Poser Materials
Concepts & Elements

Poser offers a lot of tools and options to shape, stage and animate the objects in my scene. But to turn them into an interesting and colorful image or movie, Materials are what I need.

This document discusses the buzzwords and the way things work (together) as far as the Poser Material Room is concerned. It also discusses the interactions with Poser Lighting and with the Poser Firefly renderer. Each chapter addresses just the question stated in its title, and all chapters are meant to be accessed separately in a random order as a website with internal links. As a consequence, repetition of introductory remarks and duplication of some content is introduced intentionally to support easy reading individual articles, instead of having to click around for a complete picture. My apologies for any inconvenience caused that way.

Further documents on Poser Materials will discuss:

- Real Life Materials & Poser Photorealism.
  Good lighting as well as a dedicated materials setup are essential in establishing “photo real” results (with the Poser Firefly renderer) at any level of quality. In most cases a proper understanding of the way things work in real life, in nature, does help a lot – to say the least. This section discusses those materials setups, and their real life backgrounds.

- Poser Material Tips, Tricks, Techniques, Tools & Context.
  This section discusses tools and tricks in using the Poser Material Room, in creating and handling materials. This section also offers summaries of and references to the relevant threads in the various forums.

- Interfacing external renderers.
  These sections discuss the transfer from Poser scenes to other rendering systems, like Octane, Reality / LuxRender and Vue. Generally, 3D meshes are well transferred, lights are hardly transferred at all and have to be rebuild in the ‘other system’, while material definitions are translated partially, and might be dealt with in a similar, or in a completely different way.
  So, understanding how materials behave in the interaction with those external renderers is closely related to the Poser material definition setup.

April 2014

Contents

Poser Material Concepts & Elements

This document discusses to some extent the buzzwords, the interactions with Poser Lighting and with the Poser Firefly renderer, and the way things work (together), all as far as the Poser Material Room is concerned. It discusses intensively the technical details of all elements that build the shader definitions within the Material Room.

That's quite a lot, and for that reason the information is presented in various sections:

I Introduction.
The articles in this section discuss some terminology, and the various interfaces to the materials definitions:

- 01 What's a material, a shader, a texture, a map?
- 02 How do I access a Material?
- 03 Material Room offers a Simple interface. What do I miss?
- 04 Material Room Simple and Advanced interface - how do these relate?

II Simple Surface Definitions.
The articles in this section discuss material definitions for object surfaces, which can be handled through the Simple Interface, and do not require a deep understanding of Material Room principles. Each article however also presents the Intermediate approach, using the Advanced interface for the same subject at hand. This is to avoid multiple articles answering the same question.

- 05 What's the Diffuse component intended for?
- 06 What's the Specular / Highlight component intended for?
- 07 What's the Ambient component intended for?
- 08 What's the Transparency component intended for?
- 09 What's the Reflection component intended for?
- 10 What are the Bump and Displacement components intended for?

III Intermediate Surface Definitions.
The articles in this section discuss some material definitions for object surfaces (the PoserSurface), which are handled through the Advanced Interface: the nodes from the Lighting group, and the nodes on image-maps and movies. It also discusses some principles on dealing with the PoserSurface root node.

- From 20 A PoserSurface material offers Diffuse, Specular, etc. How do these work together? on, articles discuss the workings of the PoserSurface root node in general.
- From 30 Can I get a (simplified) explanation on Lambert and diffuse shading? on, articles discuss some elaborate details of components available in the Simple interface: diffuse shading, reflection details, and render settings.
- From 35 What's the Shadow Catch Only intended for? on, articles discuss the additional components of the PoserSurface node, like Translucency, Refraction, ToonID and the like.
- From 40 What are the Alternate_Diffuse and Alternate_Specular components intended for? on, articles discuss Alternate_Diffuse, Alternate_Specular and all the nodes from the Lighting > Diffuse and Lighting > Specular groups.
- From 51 What's the Special > Skin node used for? on, articles discuss the nodes from the Lighting > Special group, like Scatter and Hair. In 58 What's the difference between all those Scatter nodes? the various scatter nodes are compared.
- From 59 What's the Raytrace > Reflect node used for? on, articles discuss the nodes from the Lighting > Raytrace group. In 64 How do I properly combine Transparency and Raytracing? their complex relationship with Transparency is dealt with.

- From 65 What's the Environment Map > Spherical Map node used for? on, articles discuss spherical mapping, image maps and movie-based textures.

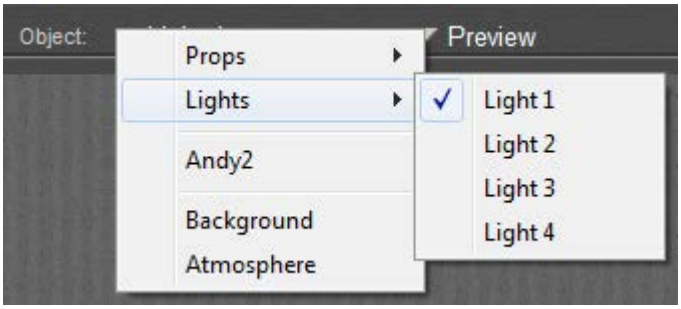
IV Advanced Surface Definitions.
The articles in this section discuss all Material Room nodes required for either procedural textures, and the ones explicitly aimed at node-tree building.

A procedural texture is not derived from an (eventually color filtered) external image or movie still, but is mathematically generated internally from surface or spatial coordinates. The nodes to accomplish such textures can be found in the 2D Textures (70 What are the 2D surface texture building blocks used for?) and 3D Textures groups (71 What are the 3D surface texture building blocks used for? and on).

Materials are applied to objects, objects parts and more precise: to specific Material Zones within those objects and parts. Details are discussed in 74 Material Zones, or: to which Body Parts do the materials apply to?.

Material Room supports the creation of quite elaborate node-trees, like a programming language into material definitions. This section will not address the art of such programming itself, but will present and discuss the building blocks alone. These can be found in the Variables (77 What are the Variable nodes used for?) and Math (81 What are the Math nodes used for?) groups.

V Materials for Non-Objects.
The articles in this section discuss properties for Atmosphere, Background and Lights. These are not objects with a surface, but do have properties which are handled in Material Room. These properties can be accessed via the Object selector



- The scene atmosphere is discussed from V1 Poser Atmosphere on.
- The scene / render background (color, picture or movie) is discussed from V2 Poser Background on.
- The coloring properties of Lights are discussed from V3 Poser Lights on.

Most of those topics are considered Intermediate level, although various configurations can be setup via Material Room menus, and can be managed through the Simple interface.
On the other hand, managing the details of a scene Atmosphere requires the use of nodes from the 3D Texture group, which by itself is considered Advanced.

This section concludes with some varied, advanced topics like mapping for IBL (97 How are Image Based Lighting, probeLight objects and IDL Sky Domes related?) and Gamma Correction (in 98 What's the fuzz about Gamma Correction? and 99 Gamma, Transparency and Blends).

I Introduction

The articles in this section discuss some terminology, and the various interfaces to the materials definitions:

- 01 What's a material, a shader, a texture, a map?
- 02 How do I access a Material?
- 03 Material Room offers a Simple interface. What do I miss?
- 04 Material Room Simple and Advanced interface - how do these relate?

Next sections present articles on defining the surface properties of objects, on a Simple, Intermediate and Advanced level:

- II Simple Surface Definitions
- III Intermediate Surface Definitions
- IV Advanced Surface Definitions

as well as on atmosphere, background, lights (V Materials for Non-Objects) as far as these are handled through the Material Room interface. The Appendix: Poser nodes / settings and versions lists all Material Room nodes and relevant Render Settings, and their availability in the various Poser versions.

01 What's a material, a shader, a texture, a map?

In real life, a **material** is the stuff something is made of. Rock, brick, sand, knitted red wool, thin leaded glass, anything. Real life materials not only have a look, they also have a feel, a smell, and a response to our actions determined by a weight, flexibility, and the like.

In virtual life, like a Poser scene, a **shader** refers to a set of object (surface) properties that mimics the looks of a real life material, when rendered. So we can have a rock shader, a knitted red wool shader, etcetera. Shaders do not have a feel, or a smell, they're inside the computer. But since everyone can tell real life from virtual, the word "material" is also used in these cases, at least in some software communities.
So, in Poser one has a Material Room, to make a "brick material" and to assign it to a wall in the Poser scene. In Poser communities, "shader" is rarely used.

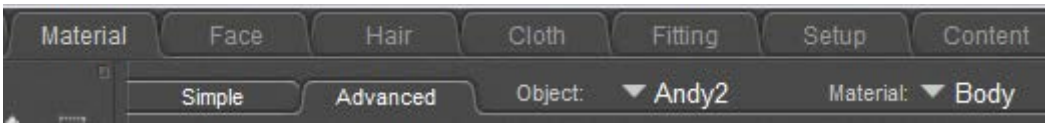
In real life, **texture** relates to the feel of the thing at hand. The surface roughness of the brick when I rub it with my hand, the structure of the fish I feel with my tongue when tasting it.
In virtual life however, texture usually refers to the colors of an object surface. A texture then is an image used to assign such colors to elements in my Poser scene. However, since people are somewhat relaxed in their choice of words, they're happy to assign a "brick texture" to a wall; not only implying color but roughness and reflectivity as well. So in those cases texture means material means shader.

While texture usually refers to an image which is used to assign colors to a surface(property), a **map** refers to an image which is used to vary the amount of something. A bumpmap to vary the amount of roughness, a transparency map to vary the opaqueness, and so on. Maps in those cases tend to be black & white, which refer to 0% .. 100% and have greyscales for everything in between.

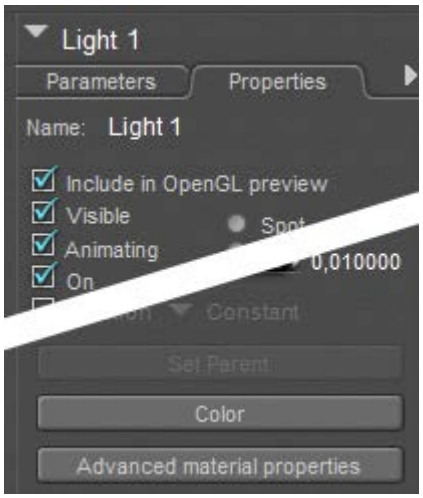
On the other hand, **mapping** (as in: UV-mapping) is the term for assigning images in general to an object surface whether it's for coloring or for determining roughness or reflectivity. So some people might use "map" while referring to the image driving the coloring process too.
Fortunately, there is some method in this madness: as shader is hardly used in the Poser community, material or texture is used instead. The people using material for the whole thing tend to use texture for the coloring images. The people using texture for the whole thing tend to use texture-map for the images. But be aware; without context or background info, "brick texture" still might mean either the whole thing or just the color-driving image.

02 How do I access a Material?

The straightforward way is to use the Material tab to enter the Material Room. An object or part of it already can have been selected, or can be selected from within the room itself. The Material Room offers a Simple User Interface, as well as an Advanced one. See 03 Material Room offers a Simple interface. What do I miss? and 04 Material Room Simple and Advanced interface - how do these relate? on them.



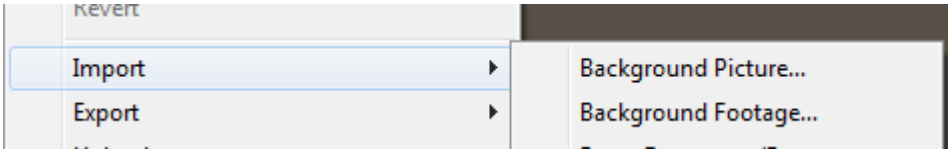
Intermediate



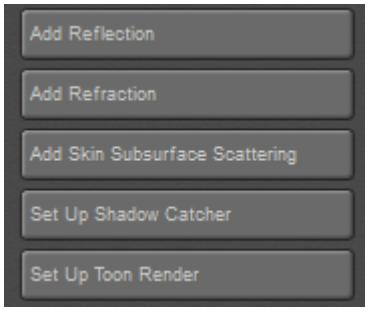
Next to that, there are some additional ways into Material Room, for the 'material properties' of Lights, Atmospheres, Backgrounds, and for some specific surface properties:

1) When a Light is selected, its Properties tab offers an [Advanced material properties] button which brings me into the Material Room, for the coloring properties of that light. See 93 How do I set the color and intensity of my light(s)? and on for details.

2) With menus File > Import > Background Picture or … > Background Footage,

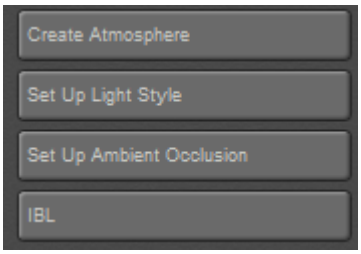


and with the Shadow Color picker just right/below the Document window one affects the contents of the background material. See V2 Poser Background and on for details.



3) From within Material Room some buttons on the right affect the object surface material at hand:

- Add Reflection and Add Refraction
- Add Skin Subsurface Scattering
- Setup Shadow Catcher and Setup Toon Render



The [Create Atmosphere] button however affects the Atmosphere material (see V1 Poser Atmosphere and on for details), while the next buttons

- Setup Light Style
- Setup Ambient Occlusion
- IBL

affect the various coloring properties for Lights (see V3 Poser Lights and on for details).

03 Material Room offers a Simple interface. What do I miss?

When using the Simple interface, I miss:

- About anything more advanced than assigning color and an eventual image map to any feature. As a result, my render will keep that artificial, hard plastic-like feel.

- The option to have Bump and Displacement both in one surface definition, and the option to use Normal maps.
  As a result, I cannot distinguish large scale (displacement) from small scale (bump) surface variations. And I can't use Normal maps, which are common in shading game characters and objects.

- Access to more real-life optical effects like Translucency and Refraction.
  As a result, creating believable glass and fluids will remain an issue.

- Access to the Preview / Diffuse / Specular split in direct light properties
  As a result, I'll keep on having issues with handling Indirect Lighting (IDL) in an appropriate way, in preview as well as in rendering.

- Access to advanced render features (Custom_output).
  Honestly, these are hardly used anyway and can be considered high-end pro stuff.

Intermediate

Generally, all features which remain unsupported by the Simple interface will also go unsupported when exporting Poser scenes and objects to other formats or programs. Exporting to OBJ, integrating Poser with LuxRender, Octane, Vue or you name it; all tend to lose the material properties which are not supported in the Simple interface. And even some of those might get lost in translation.

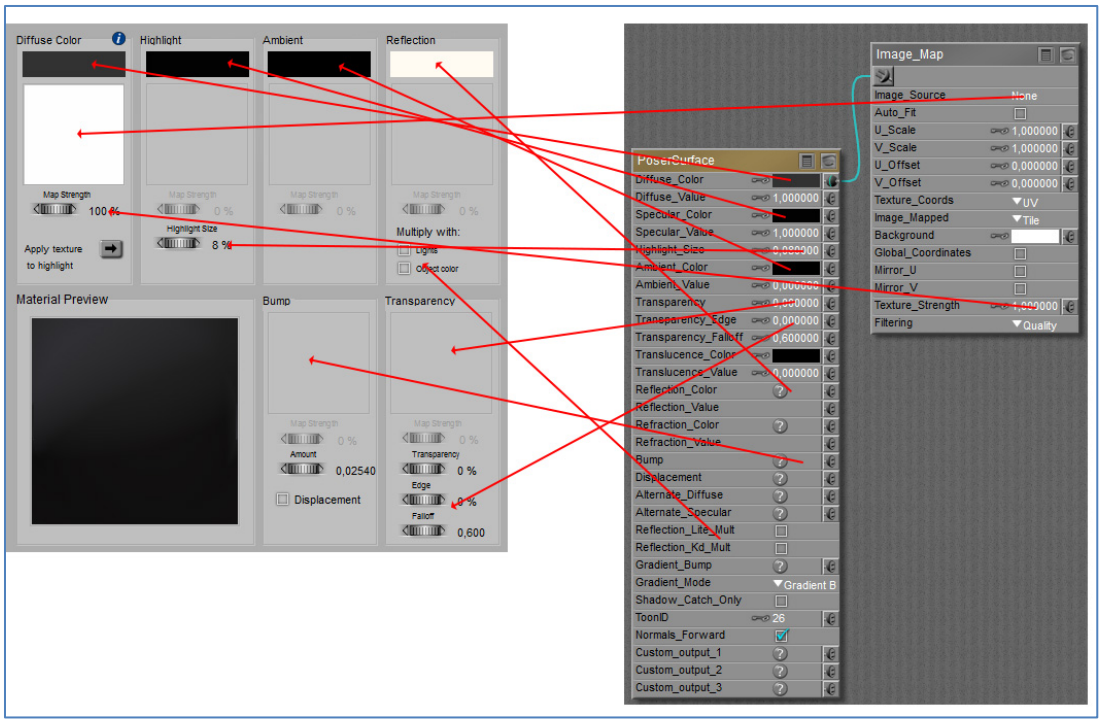
In other words: when Poser is just my scene building and posing tool but not my final renderer, I consider the Material Room Simple interface as the recommended one.

The question: what do I miss, can be inverted to: what elements from the Advanced interface go (un)supported by the Simple interface. This is addressed in 04 Material Room Simple and Advanced interface - how do these relate?.

04 Material Room Simple and Advanced interface - how do these relate?

The Advanced interface to Material Room offers access to some more properties of the same material on one hand, and offers access to far more ways to define the details of all properties in the other hand. Creating and managing materials through the Advanced interface is considered Intermediate to Advanced level. While working from the Simple interface, one might wonder: “what do I miss?”. That is addressed in 03 MATERIAL ROOM OFFERS A SIMPLE INTERFACE. WHAT DO I MISS?.

Intermediate



The following features are / are not supported in the Simple interface:

- **Supported**: Diffuse, Specular and Ambient Color, plus an eventual Image_Map or Movie node from which the Image_Source and the Texture_Strength properties are supported. For Specular, the Highlight_Size property is supported.
  Not supported: Neither the Diffuse/Specular/Ambient-Value properties, nor the Alt_Diffuse/Specular properties, nor any other node beside Image_Map and Movie are supported in the Simple interface. No Clay, no Subsurface Scattering (translucency). Translucency from the Advanced interface is not supported in any way.

- **Supported**: For Reflection, the Image_Map node as well as the Reflect (raytrace) node are supported, including the Color. Also the Light color and Object color multiplicators (Reflection Lite Mult and Reflection Kd Mult checkboxes) are supported.
  Not supported: Neither the Reflection Value property nor any other nodes beside Image_Map and Movie are supported.

- **Supported**: Transparency, with an eventual Image_Map, and including Edge and Falloff, is supported.
  Not supported: Refraction, Fresnel nodes and similar are not supported.

- **Supported**: Bump / Displacement, with a required Image_Map (no map, no effect), plus the Amount option are supported.

But: the checkbox in the Simple interface forces me to choose between either Bump or Displacement, I cannot have it both.

- **Not supported**: Features like Gradient Bump/Mode which give access to Normal maps, and like ToonID and Custom Output for advanced render pass handling, are not available in the Simple interface. The Custom_Outputs are available in Poser Pro only, by the way.

