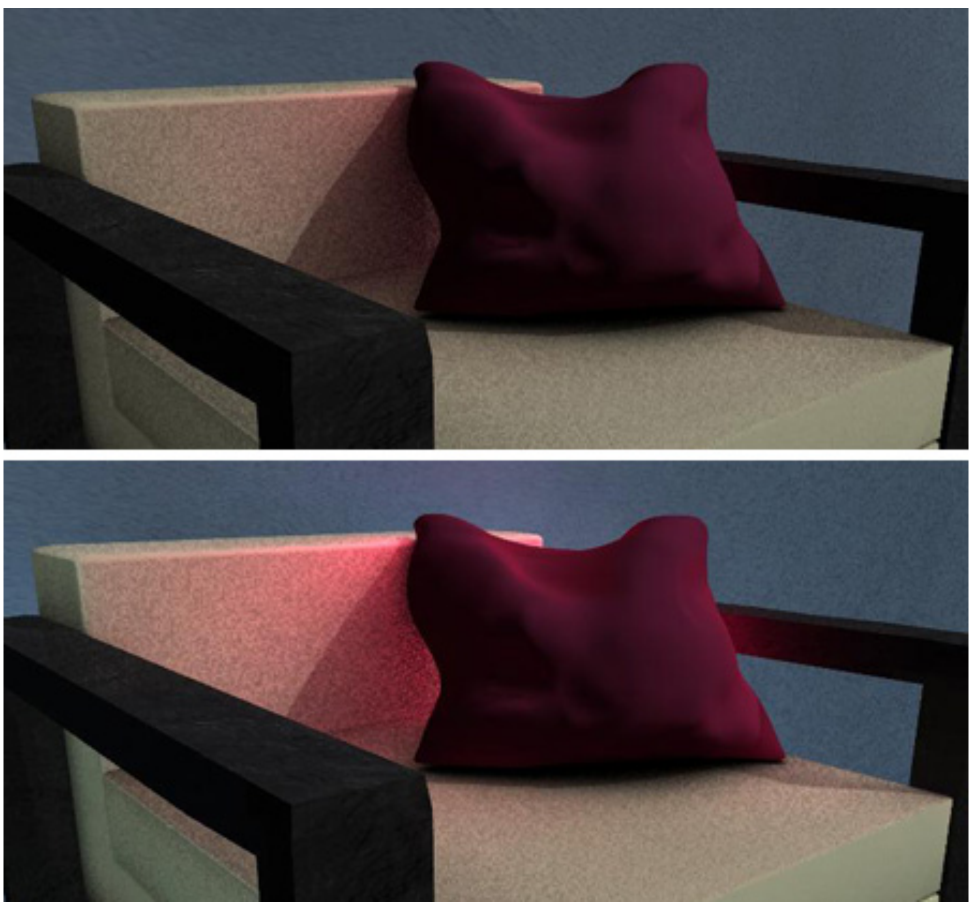


Figure 1. *As you can see from these two images, by adjusting only the radiosity value of the Lambert Material shader node, when a scene is lit via global illumination, the color from the pillow bounces off onto the cushion of the chair, and vice versa. The resulting color is a blend which mixes the beige of the cushion with the maroon diffuse and dark purple ambient colors of the pillow. In the bottom image, the radiosity values have been increased, and so the pillow throws more color into the surrounding scene. This is an example of lighting with paint, because the light itself is merely standard white light with identical values in both scenes, and the material parameters chosen within the Render Tree are driving the effect.*

At the most rudimentary level, paint uses subtractive color mixing, while light uses additive color mixing. In the realm of 3D, the artist must be cognizant of both these techniques and how they work in conjunction with each other, along with the complimentary colors they produce. In theatrical & film lighting as well as scenic design, you should be knowledgeable about the precise characteristics of various physical materials, be aware of how much "bounce" a color will produce on other surfaces when lit, and what the choice of a particular color of light will do to change the color characteristics of surfaces. As Figure 1 demonstrates, this becomes extremely important when global illumination algorithms such as radiosity are calculated, as they are designed to mimic these natural effects of light, in combination with particular material colors.