II Simple Surface Definitions

The articles in this section discuss material definitions for object surfaces, which can be handled through the Simple Interface, and do not require a deep understanding of Material Room principles. Each article however also presents an Intermediate approach as well, using the Advanced interface for the same purpose. This is to avoid multiple articles answering the same question.

- 05 What's the Diffuse component intended for?
- 06 What's the Specular / Highlight component intended for?
- 07 What's the Ambient component intended for?
- 08 What's the Transparency component intended for?
- 09 What's the Reflection component intended for?
- 10 What are the Bump and Displacement components intended for?

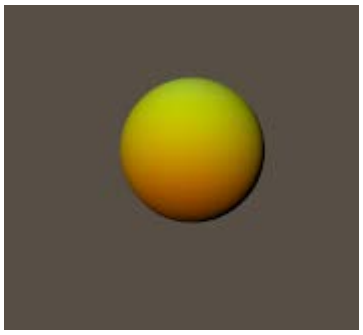
Next sections present similar articles on an Intermediate and Advanced level

- III Intermediate Surface Definitions
- IV Advanced Surface Definitions

as well as on atmosphere, background, lights (V Materials for Non-Objects) as far as these are handled through the Material Room interface. The Appendix: Poser nodes / settings and versions lists all Material Room nodes and relevant Render Settings, and their availability in the various Poser versions.

05 What's the Diffuse component intended for?

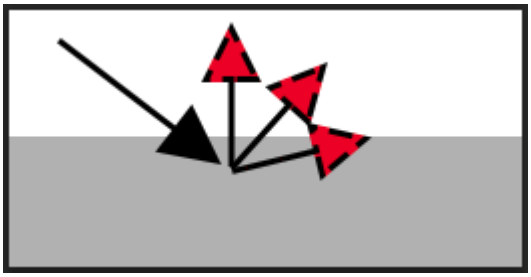
For short: Diffusion is equivalent to Object Color. This is the place to turn an object simply green, and/or to assign an image to it for a detailed coloring of the surface. The color swatch then works as a filter: when I assign an image as well as turn the swatch to green, it's like I'm looking at the image through green glasses or through a sheet of green transparent plastic. A white swatch means: no filtering, and is generally recommended when images are used.

makes 

Clicking the larger area opens the Texture Manager, which offers the option to import an image.

More in detail

In nature, objects get their color by scattering back some of the light that falls upon them, and do so in a color-filtered way. So white light shining on a plants leaf will make it look green because the leaf scatters back the green portion out of the white light, at the place and time the light hits the leaf. When the light is pure red without any green in it, then the leaf cannot scatter anything back and hence will look black instead. This "scattering back" is called Diffusion, not to be confused with Reflection as discussed in 09 WHAT'S THE REFLECTION COMPONENT INTENDED FOR?.



The amount of light which is received by a "unit of surface" (say 1 cm^2 or in^2) depends on the angle the light makes with the surface. Perpendicular lighting makes high intensities; skew angles make low intensities as the same amount of light has to shine on a larger area.

Also, the diffuse response to light usually will not be equal in all directions: the response perpendicular to the surface might be stronger than that parallel to it, making objects look darker at skew angles towards the camera. That is: at their edges.

Both effects are referred to as: shading, in contrast to shadowing which includes blocking the light by other objects, or other parts of the same object. In Poser all this is understood in the Diffuse part of the material definition.

Intermediate

The Advanced interface into Poser Material Room offers a **Diffuse_Color** which makes it the equivalent of Simple interface, and offers a **Diffuse_Value** next to it which acts as an extra filter. Intensities are reduced by that factor, and it can be driven by a (greyscale) image map as well. This way one can easily make dark stains on a surface.

See 04 Material Room Simple and Advanced interface - how do these relate? and 24 A PoserSurface material component, offers Color and Value. How do these work together?.

Recommendations:

- To prevent artefacts in rendering when applying the Gamma mechanism (recommended, available in PoserPro and Poser 10 and up, see 98 What's the fuzz about Gamma Correction? for details) the Value setting should be kept to 1.0 only (or 0.0 but no intermediate values).
- To prevent overlighting when combined with other aspects of lighting and material definitions, it's recommended not to exceed an 80% brightness in the Color-swatch and maps (or in the Value setting when not applying Gamma).

As said, in nature the diffuse response to light usually will not be equal in all directions. This was already investigated upon by the mathematician J.H. Lambert (about 1750). In Poser, this "Lambert diffusion" is embedded in the Diffuse part of the material definition (Simple and Advanced interface), as well as implemented in the diffuse node (Advanced interface only).
See 30 Can I get a (simplified) explanation on Lambert and diffuse shading? for more background on Lambert, see 41 What's the difference between the Diffuse component, and the Diffuse node? on details about the Diffuse node, and see 66 How can I assign an image to a material? or 68 How can I assign a movie to a material? on assigning either an image or a movie respectively to the Diffuse slot(s).

Anyway, the diffuse light is scattered outward, that is: with following the surface normal, which is a vector perpendicular to the surface which generally should be pointing outward. For various reasons the latter is not always the case, depending on the way the object is made and imported into Poser. If not, it can make the scattering go in the wrong direction, causing black spots in the render. The solution is to force Poser to reconsider the surface normals, and I can make it doing so by ticking the **Normals_Forward** checkbox. It's in the node, as well as at the bottom end of the PoserSurface definition itself. It's not available in the Simple interface.

That aside, the results from the "Lambert approximation" do not look utterly realistic for organic, porous surfaces. Therefore, Poser offers alternatives to the Diffuse component, like the Alternate_Diffuse part of the material definition (see 40 What are the Alternate_Diffuse and Alternate_Specular components intended for?), plus a Clay node with enhanced properties (see 42 What's the Clay node used for?).

On top of all this: in nature, **non-metallic objects** get their color from diffusion, while **metals** get theirs from reflection. Especially when I'm into photorealism, it will be good to understand – and to implement – this difference. Metals don't diffuse, they reflect only.

06 What's the Specular / Highlight component intended for?

For short: specularity makes highlights on an object surface, representing the (blurred) reflections of the direct lights (infinite, spot, point) in the scene. Although color and an eventual image map both work as an extra filter, such a coloring filter should be applied to make believable metallic surfaces only (metallic car paint included). In the vast majority of cases, the color swatch should be a shade of grey. Small highlight sizes represent hard, smooth surfaces.

Note that in Poser IBL (image based lights) and IDL (indirect lighting) cannot make highlights themselves; I do need regular direct lights in my scene to do so.

More in detail

When light hits an object surface, some portion of it might get reflected. In the good old days, when computers were short of raw CPU power, this handling of reflecting light rays was something to be avoided in order to get industrial render projects done in due time.

Faking reflections by using images of the surrounding scene ('reflection maps') was one way to get out (see 09 WHAT'S THE REFLECTION COMPONENT INTENDED FOR? on details), and another one was to concentrate on the – usually blurred – highlight producing reflections of the direct lights in the scene only. The latter is referred to as specularity.

In nature, as well in modern ray-tracing based renderers like LuxRender or Octane, there is reflection only, handling all sorts of light. But Poser supports the easy to calculate Specularity for direct lights and the more computational intense Reflection for the indirect light bouncing (or emitted otherwise) from objects in the scene. Two sides of the same coin, served separately for convenience especially aimed at reducing time and resource requirements at rendering. The downside is that I have to cater for both aspects separately in my material definition.

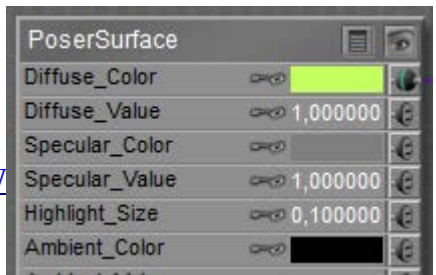
Intermediate

The Advanced interface into Poser Material Room offers a **Specular_Color** which makes it the equivalent of Simple interface, and offers a **Specular_Value** next to it which acts as an extra filter. Intensities are reduced with that factor, and it can be driven by a (greyscale) image map as well. This way one can easily make non-reflective stains on a surface. See 04 MATERIAL ROOM SIMPLE AND ADVANCED INTERFACE - HOW DO THESE RELATE? and24 A POSERSURFACE MATERIAL COMPONENT, OFFERS COLOR AND VALUE. HOW DO THESE WORK TOGETHER?.

Recommendations:

- To prevent artefacts in rendering when applying the Gamma mechanism (recommended, available in PoserPro and Poser 10 and up, see 98 WHAT'S THE FUZZ ABOUT GAMMA CORRECTION? for details) the Value setting should be kept to 1.0 only (or 0.0 but no intermediate values).
- To prevent overlighting when combined with other aspects of lighting and material definitions, it's recommended not to exceed a (100% - Diffuse) brightness in the Color-swatch and maps (or in the Value setting when not applying Gamma). So with Diffuse set to 80%, Specularity should not exceed 20%. See 20 A POSERSURFACE MATERIAL OFFERS DIFFUSE, SPECULAR, ETC. HOW DO THESE WORK TOGETHER? on details.

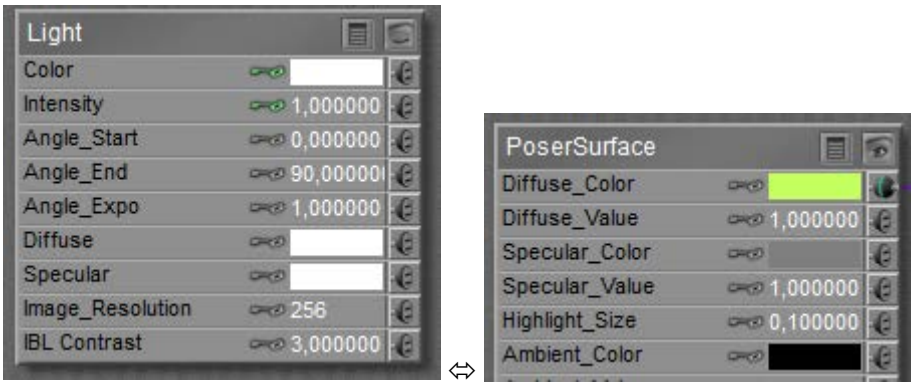
Highlight_Size, available in Simple as well as in Advanced interface, is a straightforward parameter: small values make small, intense highlights



representing hard, smooth, shiny surfaces while larger values make larger, blurred highlights instead. Like for Diffuse, Poser offers a Specular node (Advanced interface only) which offers features similar to the Specular component in the PoserSurface. See 43 WHAT'S THE DIFFERENCE BETWEEN THE SPECULAR COMPONENT, AND THE SPECULAR NODE? on details about this node, and 66 HOW CAN I ASSIGN AN IMAGE TO A MATERIAL? or 68 HOW CAN I ASSIGN A MOVIE TO A MATERIAL? on assigning either an image or a movie respectively to the Specular slots.

Making Highlights

As said, Poser discriminates between Reflecting (diffuse) light from objects, and Specular light from direct light sources in the scene. To do so, direct lights in Poser offer a diffuse as well as an independent specular component. The first component will trigger responses from the Diffuse part in the material definition (and similar parts as well) while the second component of the direct light will trigger responses from the Specular part in the definition.



The Diffuse and Specular components in the Light relate to the Diffuse and Specular components in the Material definition of an object.

So, in order to get highlights, I need this Specular component in the direct lights in the scene to be available. But I can black out the diffuse component, turning the light into a Specular-only one. See 94 CAN I USE A LIGHT FOR DIFFUSE OR SPECULAR LIGHTING ONLY? for details on this.

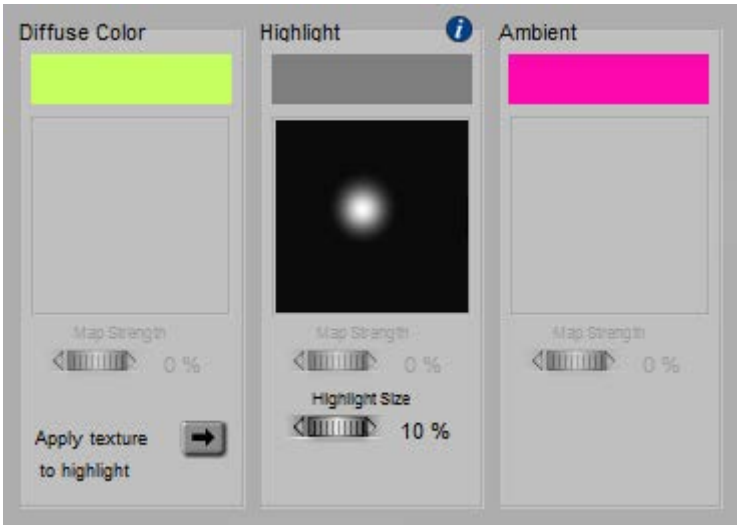
The shape of Highlights

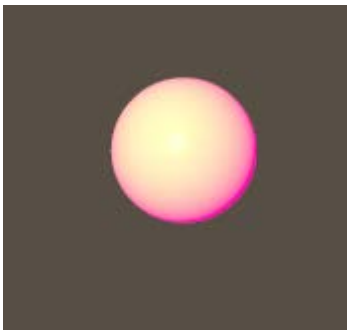
Although Poser is not raytracing to derive the specular highlights, the blurred highlights are still decently described by the "angle in equals angle out" concept as is the case with mirrors. This implies that the outgoing light intensity is highest in a direction determined by the angle of the surface with the incoming light. Around that "angle out" range the intensity drops - depending on the highlight size. Unfortunately, and though it calculates and renders pretty fast, the standard highlighting method produces quite unrealistic results and makes everything look like a bad representation of hard plastic.

This is why Poser offers additional features via the Advanced interface, especially alternatives for the Specular node, like Blinn (44 WHAT'S THE BLINN NODE USED FOR?), or Anisotropic for microscopically grooved surfaces (46 WHAT'S THE ANISOTROPIC NODE USED FOR?).

07 What's the Ambient component intended for?

For short: ambient make objects glow, and emit light independent of any lighting sources in the scene. This is to represent led-lights, or some phosphorescence of fluorescence effects.



=>


Note that light, and hence ambient glow as well, serves the visibility and presence of objects in the scene while shadows serve shape and surface structure details of an object. So, when using ambient glow, I will gain object visibility but I will lose some of the shape and surface details, and hence I'll "flatten" the image. In other words, I should not overdo.

Intermediate

In Poser, Ambient (both Simple and Advanced interface) as well as its equivalent Translucence (Advanced interface only, see 39 WHAT'S THE TRANSLUCENCE COMPONENT INTENDED FOR? on details) will make an object glow. There are various reasons to have an Ambient component in the materials definition.

First, the truly legitimate reason, is to implement the objects glow as it appears in real life. Led-lights, or some phosphorescence of fluorescence effects, were already mentioned above.

Second, considered a half-legitimate reason, is to represent translucency as can be found in skin, wax candles and alike, in a render setup which is poor on raytracing and hardly supports sub-surface scattering. Early Poser versions, PC's with less computing power, or the requirement for a massive output stream as in animations might be the reason for this. A resource intensive process is then replaced by a much faster process with – sometimes slightly – less realistic results.

Third, nowadays considered almost an illegitimate reason, is to compensate the object for poor lighting conditions at the spot. Adding some local point- or spotlights, and especially switching to indirect lighting (IDL) conditions supported by modern Poser versions, resolve the lighting issue the way it should be.

Some additional notes:

- In contrast to Diffuse and Specular, there is no angular distribution of the light send towards the camera. Edges are as intense as areas in the middle part of the object, which makes it harder to distinguish shape details. This is the way nature works too, by approximation.
- See 24 A POSERSURFACE MATERIAL COMPONENT, OFFERS COLOR AND VALUE. HOW DO THESE WORK TOGETHER? on plugging Material Room nodes into the Ambient (and/or Translucency) slots of a PoserSurface material definition.

- Ambient (and/or Translucency) generate some local lighting levels which are independent of the intensities of external sources of light. Hence, when I decide to dim the lighting levels in my scene to avoid overlighting, I've got to dim those Ambient aspects of the materials as well to keep the balance.
- Ambient (and/or Translucency) are proper candidates for turning an object into a light source under indirect lighting (IDL) conditions. This is the way to make an all surrounding dome object lighting the entre scene. I may have to pump up the intensities though.

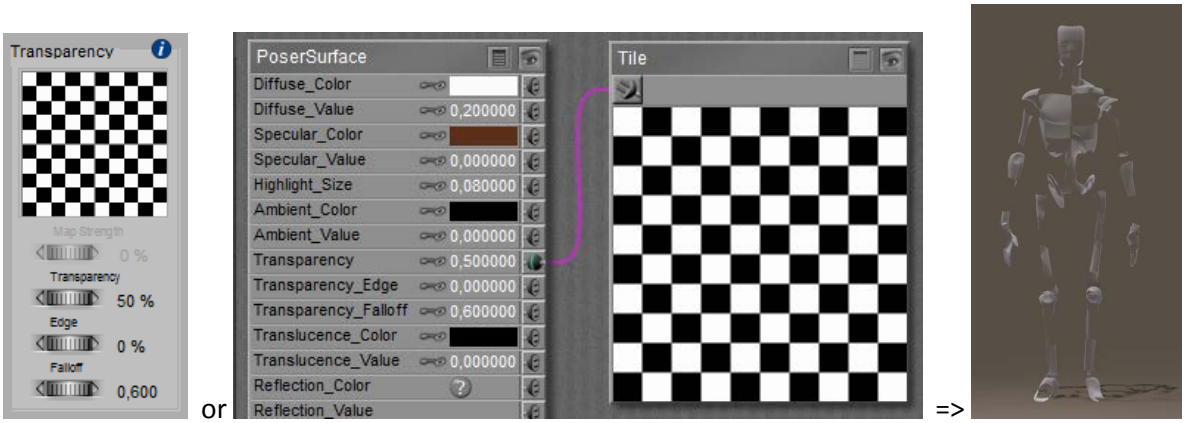
Recommendations:

- To prevent artefacts in rendering when applying the Gamma mechanism (recommended, available in PoserPro and Poser 10 and up, see 98 WHAT'S THE FUZZ ABOUT GAMMA CORRECTION? for details) the Value setting should be kept to 1.0 only (or 0.0 but no intermediate values).
- To prevent overlighting when combined with other aspects of lighting and material definitions, it's recommended not to exceed a (100% - (Diffuse+Specular)) brightness in the Color-swatch and maps (or in the Value setting when not applying Gamma). So with Diffuse set to 70% and Specularity set to 20%, Ambient should be limited to 10% at most.
  See 20 A POSERSURFACE MATERIAL OFFERS DIFFUSE, SPECULAR, ETC. HOW DO THESE WORK TOGETHER? on details.

08 What's the Transparency component intended for?

Although I'll grasp transparency from my own experience, and although most characteristics of it are available through the Simple interface into Material Room, its implementation in Poser is not the easiest to comprehend. That turns the presentation of this topic into a mixture of Simple, and Intermediate elements, plus some Advanced issues too. Sorry for that.

Formally stated, transparency is the ability of a surface to let light to pass through. Transparency is a number, 0-100%. When that number has different values at various spots on the subject, a (greyscale) image can be used. Note that – in Poser – the image is used **in an inverted way**: 100% or **black** means: fully transparent, while 0% or **white** means: fully opaque.