Dutch painters of the 17th & 18th Centuries used contrasting surfaces, like metallics against light-absorbing objects like matte woods, or velvet fabric to create visual interest, as well as patterned objects near single-color surfaces. Naturalistic lighting scenes typically exploit a complimentary color palate, so in daytime the key light would be simulating a yellow-hued sun, while a fill light would emulate the blue of the sky. At night, this scheme would be reversed with the outdoor light having a blue hue, complimented by warm-colored interior lights. By choosing material shader colors in much the same way, you can highlight or intentionally de-emphasize certain scene elements, and significantly effect emotional responses to your scenes. In the television show "Buffy The Vampire Slayer" which for obvious reasons included many night shots, you'll find the cinematographers often lit the background with blue light, whereas foreground objects and characters are lit and/or painted & dressed with warmer tones, to differentiate them from the background.

What 3D adds above and beyond real-world physical lighting, is the ability to do things like create lights which interact with only one surface characteristic of an object like it's specular highlight, but not it's ambient value, or which have a negative illumination value that act as dimmers for certain objects within a scene, or programming photons or materials to alter RGB values based upon their proximity to the camera via a **Z-Depth shader**. All these effects are driven in Autodesk® Softimage® with the Render Tree.

Node-Based Material Creation

As with most things in Autodesk® Softimage®, there are multiple ways to approach any task. You can develop Mental Ray shader parameters directly at the code level, or use a custom visual user interface (UI) called the *Render Tree* to achieve the same effect. This flexibility invites more people into the creative process, allowing both technicians and artists to work cooperatively to achieve unique effects, and increased realism. Within the context of this article, we will discuss but a few aspects of the complexities inherent within the Render Tree.

Every new scene you create will have a default shader in the scene's root named **Scene_Material** applied to it. All objects within your scene will have this default material until you assign a different one to it. You do not want to enter the Render Tree and just start attaching material or other properties to this node, or you will change the material characteristics of all the objects in your scene. Instead, you want to begin by picking your object, such as a car fender, and assigning a particular shader type to it, such as a Phong or Blinn shader, which come with specular parameters that will allow you to simulate the shininess of metal. You do this by going to the left-hand toolbar and selecting **Get> Material**, and selecting the appropriate shader from the drop down list.

The Render Tree itself can be accessed via a drop down options available in each viewport within the interface. By clicking the **Update** icon (which looks a bit like the rotating arrows of a recycle symbol), you can refresh the Render Tree window and see your shader network in order to change parameters or attach new nodes.

The three primary aspects of a basic shader are:

- **Diffuse** - The primary color of the object.
- **Ambient** - Multiplied by the scene's ambience value, it simulates non-directional lighting. Pixar often uses a very particular technique where the diffuse and ambient colors on an object are opposite from each other on the color wheel, creating an elegant softness to the object or character.
- **Specular** - This defines the shiny highlights on a surface. Specular decay determines how large the highlight is.

The basic material shader selections available in Softimage are:

The **Constant** shader provides a single, uniform, flat diffuse color to an object. This is excellent for applying as a base for background images to simulate far-off landscapes.

The **Lambert** shader has ambient and diffuse capabilities, but no specular values to simulate matte surfaces like dirt, rock, or cloth.

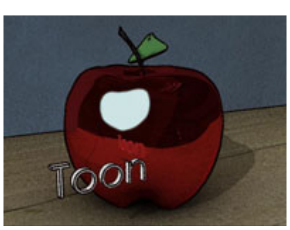
Phong has ambient, diffuse, and specular properties, and is excellent for creating plastics, metallics, or glossy ceramics like porcelain.

The specular highlight on a **Blinn** shader pictured here is brighter than on the Phong shader and is better at reflecting lighting when there is a high incidence angle between the lights and the camera. Perfect for metals or glass. As you can see by comparing these shader images, this is probably the best shader to use if you wish to create a realistic-looking apple.

Cook-Torrrance is excellent for creating materials that have a broad, rough specular highlight such as leather.

The **Strauss** shader uses only the diffuse color input to simulate a metal surface. A specular highlight is created by adjusting the smoothness and metalness parameters. With the Strauss shader you can create metals that have dual qualities highly reflective at one angle, and diffuse at other angles.

The **Anisotropic/Ward** shader is used to create brushed metal surfaces such as anodized aluminum. The specular values are calculated using the UV coordinates of whatever texture you plug into it.