See 04 MATERIAL ROOM SIMPLE AND ADVANCED INTERFACE - HOW DO THESE RELATE? on both interfaces.

The behavior of Transparency can also be stated in another way:

- When it's just a value, then that value represents transparency (100% is fully transparent)
- When an image is attached to it (or any nodes are attached to it in Advanced interface), its meaning flips and the value is a multiplier for the resulting image_map which represents opaqueness instead: 1 (white) is fully opaque and 0 (black) is fully transparent.

More in detail

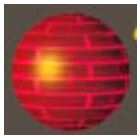
There are three different reasons surfaces can let light pass through:

- Because of lacking material. This is a **lace-like transparency**, usually supported by an image (the transparency map) representing the pattern of this materials presence or absence. This is the way to drill holes in a wall or in a jacket, and let the background (or an underlying shirt) shine through. White means presence of material and black means absence.
  For very fine – and hence almost invisible – patterns (e.g. nylon stockings), a uniform grayscale or just a transparency value can be used instead.
- Because a perfectly clear object is covered with a thin layer which filters all the light upon it. This is what's meant by setting an intermediate value for transparency, or using an image (greyscale) map instead. Like the lace-like variant just mentioned, this **thin-layer transparency** is a pure surface effect. And in Poser, all object materials are surface properties.
- Because the material filters and colors the light passing through, as is the case in liquids and glass; it's a **glass-like transparency**. In this case, using a transparency value is the common way to represent the amount of light which can pass through.
  Any colors from an image map will be ignored, and the (volumetric) effect has to be translated to a surface effect. For instance, a surface transparency set to 90% will represent an object transparency of 90% x 90% = 81% regardless of local thickness and shape, as the

light passes through the front and back surfaces.
Poser can't handle thickness and shape but presents some tricks for escape, see Edges & Bends below. When I want to color (filter) the light passing through I've got to use the Refraction_Color or the Alternate_Diffuse component, which is available through the Advanced interface only (see 38 WHAT'S THE REFRACTION COMPONENT INTENDED FOR? for details). Then things get complicated.

Note:

When the material has a diffuse (surface coloring) component too, then this will show as far as the material is non-transparent. The diffuse material will not show in the holes in the surface, as driven by the transparency-map. Do note that the more transparent a surface becomes, the less it gets colored as the color is a property of the non-transparent portion of the surface.
The specular component however is not affected at all by the transparency settings, so transparent areas on the surface can be as shiny as less (or non-)transparent ones. This is good for thin-layer and glass-like transparency. But for lace-like transparent surfaces, the holes should not be shiny at all, as there is no material in place to reflect the lights. Hence, for such material the transparency patterns should be copied to the specular settings.

 glass-like (highlights overall) ⇔  lace-like (highlights on substance only)

The same holds for Ambient, Bump, Displacement, and Reflection: when there is no material then there can't be a surface response to light. But I have to tell Poser myself, as Poser does not distinguish the various interpretations of transparency (lace-like vs layer/glass-like).
When using the Advanced interface: Transparency also does affect Alternate_Diffuse but does not affect Translucence, Refraction or Alternate_Specular.

Edges and Bends

When I look at, or through, a (semi) transparent object with various curves and bends, I will note that the frontal view differs from the sides. For hollow objects, light rays passing the front and back might meet less material underway compared with rays traveling through the sides. For solid objects this might be the opposite.
On top of that, various causes for light-bends (called Refraction, see 38 WHAT'S THE REFRACTION COMPONENT INTENDED FOR?), absorption and scattering will reduce the transparency of objects at the edges.
Next to all this, in real life transparency depends on the angle towards the camera: perpendicular angles have a much higher transparency than skew ones; for the latter, light rays just bounce off, and transparency decreases while reflectivity goes up. This advanced effect (known as Fresnel) is supported by recent versions of Poser, at the cost of serious prolongation of render times and increased memory requirements.

So when one wants an improvement over just transparency, without requiring the high level of photorealism that comes with Refraction (38 WHAT'S THE REFRACTION COMPONENT INTENDED FOR?), Poser offers a way to cheat to compensate for its inability to handle any kind of volumetric effects:

- **Transparency_Edge** defines the transparency at the edge, while
- **Transparency_Falloff** defines the thickness of the edge.

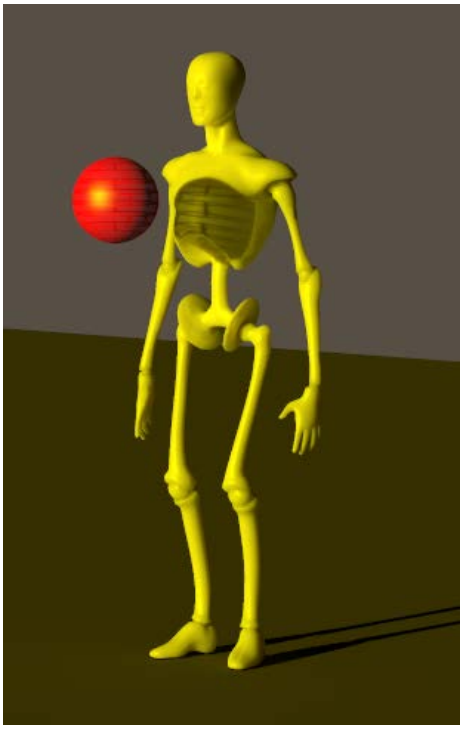
As Transparency 1.0 means fully transparent, the Edge should have a (much) lower value to mimic the mentioned effects in a believable way. The default is 0.0, completely opaque at the edges. When I do it the other way around however, and set the edges to a high value like 1.0 (combined with a low value for transparency), the edge will fade away like I'm presenting a gaseous, cloudlike object.

Small values for Falloff make a fast transition from edge to regular surface, and hence make a thin "surface" layer for the object. This surface thickness seems proportional to the size of the object; the Falloff is a percentage of something, it's not an absolute value. And although results can become quite unnatural, high values like 10 and up are possible and allowed.

Note:

Although Edge and Falloff make some sort of skin or surface layer for a transparent object, there is no sharp boundary at the inside. As a result, these parameters cannot make a hollow object.

Raising Awareness



One image says it all: in Poser, yellow light passing through a **red** patterned-transparent ball will cast a patterned **black** shadow onto the object behind it, mixed with the yellow of the light. In nature, such a shadow would be red (as the passing light gets filtered).

Transparency won't handle color at all, but it does make the pattern in the shadow.
Diffuse will filter the light shining onto the ball and bouncing towards the camera.
Refraction will sort filter the light passing through the ball from behind towards the camera. It's objects only, and does not handle direct light at all but produces dark shadows instead.

Neither transparency, nor a diffuse or a refractive color will color-filter the lightrays passing through the ball as such, and no reddening of the shadowed area will occur in any way.

Poser Firefly does have its limitations, and this is one of them.

Advanced

Transparency and the effect of Viewing and Lighting angles

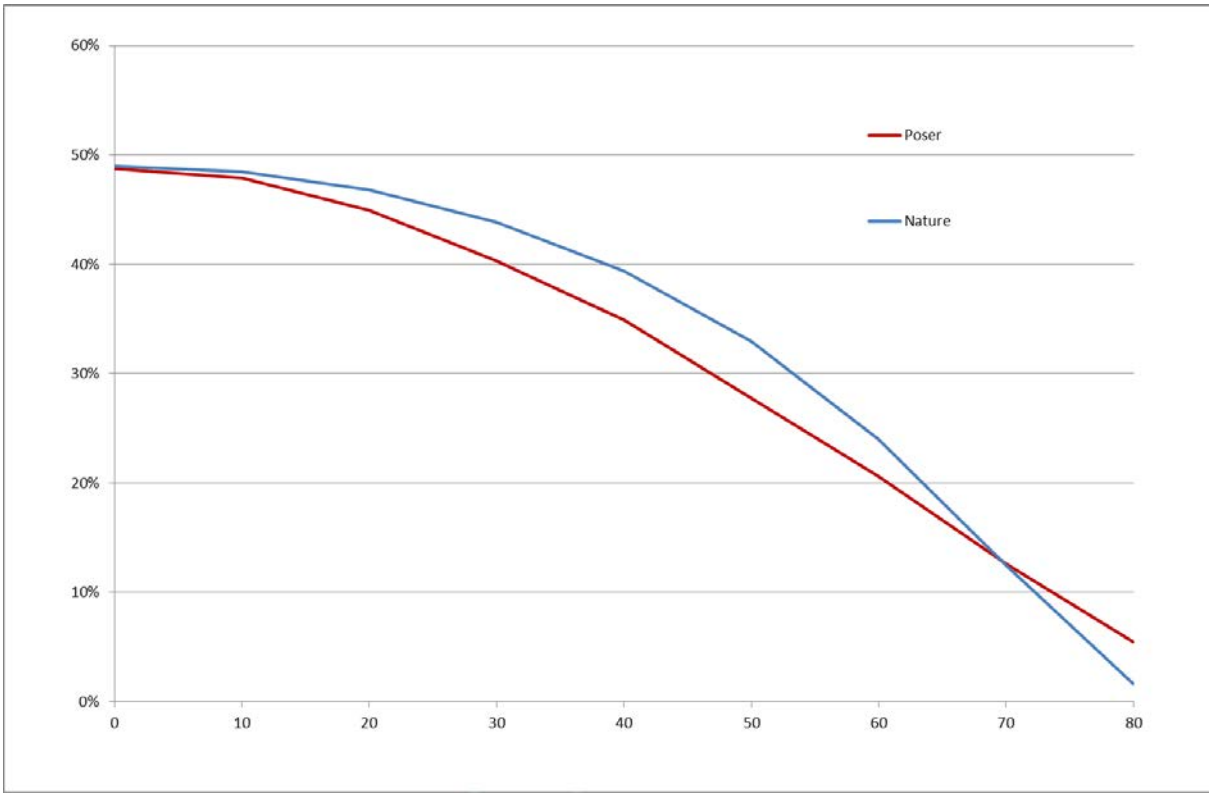
Poser does not do anything with material properties or object thickness; it does not deal with object (volumetric) transparency but uses a surface transparency value instead. Poser behaves like a semi-transparent foil is wrapped around the object, and while the light passes through the object, the foil is passed twice. So, when the PoserSurface transparency is set to 70%, the object transparency equals about 50%, and the render will match nature's behavior.

When the light passes under an angle, nature will reduce the transparency of the object because the path within the object (or through the foil, which does seem have some thickness anyway) becomes longer, and more light gets absorbed along the way.

> *The math:*
> *When – in real life – light passes a block or foil of semi-transparent material, with a thickness w (in meters or alike) and under an angle a, then a portion of that light will be taken out, and the remaining portion – known as object transparency – is exp{ -k*w/cos(a) } with k being a*

constant describing the material itself. Low k values, close to zero, make high transparency and large values make low transparency. The same way thick objects (large w value) are less transparent than thin ones. But as Poser does not support material properties or thicknesses, all I can do is set a surface transparency T in PoserSurface. Then the final object transparency will be $T^{2/\cos(a)}$.

Poser does support this angular behavior, but not exactly. I guess because that would require too much calculation during the render. But as the following graph shows, it does a nice job:



This is for a block with 70% surface transparency. Light which comes straight through will be dimmed to 70% x 70% = 49% object transparency. When it passes at a 30° angle Poser (red line) will dim it to 40% while in nature (blue line) it will be 44%, and at 60° Poser will dim to 20% while nature will reveal 23%.

Conclusion:
For an application which is not doing any volumetrics, and is not raytracing except when assigned to do so for specific surfaces (reflection, refraction), Poser does quite a nice attempt to stay close to "the real thing" in an efficient and effective way. Transparency is a surface property, as if a foil surrounds the object at hand, but that foil is supposed to have some finite thickness and Poser supports the angular dependency which comes from that.
At the detailed level, it does show differences which cannot be eliminated any further. Poser does have its limitations, and this is one of them.

Transparency and the effect of Gamma Correction

Consider an object A blocked by another object B, that is: B will cast some shadows onto A. For various reasons those shadows tend to look overly dark. For instance, because in real life light scatters around in the air thanks to tiny dust particles floating around, and because ceilings, walls and objects in the scene scatter light around as well. Poser offers a correction mechanism for that: Gamma Correction (aka GC), as discussed in 98 WHAT'S THE FUZZ ABOUT GAMMA CORRECTION?, present in

Poser 10 and Pro 2010, and up. GC will brighten up shadows and will dim highlights to give the render a more pleasant and more realistic look. So far so good.

Now, consider object B to be partially transparent, say 90% per surface. Light passing from its front to back will be reduced to 90% x 90% = 81% and object A will just receive less light instead of no light at all in the deep shadow. That will be catered properly by GC again. Actually, GC will brighten up the dimmed area back to about 90%, and keep it in sync with all other shadows and shades in the scene. But any light from A, through B towards the camera will be dimmed to 81% as well as being filtered by B's transparency too. And as Poser cannot tell the difference from all those dimming effects, shadowing or transparency or anything, this transparency effect is brightened up to about 90% as well. As a result, the object B looks far more transparent than it's designed to be, thanks to GC.

Now what should I do? Should I let Transparency bypass the Gamma mechanism resulting in proper shadows and a slightly too bright object, or should I let Transparency participate in the Gamma mechanism, with probably a more accurate object but too dark shadows, which can be brightened up in other ways? This is discussed in detail in 99 GAMMA, TRANSPARENCY AND BLENDS, but for short: I should do the first. Transparency should bypass Gamma like Bump and Displacement do. The object will be a bit off then, but when I act the other way, it will be off at least twice as much.

09 What's the Reflection component intended for?

Well, I know about reflections in nature, from mirrors, metal balls, glass, and water surfaces. So the answer can't be that difficult.

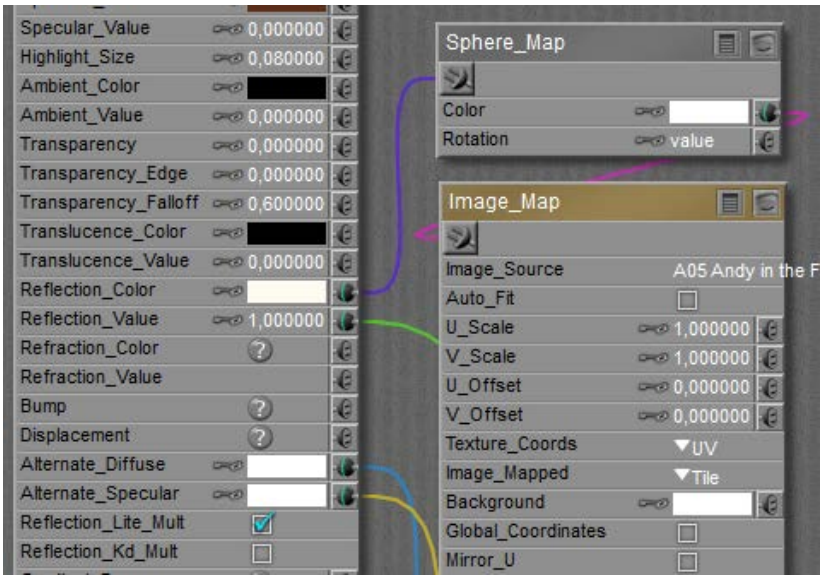
Unfortunately, it can be quite complicated in a 3D program, because the proper handling of reflections requires a full fletched use of raytracing, with an infinite number of bounces. In the old days this required so much computing power that rendering took about till infinity. From this recent past, which is just one to two decades away, three approaches remain which are discussed below:

- Reflection maps
- Specularity
- Raytracing

Although most characteristics of these are available through the Simple interface into Material Room, the implementations in Poser are not always the easiest to comprehend. That turns the presentation of this topic into a mixture of Simple and Intermediate elements. Sorry for that.

Reflection Maps

First, the reflections can be faked by assigning just an image to it. For quite blurred reflections and an image which matches the surrounding scene quite well, this might work and will speed up rendering tremendously, which is a serious advantage in lengthy animations. It's used often on chrome car accessories.

or


Note however that the brightness of such reflections is autonomous, independent of the surrounding lighting levels. So when the lights dim, or the object resides in a shadowy area, the intensity of this material property should be adjusted accordingly. Poser supports this, by multiplying the reflections by the brightness (and color) of the (diffuse) light on the surface of the object. The **Reflection Lite Mult** option (or **Multiply with Lights** in the Simple interface) caters for this, and is ON by default.

So, when I animate a car driving through a sunny lane with shadowy trees, and the shiny chrome wheels are made pseudo-reflective by using just images, then the reflectivity is dimmed when a wheel passes through a shadow.

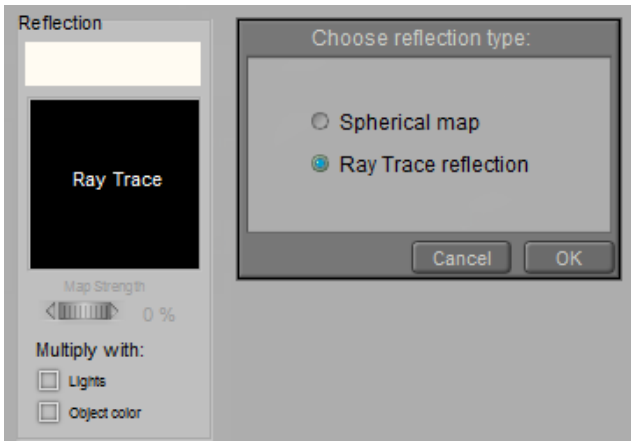
The option should be OFF however when actual raytracing (see 'third' below) is used to handle the reflections because then the amount of reflected light is properly derived from the surroundings themselves.

Specularity

Second, specularity was introduced to create highlights (see 06 WHAT'S THE SPECULAR / HIGHLIGHT COMPONENT INTENDED FOR?), representing blurred reflections from direct light sources only. In Poser, an active specular channel of a direct spot-, point- or infinite light in combination with an active specular part in the objects material, are both required for getting such highlights.
Note that indirect (IDL) lighting schemes as well as image based (IBL) lights deal with diffuse light only, and will not produce any highlights. See 94 CAN I USE A LIGHT FOR DIFFUSE OR SPECULAR LIGHTING ONLY? on this.