Toon shading creates a combination of a 3D look and a traditional cel-shading look. The appearance can change radically depending upon how you arrange your lights. When combined with the Toon Ink Lens shader attached to a Render Pass Camera, you can create ink lines that mimic traditional cartooning. This is fantastic for creating vector animations usable in Adobe® Flash®.

The key thing to understand when working in the Render Tree is that you are creating a directed acyclic graph (flow chart) of processes. A Material node alone contains a drop down list of input plugs including **Surface**, **Volume**, **Environment**, **Contour**, **Displacement**, **Shadow**, **Photon**, **Photon Volume**, **Bump Maps**, and **Realtime** capabilities. Each type of input plug is color-coded so that you can tell which type of node you can plug into it via a connection arrow. If you attach a network of shader nodes to the Surface input plug, you can then control additional values such as transparency, specular reflectivity, incandescence, and color (see Figure 2).

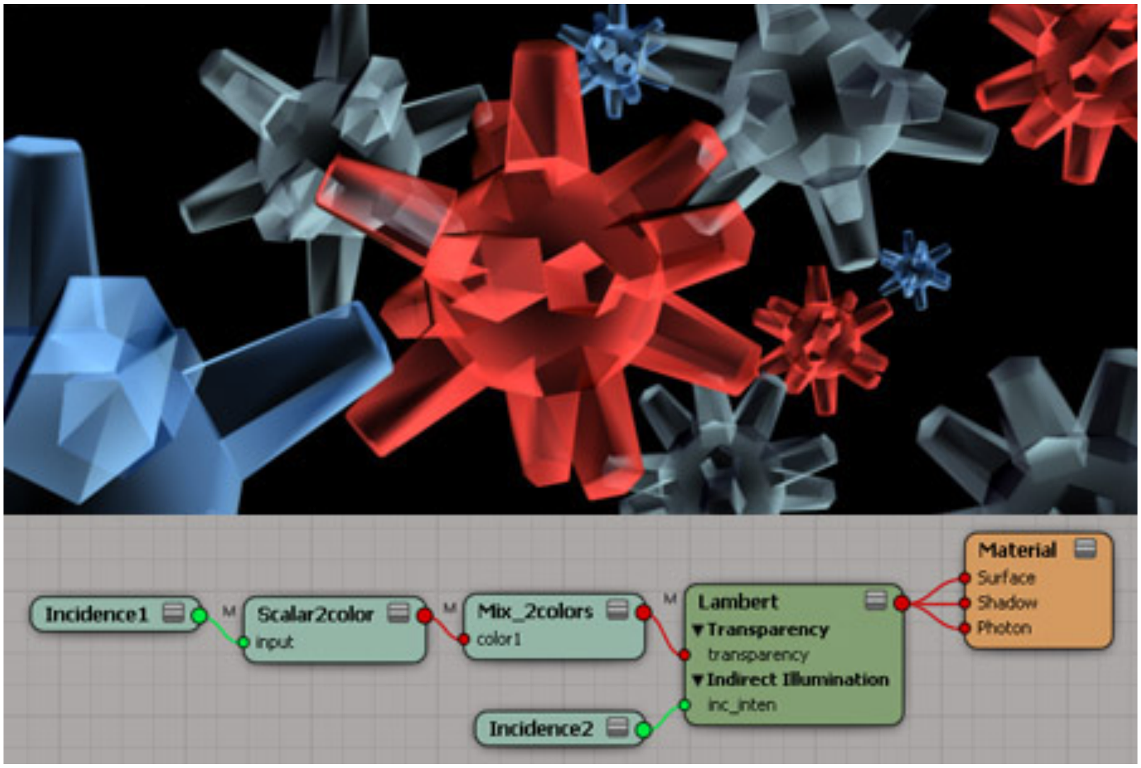


Figure 2. *Here is an example of material tree which creates an x-ray effect. The Incidence node is used to tell the shader how to display the shading values based upon the angle of the surface relative to the direction of the camera. You can also tell the node to change those values based upon it's relationship to a light or a particular vector. The second Incidence node drives the intensity of the color returned by a photon-based renderer such as global illumination or final gather. The Scalar2Color node converts the scalar or numeric values derived from the Incidence node into RGBA values. The Mix2Colors Node is used in this case primarily to drive how the alpha transparency is mixed (in this case multiplying the weight of the alpha by calculating the average color value of each pixel via the default Mix Mode), then output to the transparency input of the main Lambert Material Shader node.*