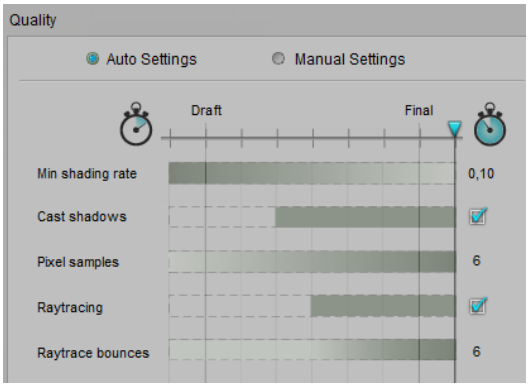
Raytracing

Third, real raytracing can be implemented by plugging the raytrace node into the PoserSurface definition. This is directly available in the Advanced interface only, see 59 WHAT'S THE RAYTRACE > REFLECT NODE USED FOR? on details. In the Simple interface, just choose the Raytrace option:

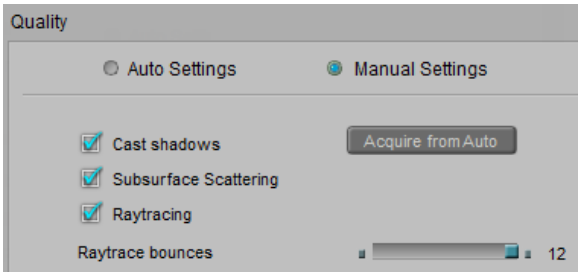


Note that in his case, those 'real reflections' will only consider the objects in the scene, and not the direct light sources around. To get the shine of lamps in the render, I either have to include some additional glowing objects in the scene, or I have to use specularity for this purpose.
Also note, as said above, that the **Reflection Lite Mult** option (or: **Multiply with Lights** in the Simple interface) should be OFF for raytraced reflections as it serves the brightness adjustments of image based reflections (reflection maps). Raytracing makes the adjustments itself automatically, and having the option ON will then enforce the adjustments twice.

Plus… raytracing has to be enabled in the Render Settings, and the (maximum) amount of raytrace bounces has to be set high enough. That is: the highest value is the best as nature does not have any maximum at all, but some people report that lower values speed up their renders without a recognizable loss of quality. See the notes on Transparency below.



or



Poser offers **Reflection Color**, and in the Advanced interface **Reflection Value** as well. See

- 31 WHAT'S A PROPER COLOR FOR REFLECTIONS? on the color of reflections, and the Multiply with Object Color (Kd_Mult) option
- 32 WHAT'S A PROPER VALUE FOR REFLECTIONS? on the value of reflections
- 34 DO RENDER SETTINGS AFFECT THE BEHAVIOR OF MATERIALS? on Raytrace Bounces

Also see 24 A POSERSURFACE MATERIAL COMPONENT, OFFERS COLOR AND VALUE. HOW DO THESE WORK TOGETHER? on plugging Material Room nodes into the Reflection (and Refraction) slots of a PoserSurface material definition.

Advanced

For short: when I use GC for my renders, I leave the Reflection_Value at 1.0 and adjust the color swatch instead. This is discussed in 98 WHAT'S THE FUZZ ABOUT GAMMA CORRECTION?. And as discussed below, when I use transparency next to reflection, I use Alternate_Diffuse instead of the Reflection component. This happens to be a color swatch as well and so caters for the GC adjustment too. And when I'm not using GC or transparency, there are no losses in using Alternate_Diffuse anyway. So that is the preferred way of work.

Combining (raytraced) Reflections with Transparency

Most objects which are (semi)transparent, are reflective as well, at least to some extent. When I combine (raytraced) reflections with transparency in one PoserSurface, I have to take notice of various things.

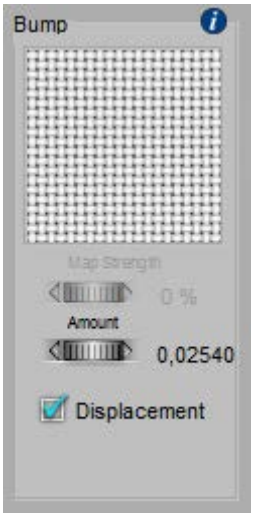
- The combination of raytraced reflections and transparency will generate internal reflections within the object. Poser does a reasonable (*) job handling these, but as the amount of ray bounces is about infinite, render times will go up immensely, and any reduction of Raytrace Bounces in the Render Settings might reduce the quality of the result.
- Light which gets reflected cannot pass the surface, so for reflectivity R (say 80%) the transparency should not exceed 1-R (in this case: 20%) or vice versa. Otherwise, I'll introduce some magical source of extra light, and things will start looking overly bright.
  In any case, I should not combine any reflectivity with 100% transparency, or 100% reflectivity with any transparency. Such a surface cannot exist.
- In the Simple interface, and in the Advanced interface with an image_map or Reflect node plugged into the Reflection component, I have to balance transparency and reflectivity manually. Transparency does not reduce the Reflection component as it does with Diffuse. However, Transparency does work on the (Alternate_)Diffuse component so when the Reflect node is plugged into Alternate_Diffuse, the reflection is reduced automatically in a correct proportion.
  That is: with a 90% surface transparency set, and reflectivity set to 80%, the final reflection will be reduced to 1-90% = 10% of that, making 8% net surface reflectivity. This then is boosted up due to internal reflections.
- (*) due to internal reflections, the final reflectivity goes up to $R*[1+T^2/(1-R^2)]$ for a surface reflectivity set to R and a surface transparency set to T. This is what Poser does, as it treats transparency as a surface property. With R=80%, T=20% we'll get 89% final reflectivity.
  For a semi-transparent solid object however, the final reflectivity should go up to $R*[\ 1+T^2/(1-(RT)^2)\ ]$ which results in 83% with the same numbers filled in. This refers to only one-third of the earlier derived additional brightening from internal reflections.
  In the first (surface) model, the light bouncing around within the object is only absorbed when it passes the surface, leaving the object. In the latter (solid object) model, the light becomes absorbed while bouncing around in the object. Therefore, in the latter case internal reflections contribute less to the final result. Poser however does not support the volumetric approach, and so, when its surface transparency actually represents a volumetric effect, the internal reflections are (far) too strong. I can reduce the surface transparency a bit, for compensation.

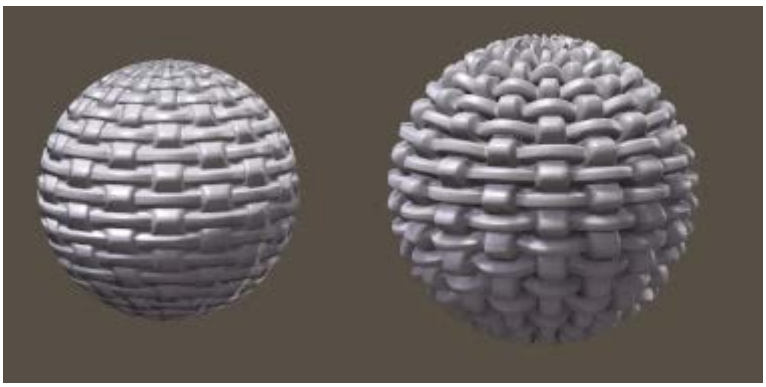
10 What are the Bump and Displacement components intended for?

Each object surface – except for the hard and smooth ones – has some irregularities, like pores in the skin. On a large distance, these details don't show in the render at all, or can be embedded as shady areas in the color texture at most. At a somewhat shorter distance, but still away from the camera, those regularities can be faked, saving a lot of render time and memory resources. That is what Bump is for. The magic is done by the texture (Bump-map) alone; no surface elements are actually moved.

When the object surface comes closer to the camera, surface elements actually do have to move a bit in order to make believable results. Those displacements still can be image driven, which is where the Displacement-map kicks in. This way of work actually saves a lot of detailed modeling. For instance, terrains are generated this way.
But again, its effects are limited, so when the object surface comes really close to the camera, it pays off to actually adjust the object mesh actually, and to model in the details themselves.

So, the other way around: I can either put the details into the object mesh which takes a lot of my modeling time, or I can apply a displacement map which requires memory and render time, or I can apply a bump map faking the displacements, or I can leave it the way it is. The best choice depends on the distance of the object to the camera. Additional considerations on this are required in animations, where this distance can vary over time.
In general, a serious difference between Bump and Displacement is that Bump does not distort the object itself, and hence leaves the edges intact. So a bumped ball still will be a perfectly round object in the render. A displaced ball will not.



makes 

And notice the difference between Bump (left) and Displacement (right). Not only the bumped one still is a perfectly round object while the displaced one has got a roughed up edge, the bumped one also is smaller, for the same amount used. This is why:

- In Bump, white gives maximum effect outward, while black gives maximum effect inward, so 50% grey is the neutral / no effect value. The Amount indicates the total difference between black and white, so the effect is half of the amount outward, and half the amount inward. Using a negative amount will reverse the effect: black will simulate an outward effect, while white will dent inward.
  And… bump is not modifying the object mesh, and hence cannot alter its size.
- In Displacement, white gives full outward growth of the object while black gives no growth at all. So, for displacement, black is the neutral / no effect value, and any displacement will

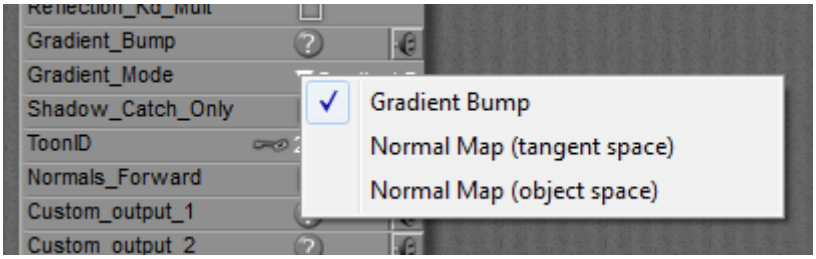
grow the ball. When displacements should be made inward, then negative amounts must be used.
Hence, displacement will actually grow the ball, and its visual effect is twice as strong as for bump.

Notes:

- Via the Simple interface, using the **Displacement** option one can choose between Bump and Displacement, it's either the one or the other. Via the Advanced interface, one can have both. For instance to drive large adjustment by a displacement, and small details by a bump.
- The **Amount** is measured in the unit I've set in my Global Preferences.
  The screen grab above was made on a system using Meters, so 0.0254 indicates 2.54 cm. Which by the way is about 10% of the balls diameter; quite a lot. When the units are changed to inches, the value will change to 1.0. I have to take this into account when using data from examples. (And… python scripts have to use inches by all means).
  This 1" value is the default, which actually is quite a lot for say skin details, but quite small for outdoor terrain surfaces.

Intermediate

The PoserSurface material definition offers a few additional options for compatibility:



- Gradient_Mode = Gradient Bump offers compatibility with old Poser versions, I can plug the existing *.bum file into the Gradient_Bump slot, as an image
- Gradient_Mode = Normal Map offers compatibility with Game Characters. If my figure comes with normal maps instead of bump/displacement maps, I can select the correct type, and plug the map into the Gradient_Bump slot as an image.

Important note: in Poser Pro and Poser 10 and up, when using image maps for Bump and/or Displacement, these should have the Gamma set to 1.0 explicitly. Otherwise, the images will get dimmed considerably before application, especially in the mid-grey areas. This will result in dimming the Bumps / Displacements themselves as well, which is wrong.

In case you wonder: there are no chapters 11 – 19. Those are reserved for later use.

III Intermediate Surface Definitions

The articles in this section discuss some material definitions for object surfaces (the PoserSurface), which are handled through the Advanced Interface: the nodes from the Lighting group, and the nodes on image-maps and movies. It also discusses some principles on dealing with the PoserSurface root node.

- First, articles discuss the workings of the PoserSurface root node in general.
  - How do the various components and elements work together
    - 20 A PoserSurface material offers Diffuse, Specular, etc. How do these work together?
    - 24 A PoserSurface material component, offers Color and Value. How do these work together?
  - Principal and practical limitations in combining components
    - 21 Are there any principal limitations in combining Diffuse, Specular, … in a single material?
    - 23 Are there any practical limitations in combining Diffuse, Specular, … in a single material?
  - A brief intro on node-tree building
    - 25 Can I get a brief intro on Node-tree building in Material Room?
- Second, articles discuss some elaborate details of components available in the Simple interface: diffuse (Lambert) shading, reflection details, render settings.
    - 30 Can I get a (simplified) explanation on Lambert and diffuse shading?
    - 31 What's a proper Color for Reflections?
    - 34 Do Render settings affect the behavior of materials?
- Third, articles discuss the additional components of the PoserSurface node, like Shadow Catch, ToonID, Custom output, refraction and translucency.
    - 35 What's the Shadow Catch Only intended for?
    - 36 What's the ToonID intended for?
    - 37 What are the Custom_outputs intended for?
    - 38 What's the Refraction component intended for?
    - 39 What's the Translucence component intended for?
- Forth, from
    - 40 What are the Alternate_Diffuse and Alternate_Specular components intended for?

  on, articles discuss Alternate_Diffuse, Alternate_Specular and all the nodes from the Lighting > Diffuse and Lighting > Specular groups. The commonly used ones:
    - 41 What's the difference between the Diffuse component, and the Diffuse node?
    - 42 What's the Clay node used for?
    - 43 What's the difference between the Specular component, and the Specular node?
    - 44 What's the Blinn node used for?
    - 46 What's the Anisotropic node used for?

  And the less commonly used ones:
    - 47 What's the Diffuse > probeLight node used for?
    - 48 What's the Diffuse > Toon node used for?
    - 45 What's the Specular > Glossy node used for?
    - 49 What's the Specular > Phong node used for?
    - 50 What's the Specular > Ks Microfacet node used for?
- Fifth, articles discuss the nodes from the Lighting > Special group, like Scatter and Hair.

- Skin, Velvet, Hair and Fastscatter
  - 51 What's the Special > Skin node used for?
  - 53 What's the Special > Velvet node used for?
  - 54 What's the Special > Hair node used for?
  - 57 What's the Special > FastScatter node used for?
- Subsurface Skin, Scatter and CustomScatter
  - 52 What's the Special > Subsurface Skin node used for?
  - 55 What's the Special >Scatter node used for?
  - 56 What's the Special >CustomScatter node used for?

In 58 What's the difference between all those Scatter nodes? the various scatter nodes are compared.

- Sixth, articles discuss the nodes from the Lighting > Raytrace group.
  - Reflect, Refract and Fresnel
    - 59 What's the Raytrace > Reflect node used for?
    - 60 What's the Raytrace > Refract node used for?
    - 63 What's the Raytrace > Fresnel node used for?

      In 64 How do I properly combine Transparency and Raytracing? their complex relationship with Transparency is dealt with.
  - Ambient Occlusion and Gather
    - 61 What's the Raytrace > Ambient Occlusion node used for?
    - 62 What's the Raytrace > Gather node used for?
- Seventh, articles discuss spherical mapping, image maps and movie-based textures.
  - 65 What's the Environment Map > Spherical Map node used for?
  - 66 How can I assign an image to a material? and 67 How does Poser handle my images for texturing?
  - 68 How can I assign a movie to a material? and 69 How does Poser handle my movies for texturing?

Next sections present similar articles on defining the surface properties of objects Advanced level (IV Advanced Surface Definitions) as well as on atmosphere, background, lights (V Materials for Non-Objects) as far as these are handled through the Material Room interface.
The Appendix: Poser nodes / settings and versions lists all Material Room nodes and relevant Render Settings, and their availability in the various Poser versions.

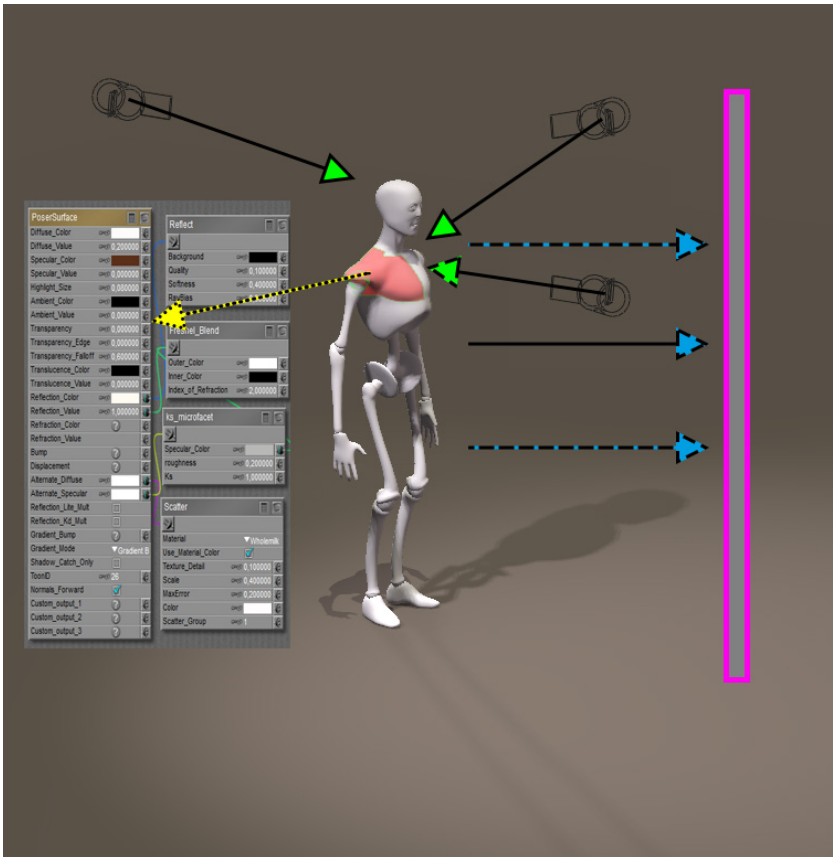
20 A PoserSurface material offers Diffuse, Specular, etc. How do these work together?

Simply stated, all elements of a surface/light interaction are added up mathematically, on a per light, per channel (like Diffuse, Specular, Ambient, …), color-by-color (Red to Red, Green to Green, Blue to Blue) basis. This may cause overlighting, as components might add up to values over 100%. This can be prevented by reducing the intensities of lights and/or by reducing the brightnesses of the various surface components.

A more detailed answer following below requires determining all the steps from scene to render, which is something for the experienced and curious users, who want to have a better understanding of the way things work.

Intermediate

Say, a portion of the render result is determined by a portion of an object surface (see 74 MATERIAL ZONES, OR: TO WHICH BODY PARTS DO THE MATERIALS APPLY TO?) which is covered by a single material. That material offers various "channels" (like Diffuse, Specular, Ambient, …) and the surface is lit by various light sources in the scene.



Object mesh (white) consists of various material-zones (red), each catching light rays (green) and using its own surface definition (yellow) to calculate the re-emitted light (blue) towards the camera / render plane (pink).

First, when my Poser version supports gamma correction (GC) and I have enabled it, each channel and light will be adjusted in a way that brightnesses are (or may be) reduced (more precise: pushed away from 100%). When my Poser version does not support GC or I've disabled it, this step will be skipped. You might like to read more about CG in 98 WHAT'S THE FUZZ ABOUT GAMMA CORRECTION?.

Second, for each channel / light combination a render subresult is derived. This will introduce reflections, shadows, highlights, glow and so on, on a per-light basis.
By the way: Poser, and especially Poser Pro, offers various scripts and features to export those subresults individually for manual re-combination in postproduction.

Third, all those render sub results are added up, per light, per channel, on a color-by-color basis. So a diffuse result RGB (60%,50%,10%) and a specular result RGB (80%, 60%,15%) on the same spot make RGB (140%, 110%, 25%). Plain simple arithmetic.

Forth, when my Poser version supports gamma correction and I have enabled it, the final result will be adjusted in a way opposite to the first step: brightnesses are increased (more precise: pulled towards 100%).
When my Poser version does not support GC or I've disabled it, this step will be skipped, as was the first step in the process.

Fifth, but only for showing the resulting image on screen and for exporting it to a standard 8-bit-per-color format (JPG, PNG, …): all colors are clipped to 100% max, which for instance might turn the result (140%, 110%, 25%) into (100%,100%,25%).
Internally, and in some export formats as well (HDR, EXR for Poser Pro) the software works in a 16-bit per color format.

A brief elaboration on Poser Gamma Correction(GC)

As a result of the third step, two identical lights will give the same result as a single light with double intensity, that is: when GC is not enabled. The arithmetic then is simple and linear. This is different from adding light effects in nature, like projecting an image on a wall via multiple projectors. In that case, my eyes will adjust to the increased lighting level. But images can't. Photoshop offers the Screen layers mode for handling this, which is considered the industry standard for handling the addition of light/surface subresults as it's very close to the behavior of good old film, and its modern electronic equivalent.

> Side note:
> *Screening is defined as "re-inverse the multiplications of the inverses" which for images a and b comes down to: result = 1- (1-a)*(1-b).*
> *Applying this to the colors mentioned above, this will result in*
> *1- (1- (60%,50%,10%))*(1- (80%, 60%,15%)) = (92%, 80%, 23.5%)*

When applying Poser GC to the colors mentioned above, the result will be RGB (97%, 76%, 11%). This does not need clipping and hence will not present any overlighting artefacts. This is one of the big reasons to enable GC when available. Next to that, the Poser result under GC is somewhat similar to the industry standard. And on top of that, GC will soften shadows and shading as if the scene is filled with air and as if other objects are bouncing light around, as in real life.
When applying GC it's much recommended to keep all Values at 1.0 (or 0.0) and to define all brightnesses through the Color (swatch). This is because Poser's GC mechanism treats colors and values differently, so a light with a 50% white (mid gray) and 100% intensity works as required but a light with 100% white and 50% intensity will produce a different (far brighter) result.
Bump, Displacement and Transparency should be excluded from the Gamma mechanism. As said, you might like to read more about CG in 98 WHAT'S THE FUZZ ABOUT GAMMA CORRECTION? and 99 GAMMA, TRANSPARENCY AND BLENDS.

21 Are there any principal limitations in combining Diffuse, Specular, … in a single material?

The short answer is NOT REALLY, and be advised that there are a lot of opinions in the forums and around on the Internet. Most of those opinions arise from the behavior and limitations on materials in nature, when a single source of light hits a surface and triggers all components. That light can be reflected, diffused, transmitted and so on, and hence these portions should not add up to a value larger than one (100%), otherwise the object re-radiates more light than it receives. It could be less, to represent absorption in the material.

However, the various portions of a Poser material respond to quite different – and quite independent – sources of light. Diffuse responds to direct and indirect diffuse lighting, Specular responds to direct specular lighting only, Reflection and Refraction responds to any (indirect) light re-emitted by any other objects in the scene – but not direct light sources themselves, and Ambient / Translucency represent some lighting or glow emitted by the object surface itself. See 22 ARE THERE MORE PRINCIPAL LIMITATIONS IN COMBINING DIFFUSE, SPECULAR, … IN A SINGLE MATERIAL? for more details and background on this subject.

This brings up a different issue. In nature, adding up multiple light sources will increase the brightness of the surface, and my eyes will adapt to that (by reducing the pupil width). Poser rendering however is not capable of doing that (yet). This may cause overlighting on the image.

Hence, there are no limitations in balancing the various portions of a single Poser material, even when representing a 'photo real' surface. Choosing any values too high however can make an object overlit, or give it a very unnatural appearance. See 23 ARE THERE ANY PRACTICAL LIMITATIONS IN COMBINING DIFFUSE, SPECULAR, … IN A SINGLE MATERIAL? on this.

Not for a "conservation of energy" concept or alike, but for a practical reason on the prevention of overlighting, there is nothing to stop me from using somewhat realistic values for Diffuse (80% to 60% will do), for Reflection (1% for organics, 4% for liquids to 10% for glass-like stuff will do, except for metals which may go 60%-98% but hardly Diffuse as a consequence) and similar values for Specularity as well.

22 Are there more principal limitations in combining Diffuse, Specular, … in a single material?

The short answer is NO, and be advised that there are a lot of contrasting opinions in the forums and around on the Internet. Some opinions are presented as religion, and their advocates act as true evangelists. Others are more relaxed. What does it matter?

Well, 3D programs like Poser are very feature rich, and far from easy to comprehend fully (at least to me). And finally, I need a "way of thinking" to use those programs effectively and efficiently. That is: for establishing a specific effect in my result, I want to get a good way to accomplish that, without wasting a lot of time (and other resources) on thinking it up, implementing it, testing, fine-tuning and rendering.
The best "way of thinking" for me depends on my background (experience, interests, education, expertise) and the kind of effect and result I want to accomplish. Stylish results for comics have different requirements from studies for oil-painting or crayon-drawing, from product-casing, from animation or from photoreal'ish rendering, either for web-galleries or for large scale fine print.

So in some cases the "let's not add up to 100%" approach is good enough, in other cases people might prefer a complete Master of Physics approach, or anything in between. It's good to have all those variants around, and to add some of them to a personal toolbox.

More in detail

Various opinions arise from the limitations on materials in nature, when a single source of light hits a surface. That light can be reflected, diffused, transmitted and so on, and hence these portions should not add up to a value larger than one (100%). It could be less, to represent absorption in the material.
However, the various portions of a Poser material respond to quite different – and quite independent – sources of light. This is presented in a basic way in 21 ARE THERE ANY PRINCIPAL LIMITATIONS IN COMBINING DIFFUSE, SPECULAR, … IN A SINGLE MATERIAL?. And there are various practical considerations to be taken into account, as discussed in 23 ARE THERE ANY PRACTICAL LIMITATIONS IN COMBINING DIFFUSE, SPECULAR, … IN A SINGLE MATERIAL?.

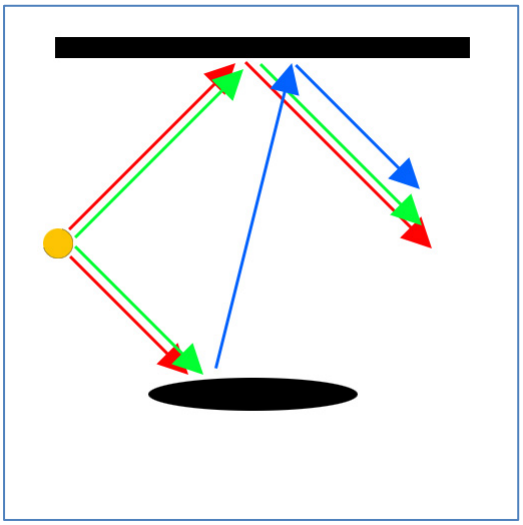
Intermediate

For a better understanding, let's review the situation for direct lighting and for indirect lighting separately. In both cases, Bump and Displacement, as well as Transparency and Refraction will only affect the distribution (and eventual reduction) of light re-emitted by the object towards the camera into the render, but will certainly not increase the amount of it. So these aspects can be ignored for the moment.

For the sake of simplicity, let's first consider a single (direct, point) source lighting an object in the midst of a scene, under non-IDL conditions:

Direct Lighting only

The diffuse portion of the light will be re-emitted by the Diffuse portion of the material, while the specular portion of the light will be re-emitted by the Specular part of the material, emulating the reflection of the direct light source (the lamp) itself. The Reflection part of the material will bounce the diffuse and specular results from the light source via other objects in the scene onto the object surface (unless a reflection-map is used to fake the effect).



Red: direct diffuse light into scene and onto object, resulting in general visibility
Green: direct specular light into scene and onto object, resulting in highlights= reflection of lamp
Blue: visibility and highlights combined from the scene reflected onto object = reflection of scene

Note again: the reflective surface of the reflecting object not only does not reflect the direct light source itself (as a lamp is not considered an object at all), it does not reflect the light rays either. So I cannot use a flashlight and a mirror to illuminate an object around the corner.

Since Reflection and Specular both represent the same reflective properties of the material for just different sources of light, they should be balanced somewhat for the objects surface. This effectively means I'm balancing the visibility of the lamp with the visibility of the scene objects, both onto the surface of the object at hand.
And – for that same object surface – I'll have to balance its Reflection / Specular at one hand to its Diffuse at the other hand, to make a neat impression of the objects reflectivity as such, despite the fact that the sources of the lighting are different.

On top of this, the material can offer some glow of its own, in the Ambient or Translucence channel (*). This will wash out some surface details of the object, and will just add to the light re-emitted from the surface by Diffuse, Specular and Reflection.

InDirect Lighting conditions

Second, let's switch on IDL conditions. Now all the light (diffuse, specular, reflections, ambient glow, etc.) re-emitted by any object in the scene works as a diffuse light onto the object (and all other objects) as well, in addition to the already available direct lighting. Effectively, this will turn up the diffuse lighting level seriously. Not only will this turn my render into an overlit one unless I lose a few lights and turn down the intensities of others, it will also distort the balances between the material portions sensitive to direct light (Diffuse, Reflection), and the portions which are not (Specular, Ambient).

For Specular, this implies that we either have to re-balance this property against Reflection and Diffuse, or we have to rebalance the specular properties of the available direct lights against the new IDL-based lighting level. The latter part might take far less effort, especially in material-rich and/or crowded scenes.

To complete our understanding on Reflection, let's re-consider the use of a flashlight, shining on a mirror in an attempt to illuminate an object 'around the corner'. The (direct) flashlight will produce a bright spot from diffuse and specular on the mirrors surface. Under IDL conditions, this spot will act as a diffuse light source, shining all around creating some extra illumination 'around the corner'. There still will be no defined bundle, though.

For Ambient/Translucency (*), the hotspots (led-lights or so) on the object might need to glow a bit more to be seen within the altered IDL lighting level.

(*) Ambient and Translucence are equivalent parts in the material definition. Just use either one or the other. The main difference is that Ambient is supported by the Simple interface, and by most exports to non-Poser file formats and renderers, while Translucence is not.
See 39 WHAT'S THE TRANSLUCENCE COMPONENT INTENDED FOR? for more details on this.

23 Are there any practical limitations in combining Diffuse, Specular, … in a single material?

The various portions of a material definition all add up to the surface response to the lighting in the scene. In order to present a realistic look – to any extend – one has to balance the contributions of those portions to each other, and to the various kinds of lighting in the scene. See 21 ARE THERE ANY PRINCIPAL LIMITATIONS IN COMBINING DIFFUSE, SPECULAR, … IN A SINGLE MATERIAL? (the short answer) and 22 ARE THERE MORE PRINCIPAL LIMITATIONS IN COMBINING DIFFUSE, SPECULAR, … IN A SINGLE MATERIAL? (elaboration) on this.

Intermediate

Next to that, all those portions of a material definition just add up mathematically, on a per-color basis. So RGB (80%, 60%, 15%) + RGB (60%, 50%, 10%) = (140%, 110%, 25%).
This may cause overlighting (any value >100%) from

= Diffuse * all direct and indirect diffuse lighting
+ Specular * all (direct) specular lighting
+ Reflection * all light from other objects in the scene
+ Ambient + Translucence (both representing glow from the object itself)

I can dim the light (which will make the ambient glow look stronger) or I can dim all material aspects proportionally. This latter will darken the image (which can be corrected for in post, using Photoshop or GIMP or alike) but at least will avoid the clipping effects. Note that applying Gamma Correction (GC, see 20 A POSERSURFACE MATERIAL OFFERS DIFFUSE, SPECULAR, ETC. HOW DO THESE WORK TOGETHER? on this) automagically caters for some darkening, overlighting reduction and re-brightening and should be used when possible. GC is available from Poser 10 on, and from Poser Pro 2010 on.

For instance,
assuming all colors to be white so we've got to consider intensities and brightnesses only,
say a light producing 90% diffuse and 80% specular light shines on an object having

- 80% diffuse
- 20% specular
- 20% ambient

This will produce 90% x 80% = 72% in the diffuse channel, plus 80% x 20% = 16% in the specular channel, plus 20% in the ambient channel, equals 72 + 16 + 20 = 108% which will get clipped down to 100% when presenting the result.

But do note

- That addressing diffuse and specular lighting separately requires the Advanced interface to the material room, using the color swatches for both
- Setting the intensities for Diffuse, Specular and Ambient can be done via the Simple interface, as well as via the Advanced interface, using the appropriate color swatches
- All color swatches for light and material will get affected by the Gamma Correction (GC) mechanism while rendering, which reduces the overlighting effect by itself. GC is available in Poser from version 10 up, and in Poser Pro from version 2010 and up. See 20 A POSERSURFACE MATERIAL OFFERS DIFFUSE, SPECULAR, ETC. HOW DO THESE WORK TOGETHER? for its use, and 98 WHAT'S THE FUZZ ABOUT GAMMA CORRECTION? for details.
- Setting the intensities for Diffuse, Specular and Ambient can be done via the Advanced interface, using the appropriate value dials. These are not affected by the GC mechanism, and hence will produce a different result from dimming the color swatches.

24 A PoserSurface material component, offers Color and Value. How do these work together?

In 20 A POSERSURFACE MATERIAL OFFERS DIFFUSE, SPECULAR, ETC. HOW DO THESE WORK TOGETHER? it is discussed that the various components of the PoserSurface definition are simply added up on a color by color basis to make the final result. This topic is on the individual elements of such a component.

In the Simple interface, I can have an image map combined with a color swatch. The color swatch acts like a filter, as if I'm looking at the image through a transparent, colored piece of plastic.

In technical terms, both are multiplied. For instance, take a red swatch, or: RGB=(100%,0%,0%) multiplied with any color (red, green, blue) on a color-by-color basis. That will give: (100% red, 0% green, 0% blue) of just (red, 0,0). The red filter will only let the red color pass through, and will kill the other color parts. That's why multiplication means: filtering.

Intermediate

The Advanced interface to Material Room offers Value slots next to Color ones, and more nodes to plug into the slots than just image-map.

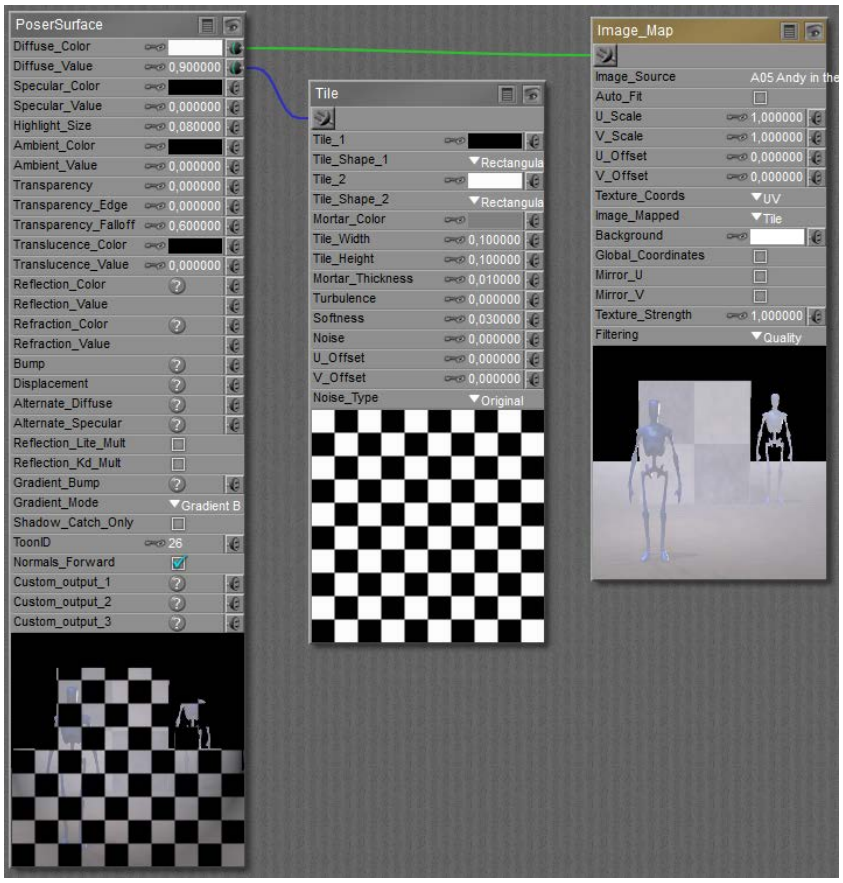
When a tree of nodes, from either no node at all to a whole complex of combinations, is plugged **into the color swatch**, then the result from the node-tree and the swatch are multiplied on a color-basis. Red times red (both in percentages, 90% x 80% = 72%), green times green and blue times blue. When a value is supplied by the node chain, then that value is used in all three color channels instead. When no node is attached to the color swatch, then just the color swatch itself results from the transaction or: a node-value of 1 is used.

When a tree of nodes, from either no node at all to a whole complex of combinations, is attached **to the value plug**, then the result from the node-tree and the plug are multiplied on a value-basis. Both in percentages, 90% x 80% = 72%. When no node is attached, then just the plug value itself results from the transaction or: a node-value of 1 is used. When the node tree offers a color instead, then this color is turned into a brightness value first.

Then, the resulting value is multiplied to the resulting color, for red, green and blue separately.

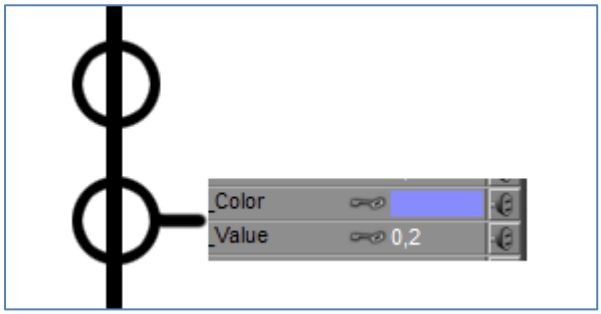
Example:

Image with brightness and color variation plugged into Diffuse Color, and different image with brightness and color contrast plugged into Diffuse Value. It's like stacking filter on filter on filter …



And for now… something different

Technically, various components offer a filtered access into the spine of the PoserSurface material definition. Whatever I plug in, it gets filtered (aka multiplied) by the Color and Value present. The Color will be affected by Gamma Correction, the Value will not. So 80% White and 100% Value will behave different from 100% White and 80% Value under GC render conditions. See 98 WHAT'S THE FUZZ ABOUT GAMMA CORRECTION? for details.



This holds for Ambient and Translucence, and for Reflection and Refraction.

Diffuse and Specular are similar, but include a build-in shader as well which manages the distribution of light intensities over the surface. See 05 WHAT'S THE DIFFUSE COMPONENT INTENDED FOR? and 41 WHAT'S THE DIFFERENCE BETWEEN THE DIFFUSE COMPONENT, AND THE DIFFUSE NODE? on Diffuse, 06 WHAT'S THE SPECULAR / HIGHLIGHT COMPONENT INTENDED FOR? and 43 WHAT'S THE DIFFERENCE BETWEEN THE SPECULAR COMPONENT, AND THE SPECULAR NODE? on Specular, and 30 CAN I GET A (SIMPLIFIED) EXPLANATION ON LAMBERT AND DIFFUSE SHADING? on the Lambert shading itself.