The nodal connection parameters are:

- **The Red Color Input/Output Plug** - Returns or outputs an RGB/HSV value for use when defining the characteristic of a surface, light, or camera.
- **The Green Scalar Input/Output Plug** - Gives a value of between 1 and 0.
- **The Yellow Vector Input/Output Plug** - Determines coordinate positions. When used as an input plug, it typically maps the position of a texture, and when used as an output plug, it returns a specific vector position value.
- **The Purple Texture/Image Clip Input/Output Plug** - Allows you to apply a specific image file to drive effects, or generate surfaces.
- **The Orange Boolean Input/Output Plug** - Generates a binary value of either 1 or 0, like an on/off switch.
- **The Blue Integer Input/Output Plug** - Generates a single numerical value (6, 34, 12, etc.)
- **The Brown Real Time Input/Output Plug** - Accepts connections from other realtime shader generators.

Creating Material Presets & Managing Shader Libraries

You can also view Render Tree shader networks and manage your multiple materials via the **Material Editor**, by pressing **CTRL-7**. As you can see in Figure 3, the top section of the pop-up window will contain mini-preview images of the output result for each shader network you've created. If you click it, you can view its shader network in the bottom window by clicking on the **Render Tree** tab at the base of the window. From there, you can also see what texture layers have been applied to that object and which image clips were used to create it, as well as obtain a list of which geometric items within your scene are currently using that shader network. This is particularly useful in large, complex scene.

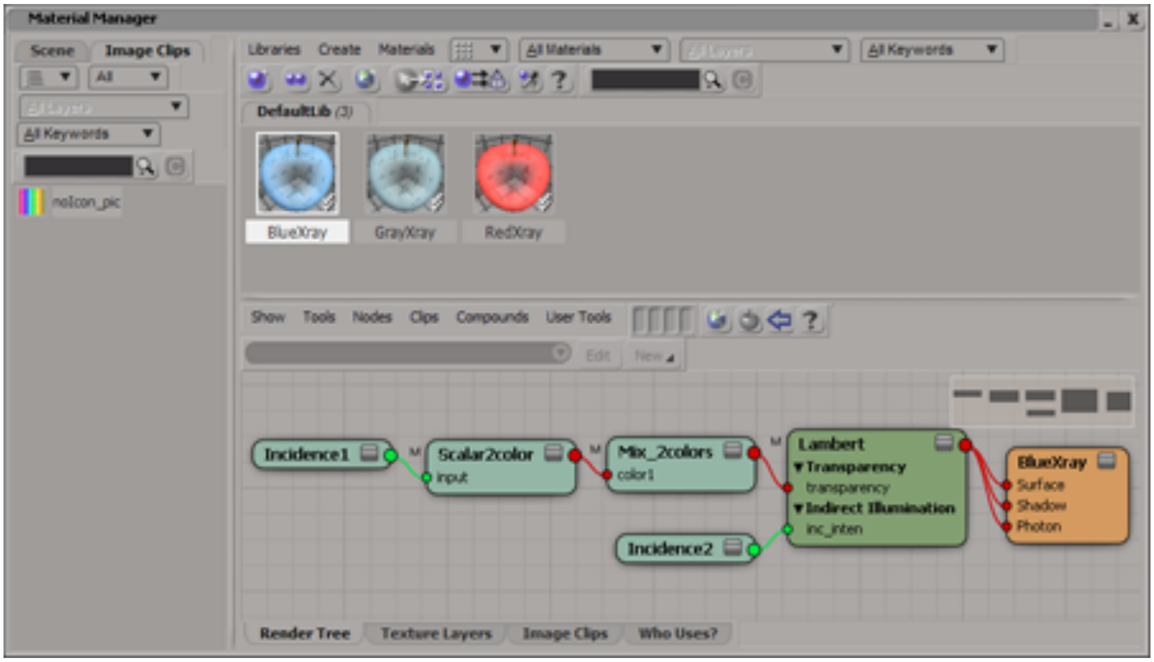


Figure 3. *The Material Editor.*

Additionally, you can drag material presets from this window by clicking on one of the preview images, dragging it over a piece of geometry in another viewport, and releasing the mouse once you've seen a plus sign (+) appear. This is a fast way to apply a single material & texture network to multiple items and helps you to optimize the size of your scene. The fewer materials the software has to calculate for, the faster your scene will render.