Alternate_Diffuse and _Specular in turn offer the color swatch (filtering) only. Bump and Displacement (and Transparency) are different, they act as special surface modifiers.

25 Can I get a brief intro on Node-tree building in Material Room?

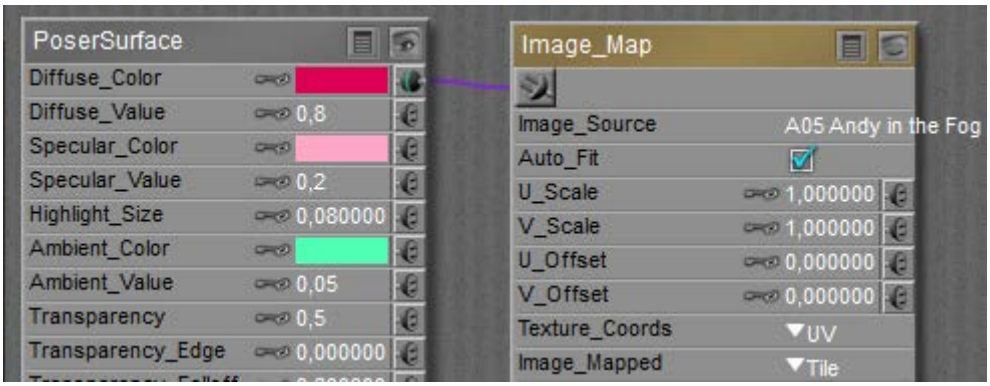
Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

Intermediate

In a mathematical way (rendering is all about endless computations, isn't it) the PoserSurface definition can be read as

PoserSurface =
(Diffuse + Alternate_Diffuse) * (1-Transparency) +
(Specular + Alternate_Specular) +
(Ambient + Translucence) +
Reflection + Refraction

while Bump and Displacement act as special modifiers (and Transparency does have various side effects when reflection and refraction are applying raytracing).



A component of the PoserSurface can be read as

Diffuse =
(Diffuse_Color * DiffuseColor-input) *
(Diffuse_Value * DiffuseValue-input)

where each "input" refers to the result of any kind of node having its output connected to it. When there is no node attached, the input acts as a 1.0, or: neutral in the multiplication.

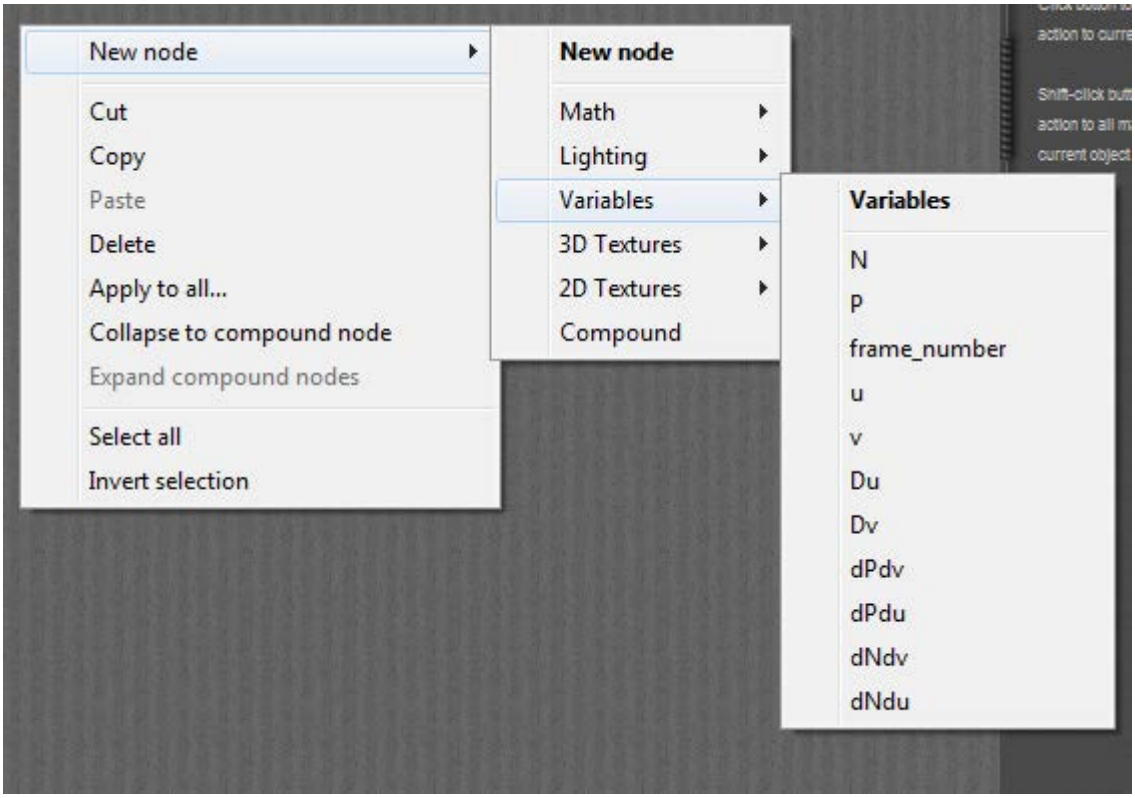
Each node offers one output which can be connected to (serve as input for) one **or more** (!) input connectors. Each input connector can have at most one output connected to it. If I want a combination of nodes attached to an input socket, I need a construction node to define the combining math myself.

As the example above reveals, a node (like Image_Map) offers inputs or parameters of various nature. Some of them are 'autonomous', like the Image_source involved or the Auto-fit option. These cannot be driven by other nodes any more. But other inputs, like U_Scale, can offer a dial-value filtering of any additional results from other nodes connected to it.

Thus far, all node inputs and outputs are visible in the interface, and can be addressed explicitly. But be aware that a lot of nodes also have some additional interactions with the Poser scene or system.

- The root node, PoserSurface, looks like a regular node with inputs but its output slot is "missing". It sends its results to the renderer instead.
- Actually, things work the other way around: for each surface element the renderer needs some results for, the PoserSurface function is called for that surface-material definition. And that function calls the other functions according to the nodes attached to its inputs, which may call functions according to the nodes attached to their inputs and so on.
  In other words: PoserSurface is not pushing results into the renderer but the renderer is pulling results from PoserSurface. Complex node trees then make large stacks of function calls, and the calls are made for each surface element in the rendering process. High numbers for pixel samples and/or low values for shading rate in the Render Settings increase the number of evaluations in the renderer, and hence the amount of PoserSurface calls as well.

- From all that, the coordinates of that surface spot, in space **and time**, are available to all node-functions called. U,V,W with respect to the surface, X,Y,Z with respect to scene space, Framenumber with respect to time, and some more as referred to in the node Variables group (see 77 WHAT ARE THE VARIABLE NODES USED FOR? for details on each of them):



- Next to the 'backdoor' communications with renderer and scene place and time, the nodes are communicating with the lights.
  - The ones in the Lighting > Diffuse group (diffuse, clay, …) require any diffuse lighting from direct or indirect sources to produce their result. No such lighting, no output (0, black, …).
  - The ones in the Lighting > Specular group (specular, Blinn, …) require any specular light, and I need a direct infinite, point- or spotlight to produce that. No specular light

on that spot on the surface, then no output. IBL lamps and any light from objects (IDL, ambient, reflection, …) are considered diffuse, even highlights from object surfaces produce diffuse light under IDL conditions and are not considered specular themselves.

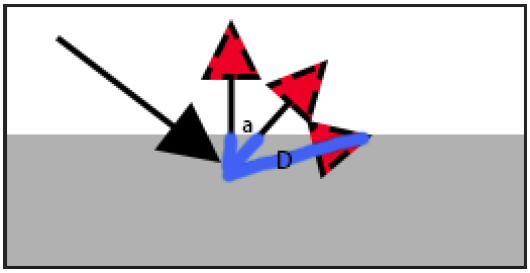
- The ones in the Lighting > Special group (skin, velvet, …) serve both purposes, they offer a response to diffuse light as well as to specular light, and even can combine that with some autonomous ambient effect. Some of them require the Subsurface scattering option in Render Settings to be switched on in order to produce any output at all.
- The ones in the Lighting > Raytrace group (reflect, refract, …) require light from surrounding objects, and do not work on the light from direct sources. When there are no such objects in front (reflect) or behind (refract) the surface at hand, they can't do their job and will produce their default 'background' response.

For example,

- I've got an object which has some response to diffuse or specular light, and produces an ambient glow too. And I've got no lights at all in the scene. Then only the ambient glow will show, as the Diffuse and Specular components have nothing to respond to, and the Reflection and Refraction lack any surrounding objects to work with.
- Now I connect the diffuse node to the Ambient_Color slot in an attempt to get some intensity distribution into the glow. What do I get? It kills the glow.
  The diffuse node will not receive any diffuse light from any source, therefore it produces a null response, which is input to Ambient_Color which will therefore produce black as well.
- So… nodes like those require light onto the surface and produce a response from that, they are not producing light distributions as such for output. No light, null response, black output.

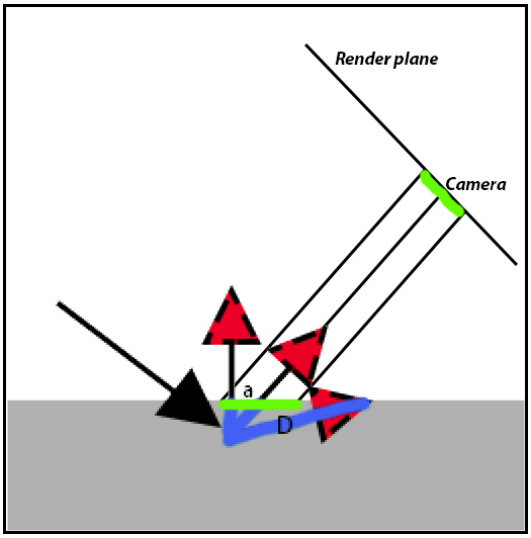
30 Can I get a (simplified) explanation on Lambert and diffuse shading?

Light from any direction hits a surface, penetrates it for a very slight amount – which will give some absorption, and then it gets re-radiated in a color-filtered way equally in all directions. The path back to the surface will give absorption too, proportional to the distance D traveled through the object surface.



As this distance D is inversely proportional to the cosine of the exiting angle ($D=D_0/\cos(a)$), the intensity of the diffuse light in that direction will be proportional: $\mathbf{I = I_0 \cos(a)}$.
This is the angular distribution of outgoing, diffuse light, according to the mathematician J.H. Lambert (about 1750).
At perpendicular scattering, angle a=0 so cos(a)=1 and the response is maximal, while at parallel scattering cos(a)=0 and there is no response at all.

And since cosine calculations are hardwired into modern CPU electronics, this is a speedy rendering approach by any means. Therefore, Poser includes Lambert shading into Diffuse (see 05 WHAT'S THE DIFFUSE COMPONENT INTENDED FOR? and 41 WHAT'S THE DIFFERENCE BETWEEN THE DIFFUSE COMPONENT, AND THE DIFFUSE NODE?). This offers a resource-friendly first step towards more realistically looking renders. And, for people who want more steps in that direction, Poser offers alternatives like Clay (see 42 WHAT'S THE CLAY NODE USED FOR?).



Now, look what will happen to the render result. A specific area on the render plane (say: a pixel), marked green in the illustration, gets its light from an area on the object surface (marked green as well). At skewer angles a between surface normal and camera, this area on the object gets larger: $\mathbf{A = A_0 / \cos(a)}$.

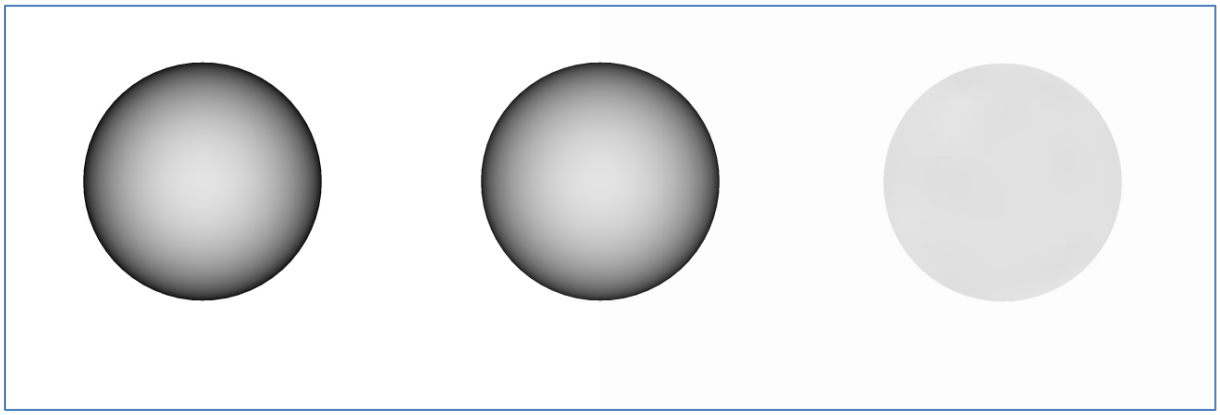
At skewer angles such an area emits less light per unit of surface (cm^2 or alike) and the resulting amount of light onto the pixel in the render plane is
$\mathbf{I * A = I_0 \cos(a) * A_0 / \cos(a) = I_0 * A_0}$ is a constant

So the Lambert shading not only matches a nice explanation on diffuse lighting, it also makes that the intensity of light in the render result does not depend on the camera angle to the surface. Because at skewer angles the pixel in the render represents a larger area on the object surface, which diffuses less light towards the camera, and both effects cancel out.

Look at the light

This leaves the effect of the incoming light itself. At skewer angles, the same amount of light will hit larger and larger areas of the object surface, so the intensity per unit of surface decreases accordingly: $\mathbf{L = L_0 \cos(b)}$.
Besides the math, this means that the extent to which shading reveals the shape of an object, depends on the 'directivity' of the light only. Point lights are quite directive, even a flat surface it lit under varying angles. An infinite light is non-directional, a flat surface is evenly lit but a ball is not. IDL lighting is hardly directive at all, the light comes from all directions and all surface areas are equally lit

whatever the shape of the object. Hence the shape of objects is less revealed, and objects will look flat.



Left: point light nearby, the edges get darker faster.
Mid: Infinite light.
Right: IDL, the ball looks like a disk.

31 What's a proper Color for Reflections?

Or: what do I have to put into the Reflection Color swatch?
For short: white, except for colored metals.
Eventually with a reduced brightness for reflectivity according to 32 WHAT'S A PROPER VALUE FOR REFLECTIONS?, which is a much recommended approach when applying Gamma Correction (as available in PoserPro, and Poser 10 and up) to the rendering process.
Merging the reflectivity value into the color swatch is even a necessity when using Alternate_Diffuse instead of Reflection_Color, as Alternate_Diffuse does not offer a Value slot. This approach is recommended when the material definition offers some Transparency as well. See 09 WHAT'S THE REFLECTION COMPONENT INTENDED FOR? on details.

On about all sorts or materials, reflecting surfaces are neutral, as having a blank lacquer on top of it. The incoming light bounces directly from the surface without entering it at all, and it's from **within** the surface that objects get their color. Reflections do not filter the color, and hence are best represented by some shade of gray (white for 100% reflection).

This however is not the case for metals. For those materials, light does not enter the surface (so formally metals don't have diffusion) but some colors bounce more effectively than others. Gold bounces best in the yellow/red range and less efficient in the blues, copper bounces best in the deeper reds, while more bluish metals bounce better in the blues than they do in the reds. See 32 WHAT'S A PROPER VALUE FOR REFLECTIONS? for details on colors of metals and alloys, and 33 WHERE DO THE REFLECTION COLOR AND VALUE TABLES COME FROM? for a detailed discussion.

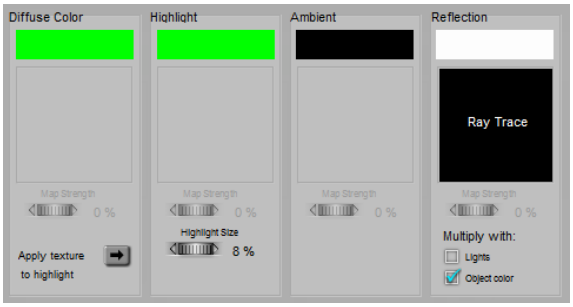
In a simple Poser setup, one can choose to color the reflections directly.

Intermediate

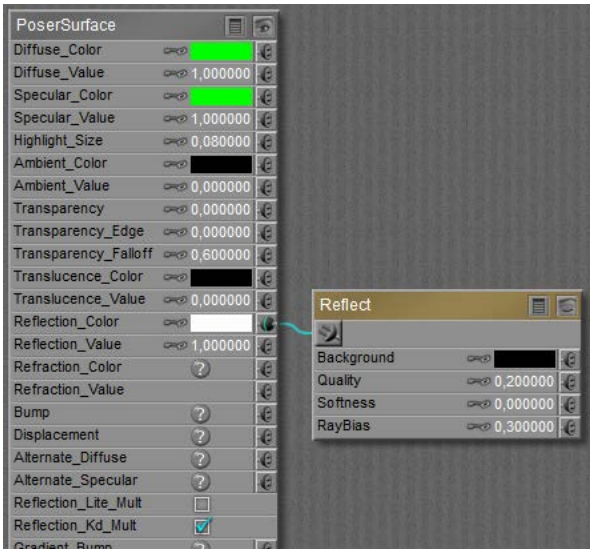
In a more complex setup, like a colorful object with dents and rusty spots, one would have to put a (perhaps even modified) copy of the diffuse channel into the reflection channel in the material definition to filter the reflections. That's pretty tedious for a large Diffuse node tree.
So Poser supports this by providing the **Reflection Kd Mult** option (called **Multiply with Object Color** in the Simple interface) which forces the reflection to be multiplied by whatever results from Diffuse Color. This option is OFF by default and is intended to be switched ON for metals, and for metallic materials like some car paints.

Option ON

Reflection applies the Object (Diffuse) color using the Multiply option. The Reflection Color itself should remain white to avoid double filtering. In this case, even metals should have a Diffuse color too!

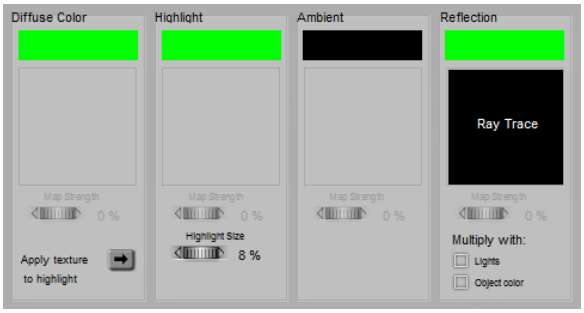


or

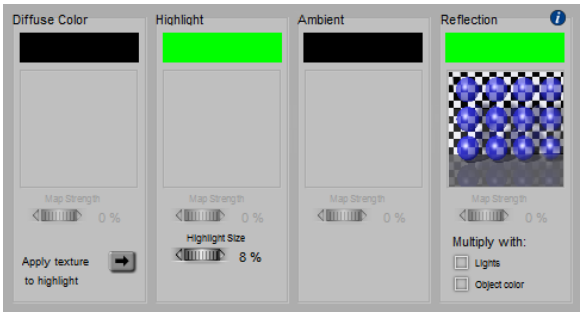


Option OFF

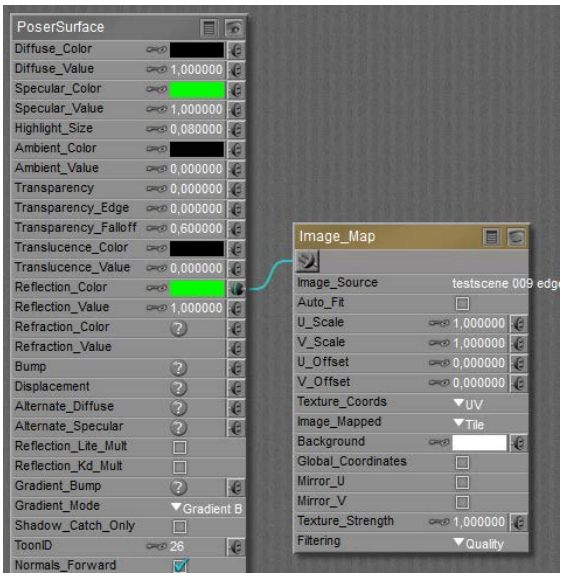
Now Reflection has a color of its own, for metals that is.



Formally, metals have a high reflectivity and no diffuse. Ensure that there is always something to reflect, then, using Raytrace (above) or an image (below). As there is no Diffuse, do not multiply with it!



or



In all cases above, note that Reflection deals with objects in the scene only. So I have to set up Highlight / Specularity in such a way that this ‘reflection of direct lights’ is handled similar to the other reflections. Metals color, non-metals don’t, as a guideline.

32 What's a proper Value for Reflections?

Or: what do I have to put into the Reflection Value field (Advanced interface)? Or: with what factor do I have to dim (multiply) the color I used according to 31 WHAT'S A PROPER COLOR FOR REFLECTIONS?
The latter is a much recommended approach when applying Gamma Correction (as available in PoserPro, and Poser 10 and up) to the rendering process. In that case, the Value itself should remain 1.0.
Merging the reflectivity value into the color swatch is even a necessity when using Alternate_Diffuse instead of Reflection_Color, as Alternate_Diffuse does not offer a Value slot. This approach is recommended when the material definition offers some Transparency as well. See 09 WHAT'S THE REFLECTION COMPONENT INTENDED FOR? on details.

The value I'm looking for is: **reflectivity**.

On contrast to common expectations, shiny and reflecting materials only bounce a very limited amount of the light received, back towards the camera, into the render. In jargon: reflectivity is low. That expectations are high is understandable:

- Highlights on a surface do stand out. This however is not due to the reflectivity of the surface, but because the lights which are reflected do stand out in their environment themselves. In Poser however, highlights are addressed by specularity, not by reflection.
- Water, car and glass surfaces do appear quite reflective. This is because reflectivity increases enormously with skew angles, and that's the usual way we see the environment reflected on those objects.

A plain, straightforward, perpendicular reflection, enabling me to see my own face reflected by the surface, is only strong in metals, and mirrors since these are coated with a metallic layer at the backside. The reflectivity of common materials can be found in the table presented below. The table also mentions the color, if any.

Table of Refraction Index (IoR) and Reflectivity (R)

Material	Refraction index (**)	Reflectivity	Coloring (RGB)	Result (RGB) = Reflectivity*Color
Water	1.33	0.020		5, 5, 5
Sugar solution	1.42	0.030		8, 8, 8
Oily fluids	1.50	0.040		10, 10, 10
Glass	1.50	0.040		10, 10, 10
Heavy Glass	1.60	0.053		13, 13, 13
Impure glass	1.80	0.082		21, 21, 21
Opal	1.45	0.034		9, 9, 9
Quartz	1.50	0.040		10, 10, 10
Salt	1.50	0.040		10, 10, 10
Amber (*)	1.55	0.047	Brown/orange (90.2%, 68.6%, 0%)	12, 12, 12
Onyx, Amethyst	1.55	0.047		12, 12, 12
Pearl	1.60	0.053		13, 13, 13
Aquamarine, Emerald (*)	1.60	0.053	Light Cyan, Green (0%, 66.7%, 45,1%)	13, 13, 13
Turquoise, Tourmaline (*)	1.65	0.060	Dark Cyan (0%, 50%, 50%)	15, 15, 15

Sapphire (*)	1.77	0.077	Dark Red (50%, 0%, 0%)	20, 20, 20
Zirconia	2.15	0.133		34, 34, 34
Diamond	2.40	0.170		43, 43, 43
Lead	2.60	0.200	Bluish Grey (50%, 50%, 62.5%)	41, 41, 51
Titanium	6.15	0.519		132, 132, 132
Tin (Sn)	6.54	0.540		138, 138, 138
Chrome	6.76	0.551		141, 141, 141
Nickel	8.79	0.633		161, 161, 161
Platinum	9.45	0.654		167, 167, 167
Copper	18.78	0.808	Reddish brown (86.3%, 35.3%, 0%)	206, 84, 0
Gold	36.81	0.897	Reddish yellow (100%, 70.6%, 0%)	229, 161, 0
Aluminum	43.43	0.912	Bluish (95%, 95%, 100%)	221, 221, 233
Silver	119.20	0.967		247, 247, 247
Zinc	17.94	0.800		204, 204, 204
Steel	11.25	0.700		178, 178, 178

See http://RefractiveIndex.info for details on all sorts of stuff.
See http://colors.findthedata.org/ for colors.

(*) for fluids, glasses and gems the color affects the light passing through. The reflection color however is white, as these aren't metals.

(**) On Refraction Index (warning: ***High School Math stuff ahead***):

For all materials, non-transparent ones like metals included, Reflectivity (R, as discussed in 09 WHAT'S THE REFLECTION COMPONENT INTENDED FOR?) and Refraction Index (IoR, as discussed in 38 WHAT'S THE REFRACTION COMPONENT INTENDED FOR?) are related:

$R = [(IoR - 1)/(IoR + 1)]^2$ and/or $IoR = (1+ \sqrt{R}) / (1- \sqrt{R})$ ($\sqrt{}$ for square root)

This too stresses that reflectivity is quite low (less than 10% in most cases) for transparent materials like fluids and glasses, while the Index of Refraction is pretty high (usually 10 and up) for non-transparent metals. It's this IoR value that has to be used in Fresnel nodes and the like.
When you like to have a more detailed understanding, try a physics class on optics for a change. Kidding, it's complex stuff.

Alloys

Metals are combined, usually for the physical properties of the resulting alloys. These are stronger, more flexible, more durable, less brittle, cheaper, and so on, compared to the pure stuff. For Poser renders, I'm doing quite well when simply using the mixing percentages for the resulting color and reflectivity.

Alloy	Mixture
Yellow Gold (22K)	92% Gold, 5% Silver, 2% Copper 1% Zinc
Red Gold (18K)	75% Gold, 25% Copper
Rose Gold	75% Gold, 22% Copper, 3% Silver
Pink Gold	75% Gold, 20% Copper, 5% Silver
White Gold	75% Gold, 25% Platinum
Soft Green Gold	75% Gold, 25% Silver
Green Gold	75% Gold, 20% Silver, 5% Copper
Purple Gold	80% Gold, 20% Aluminum
Brass	67% Copper 33% Zinc
Bronze	88% Copper 12% Tin
Yellow Copper (Messing)	60% Copper, 40% Zinc
Pewter	90% Tin 10% Lead

For example: when mixing 90% Gold (reflectivity 0.897) and 10% Silver (reflectivity 0.967) then the resulting alloy will have a reflectivity of 90% x 0.897 + 10% x 0.967 = 0.994. And I can do a similar thing to the RGB values of their respective colors.

Making an Easy Life hard

So, I can look up the color and the reflectivity of a metal, a glass or fluid, put these values in the Reflection_Color and Value fields respectively, and I'm done?

Not really, as I have to compensate for double-counting. For instance, the material at hand has no specific reflection color but does have a 70% reflectivity.
Then I can either put in White for Color and 70% for Value, or I put in 70% Grey for Color and 100% for Value. But I should not combine 70% Grey in Color with 70% in Value, as that will reduce the surface reflections to 70% x 70% = 49% instead.
This especially requires some care when a color is applied indeed. When that color has 90% brightness, then the first step in reduced reflectivity is already taken care of. Applying a reflectivity value of 80% as well will reduce the surface reflections to 90% x 80% = 72%. Vice versa, when I want that say 70% overall, then I should enter the appropriate 90%/70% = 80% in the Value field.

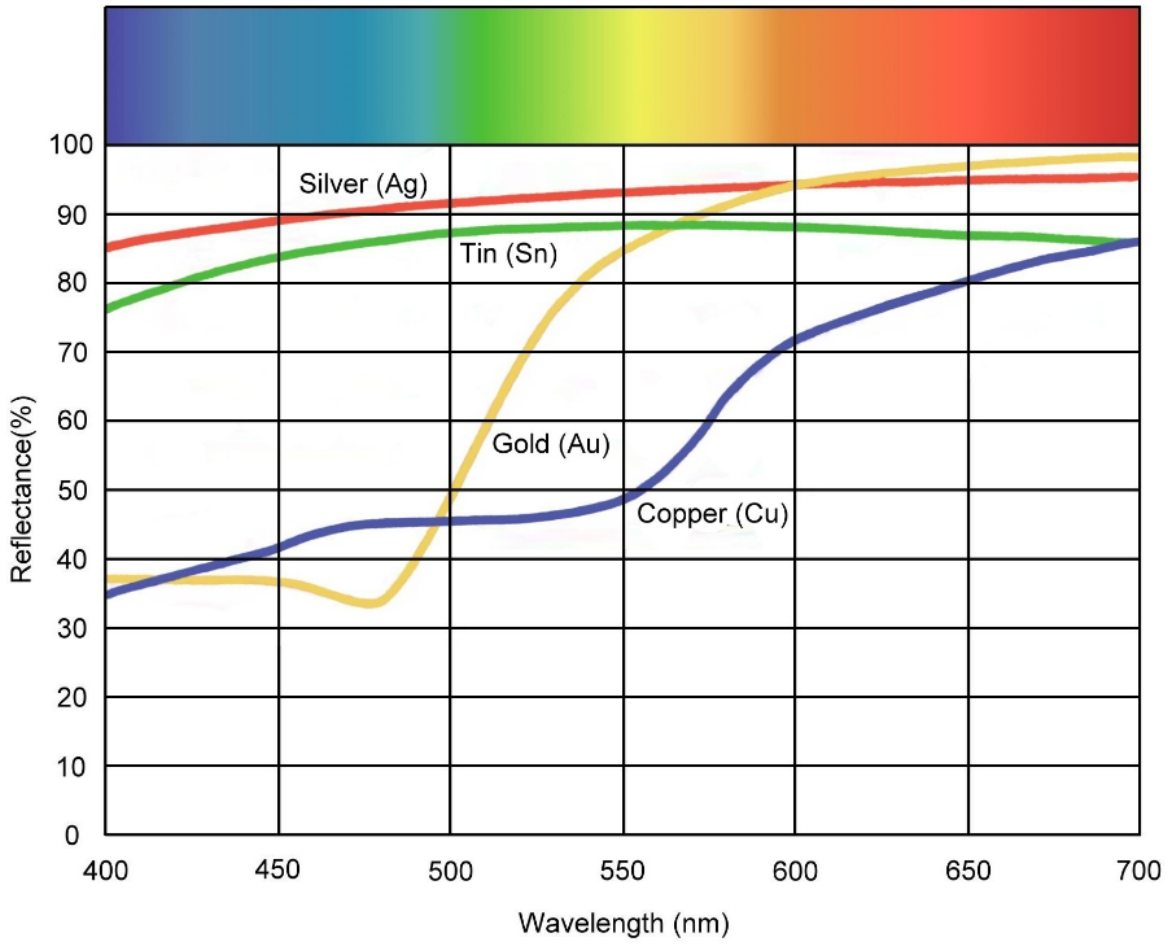
The recommended approach however is to reduce the brightness of the color swatch, and leave the Value at 1.0. Or even better, plugin any reflection node into Alternate Diffuse instead of Reflection_Color. See 09 WHAT'S THE REFLECTION COMPONENT INTENDED FOR? for details on both.

Okay, so I've got a color like RGB = (50%, 50%, 62.5%) or as Poser says: (127, 127, 159) and I want to turn it into its 100% brightness equivalent. How should I do that?

- Take the largest number, which is 62,5% or 159 for Blue in the example
- Divide all RGB values by that number, and multiply by 100% or 255 respectively
  In the example, that would make 50/62,5*100% => 80% , 80%, 100%
  or 127/159*255 => 204, 204, 255.
- Now, the Value field can get its proper Reflectivity, or the color swatch can get dimmed to the proper result. A 20% reflectivity will then result in 16%,16%,20% or 41,41,51.

33 Where do the Reflection Color and Value tables come from?

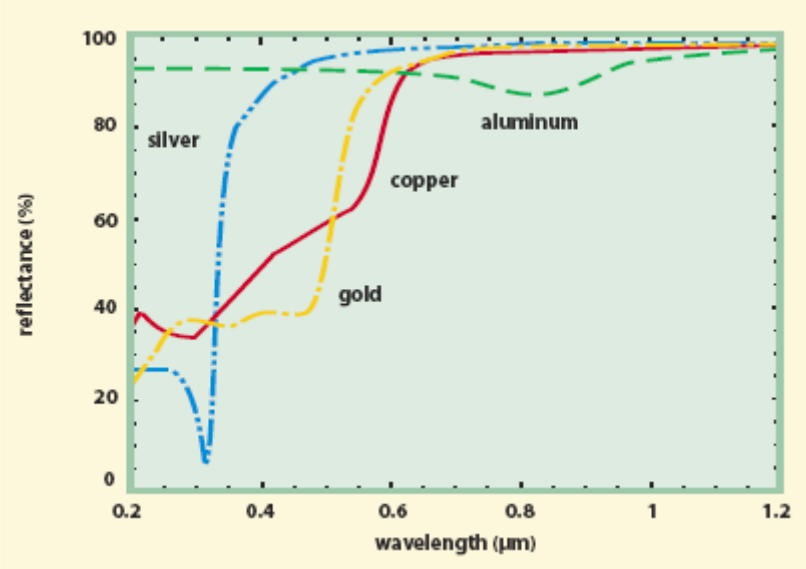
Mainly for industrial purposes, the response of light by metals is measured in detail, and published in handbooks, Wikipedia, and the like. Graphs basically look like:



Take Copper for instance. About 37% reflectivity in the blues, 47% in green and up to 80% to 85% in the reds. That might read as RGB = (82% , 47%, 37%) or Hue 14° (out of 360) Saturation 55% Brightness 82%. Taking 32 WHAT'S A PROPER VALUE FOR REFLECTIONS? in mind, I can use that color and leave value at 100%. Or I can increase the color to 100% brightness (RGB = 255,147,115 or 100%, 57%, 45%) and use the 82% reflectivity for Value. The first approach is preferred, and looks like .

Anyway, I also can see in the Graph that Gold compared to Copper is a far better reflector in the greens and yellows, and even worse in blue. That makes gold less red, more yellow, and a better overall reflector. Adding Silver for an alloy, a common practice, makes the gold even more reflective and more light-yellow. Silver itself is a very strong reflector in all colors and therefore will show White, while Tin will be slightly less reflective, and will have a very mild greenish taint over it.

The graph above is quite clear in its interpretation, but I might run into images like

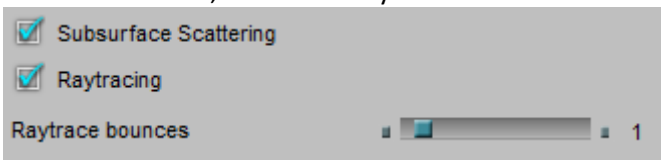


It's the same story, but showing light response for a much wider color spectrum. Visible blue matches 400 nano-meter = 0.4 micro-meter while the graph starts at 0.2. And visible red matches 700nm = 0.7µm while the graph makes it till even 1.2, infra-red heat, well reflected by all metals as we know.

And the graph adds aluminum, which as I can see reflects slightly stronger in the blues than in the reds giving it a mild bluish taint.

34 Do Render settings affect the behavior of materials?

A few of them do, and here they are:



Intermediate

Switching OFF the Subsurface Scattering option (Poser 8 / Poser Pro and up) will disable the nodes

- CustomScatter, see 56 What's the Special >CustomScatter node used for?
- Subsurface Skin, see 52 What's the Special > Subsurface Skin node used for?

to save render time at testing.

Switching OFF the Raytrace option disables IDL lighting, as well as the nodes from the Lighting > Raytrace group:

- Reflect, see 59 What's the Raytrace > Reflect node used for?
- Refract, see 60 What's the Raytrace > Refract node used for?
- Ambient Occlusion, see 61 What's the Raytrace > Ambient Occlusion node used for?
- Gather, see 62 What's the Raytrace > Gather node used for?
- Fresnel, see 63 What's the Raytrace > Fresnel node used for?

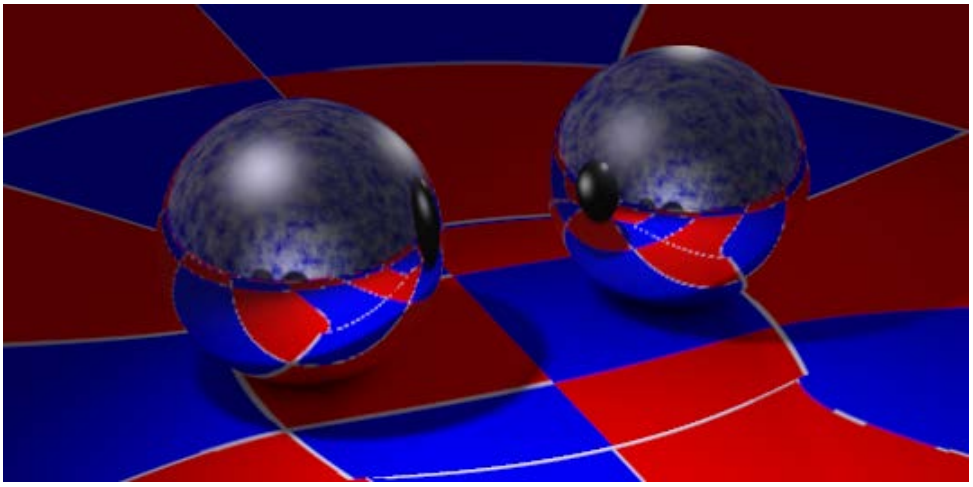
again, to save render time at testing. And, by the way, my lights can't have raytraced shadows either.

The Raytrace Bounces slider (from 0 to 12) does affect IDL lighting as well as raytracing in reflections and the like. Each surface passed or bounced at, counts for one, so passing a refractive object takes two. Rays gradually die a bit when bouncing, and when the limit as set is met, the ray gets killed anyway. This mainly affects internal reflections when reflectivity is combined with transparency.