Once you've created a shader network in Render Tree, you can save that network as a single **DS Preset** that you can reuse on multiple pieces of geometry or save to a library that will allow you to load it into future projects (see Figure 4). Within the Render Tree, networks are read right to left, with the final result being furthest to the right. That last node can be through of as the primary root of the tree.

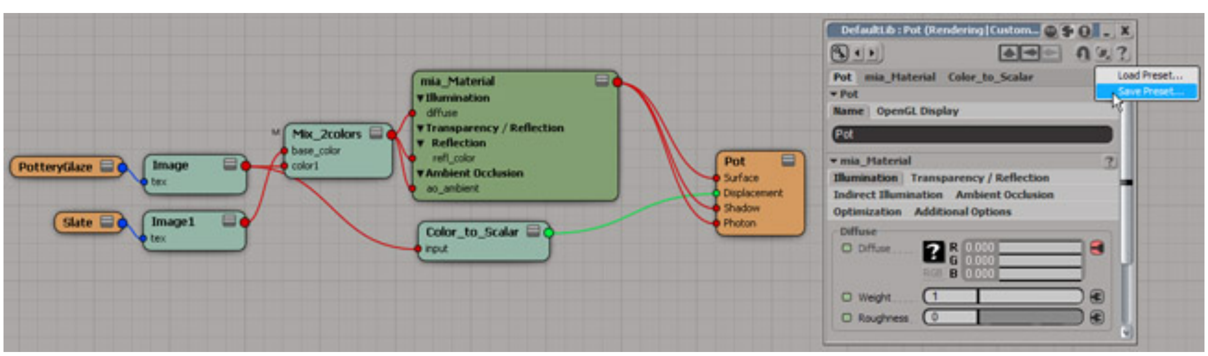


Figure 4. *Method for saving a DS Preset.*

If you double-click on that node, and store it from the **Save** icon in the top right corner of the pop-up editor, you save the entire tree. The only caveat to this is that if you are saving a shader network that includes an image file used for texturing, you must either leave that image file where it is on your central drive, or remap to it's new location (if you load it onto another computer).

Bump & Displacement Maps

Displacement shading maps allow you to add super-fine detail to your objects, without having to actually model them. This is excellent for rough things such as wrinkles, grooves, & crusts which would otherwise be too time-consuming to build poly by poly, and would ultimately weigh down your model and increase render times. As a result, displacement shading mapping is superior to bump mapping.

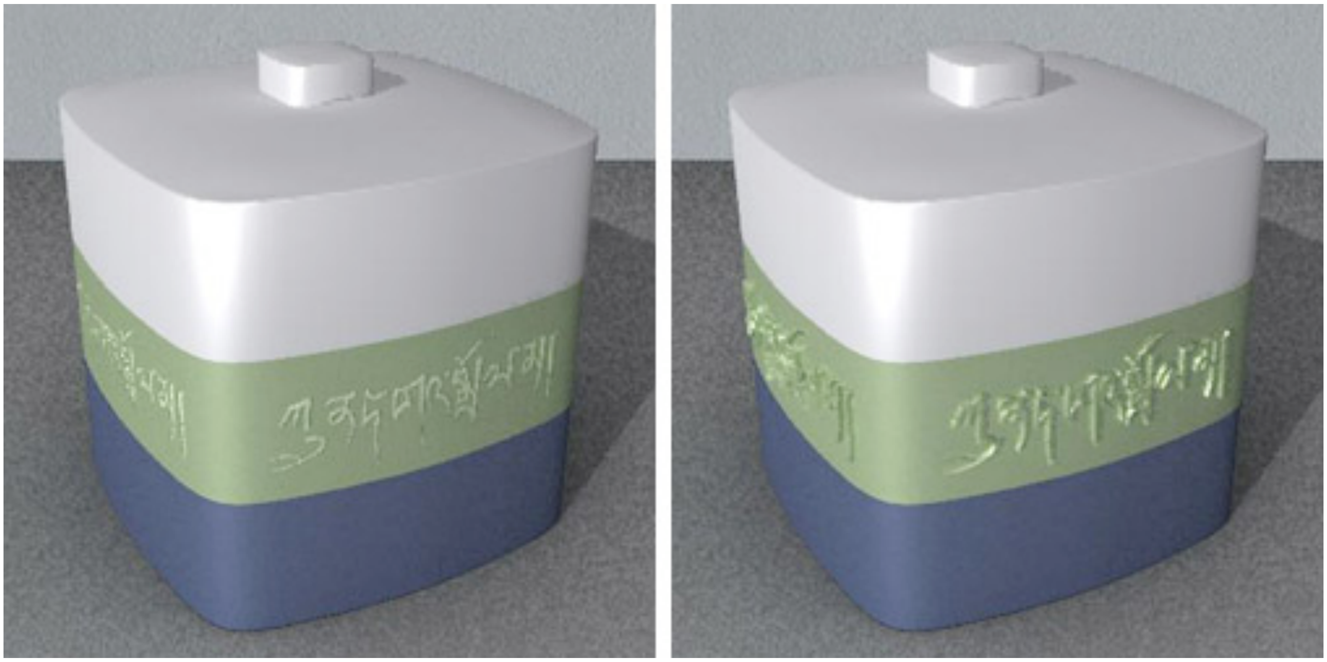


Figure 5. *The image of the Bento Box on the left shows a bump mapping applied to the Tibetan script printed on the sides of the green segment. However, as the name suggests, the displacement mapping in the image on the right does more than simply provide the illusion of roughness: It actually calculates what it would look like if that script were physically modeled geometry. You'll notice as well that while bump mapping does relatively good job of simulating fine detail roughness on an object if the geometry in question faces the camera, it doesn't do as good of a job as displacement does on surfaces at a bias angle away from the camera. For large-scale cityscapes, displacement mapping is an excellent way to add finite detail without exploding your render times with the increased polygon count from modeled geometry.*

To create a displacement map within the Render Tree, you must plug an Image Clip with an alpha channel into an Image node, then connect that to the Input of a Color_to_Scalar node that will translate the color values into an integer. That scalar output is then plugged into the **Displacement** parameter on the main Material node. To drive the effect, however, you must go into the **Explorer** viewport in Autodesk® Softimage®, and double-click the **Geometry Approximation** value of the selected geometry. The software will prompt you asking if you wish to create a local copy: I recommend that you say yes so that the only values you alter are those related to this one object in your scene. Once the **Geometry Approximation** dialogue box pops up, click the **Displacement Tab** and then click **Fine**. Next, select the **View Dependent** check box to get the most precise and accurate detail from the displacement, and adjust the slider values appropriate for your particular scenic object.

Summary

I've discussed but a few functions of the Render Tree within the context of this article, but truly, the possibilities here are as close to infinite as I've ever seen. This nodal method has also been used to create a secondary toolset within Autodesk® Softimage® of equal power and dynamism, called the *Ice Tree*, which applies the same programming logic to the production of particle effects. Once again, something which previously was accessible only to code-level technical directors and programmers, now can be executed by artists, shared with programmers, and bundled into custom presets that can be easily exchanged and refined.

The next article will go into further detail as how to apply Final Gather and Global Illumination in combination with HDR imagery (HDRI) lighting in Autodesk® Softimage® to produce realistic reflections. It will also discuss the future of realistic rendering and optimization.

Additional Recommended Reading

Barbara J. Meier. "Painterly Rendering For Animation." ACM SIGGRAPH Annual Conference Proceedings 1996.
M.E. Chevreul. "The Principles of Harmony & Contrast of Colors: And Their Application to the Arts."
Greg Ward Larson and Rob Shakespeare. "Rendering With Radiance: The Art & Science of Lighting Visualization."
Jarka Burian. "The Scenography of Josef Svoboda."
Francois X. Sillion and Claude Puech. "Radiosity & Global Illumination."

About the Author

Lucy Burton was raised in Europe and returned to the United States for college, graduating with honors from Seattle University with a degree in Drama/Political Science and obtaining Film Certification at New York University. She interned in Technical Direction at Intiman Theatre. Having worked professionally both in theater and in production on several films, she then moved into the postproduction/visual effects realm, first working on Softimage 3D Extreme at Mesmer FX. Lucy has been working with the XSI platform since its inception nearly a decade ago. After founding her own digital design studio in 2001, she went on to produce documentary videos on the humanitarian crisis in Indonesia following the tsunami and helped a nongovernmental organization that assists victims in Darfur, Afghanistan, and Uganda, among other crisis areas. She now is freelancing in and around Hollywood, California.