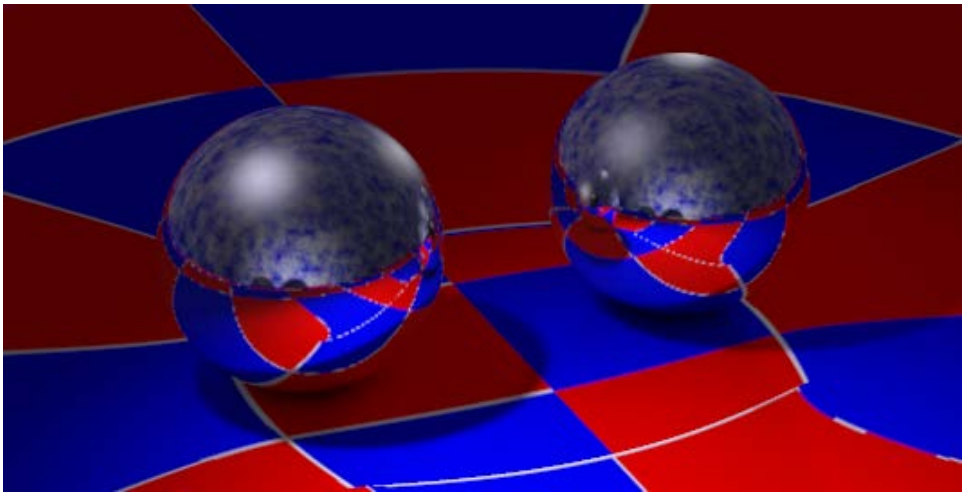
Reflection

Reducing Raytrace Bounces might result in pixels in the render which won't receive a ray of light, and remain dark. Or at least the reflection of the object is discontinued somewhat. In other words: incomplete spots in the render, artefacts. The higher the slider is set, the less is the risk that those occur.
And when the rays only need a few bounces anyway, then a high value won't make a difference and no killing takes place. The slider sets a max value.

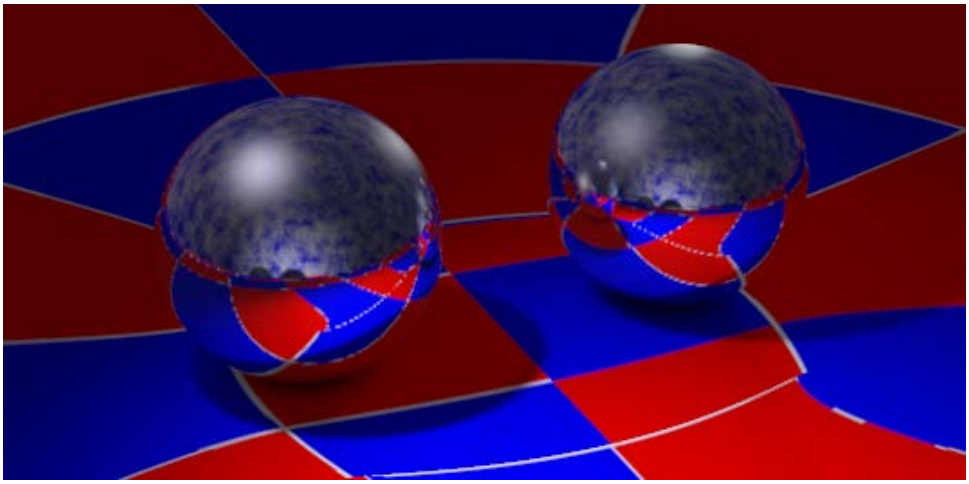
Reflect, max 1 ray, each ball reflects the other but not its surface color which is made by reflections.:



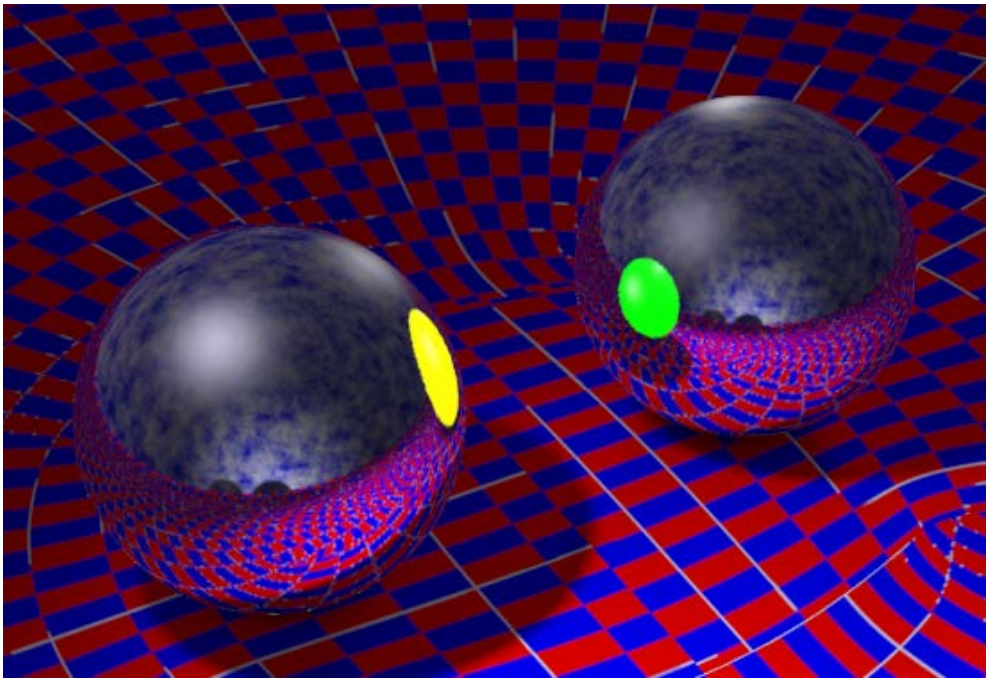
Reflect, max 2 rays, each ball can reflect the others surface but not its own reflection in that:



Reflect, max 4 rays, and render times are hardly prolonged:



The spots missing reflections are black because that's the color set as Background in the Reflect node. I can use any other color, or white with an image attached.



In that case, the Raytrace Bounces slider mixes raytraced and image mapped reflections: for the first (so many) bounces the reflection is raytraced, and from then on it's mapped.

However, increased slider settings hardly increase render times, reflection is an efficient process. This however might change drastically when transparency is introduced as well, see 64 HOW DO I PROPERLY COMBINE TRANSPARENCY AND RAYTRACING? on details. Then light not only reflects from the front, but also passes through the object to reflect from the (inside of the) backside of the object. And that ray will get reflected from the inside of the front side, and so transparency combined with reflection makes the infinite reflections of reflections of … etcetera that slows down rendering to its extremes.

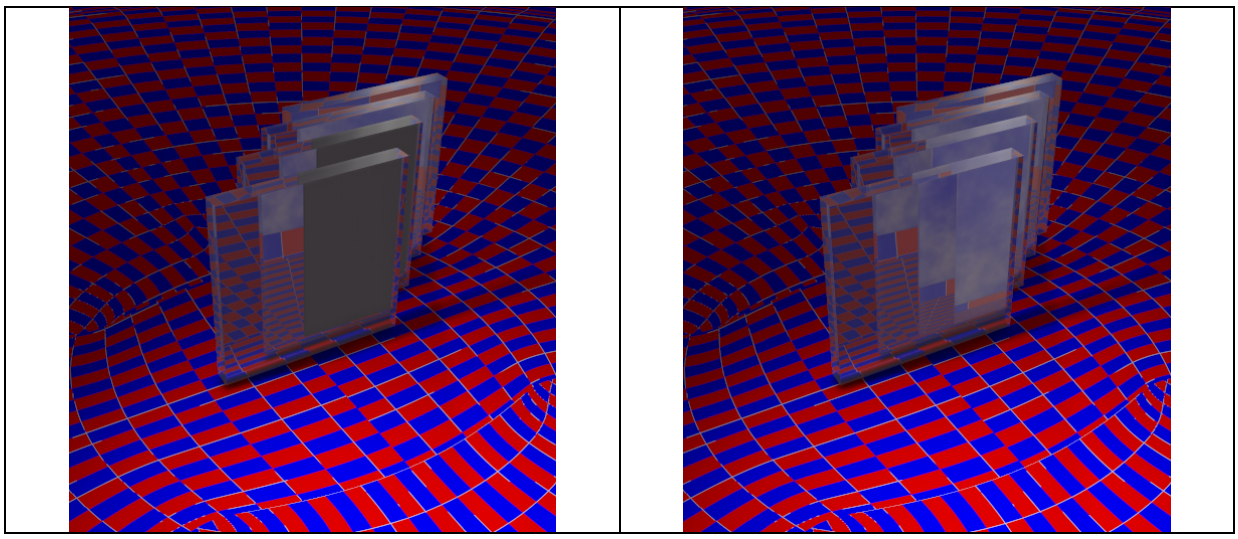
Refraction

First, we've got Transparency, which is able to let light rays pass through a surface; rays from direct light as well as from objects in the scene. And it does so without raytracing, so it's not affected at all by any Raytrace Bounces value. But it can't bend lightrays either.

Instead of – or on top of – transparency, Refraction makes lightrays bend as well when passing through a surface. And that's raytracing. But like Reflection, Poser Refraction deals with objects only and does not handle direct light itself. Without transparency, refraction will treat the surface as perfectly transparent for objects and applies bending as required.

So when objects are placed relative to each other to require refraction of refraction of … in the scene, and the Raytrace Bounces value is reduced, the light stops passing through the surface and might cause artefacts similar to Reflection. It depends on the amount of objects, each of them requires two bounces to let the light pass through, but two objects parallel to each other do not generate an infinite amount of mutual refractions like they can do with reflection. Hence, the Raytrace Bounces value does not need such high values anyway.

The number set is a maximum value, when Poser does not need them it won't use them, but if the number of bounces for a light ray exceeds this limit, the light ray is killed. This might speed up the rendering while it also might introduce artifacts (black spots) in the result. The tradeoff is mine, but as nature has an infinite number of bounces, the max value is the best when I can afford it.



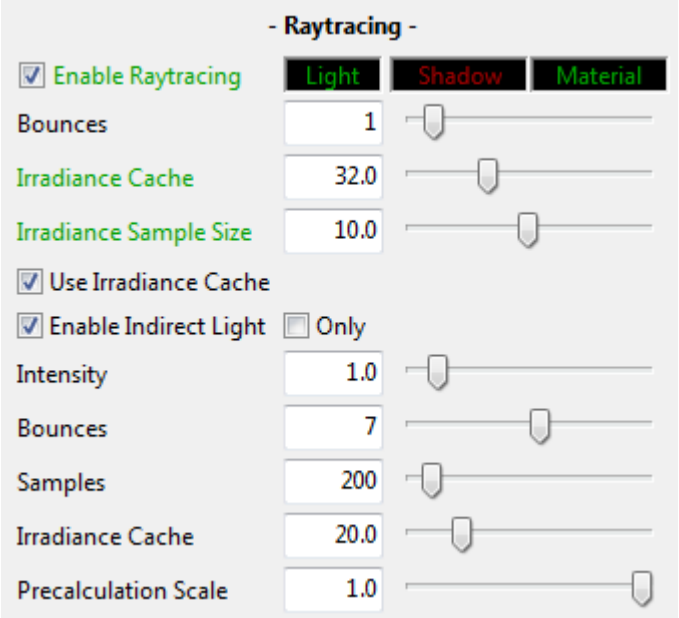
Left: Raytrace bounces set to 4, while 4 objects require 8 bounces.
Right: When the value is increased to 8 or more, all objects and surfaces can be passed.

And like reflection, refraction as such is quite an efficient process; until transparency kicks in, as discussed in 64 HOW DO I PROPERLY COMBINE TRANSPARENCY AND RAYTRACING?.

Indirect Lighting

To some extent, InDirect Lighting (IDL) is an application of reflection. Light hitting objects is diffused back into the scene, hitting other objects and so on. At each bounce the ray dies a bit, and after so many bounces it gets killed if it happens to be still around anyway. In open scenes a ray might get lost into open space, but most scenes applying IDL are encapsulated within a dome. Then killing rays really reduces the amount of rays around, and hence reduce the lighting level.

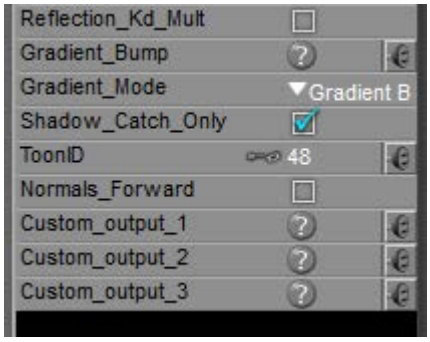
Some scenes do present dark spots under IDL lighting, especially in the corners where walls and ceilings meet. That's understandable: rays are bouncing around and one needs some luck to get a ray just in such a corner, instead of just bouncing away from the sides near to it. In such cases, killing rays early by a low Raytrace Bounces setting will increase the risk of missing a corner, and the corners will turn dark and splotchy. So an increased Raytrace Bounces value will reduce those artefacts, as it reduces artefacts in reflection itself.



Note that launching the render via the Scripts > partners > Dimension3D > Render Firefly menu gives me the opportunity to discriminate between raytrace and IDL bounce limits. So I can increase the latter without having the burden of large values for the first.

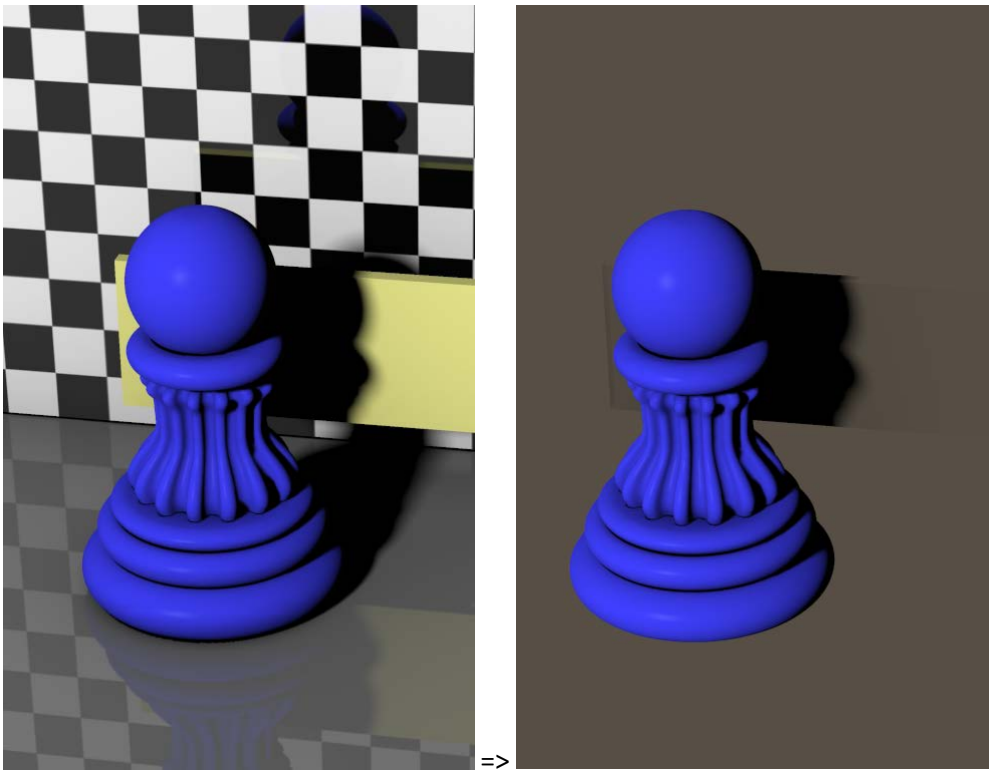
When I've also got direct lights in the scene (like a photographer uses a flash when working outdoors in the sun), this increase in IDL lighting levels will change the balance between direct and indirect light, and I might want to correct for that by altering the lighting levels at the sources of it.

35 What's the Shadow Catch Only intended for?



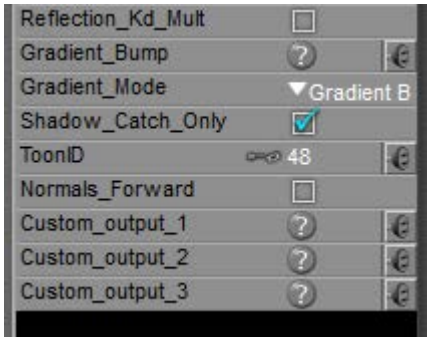
By switching ON the Shadow Catch Only option in a PoserSurface definition the surface will disappear completely, except from the shadows cast from other objects, and from itself as well.

Generally, this feature is used to create a render that blends well onto another image with objects around, and this is a way to transfer the shadows from objects in the Poser scene to the other objects in the background layer. The background objects are recreated in the Poser scene for their shape only (primitives do fine jobs in most cases, like blocks for buildings)



As the 'real' objects catching the shadows are in the background image, one should avoid having any objects behind (or visible through) those shadow catchers in the Poser scene. That's why the back wall and ground plane were made invisible in the right image. Otherwise, these will hamper the blending of the render with the background, and catch shadows as well.

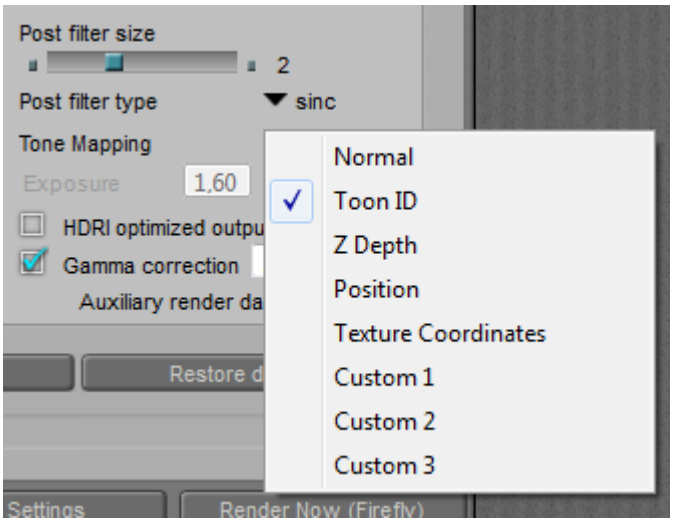
36 What's the ToonID intended for?



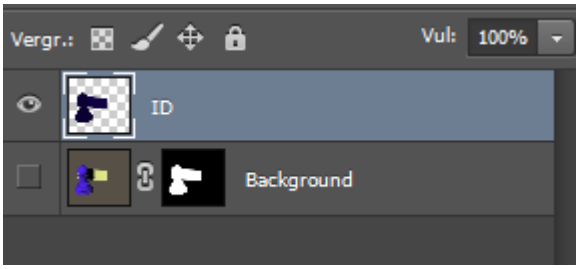
Each PoserSurface has a ToonID value assigned, and one can even make it varying over time (animated) as well as driven by a node construction, like a complex of math_functions 75 SHADER TREE CONSTRUCTION: MATH, VARIABLE AND COMPOUND NODES and variables like the frame_number 77 WHAT ARE THE VARIABLE NODES USED FOR?.

Various surfaces can get the same ID, when that helps me out. An instance of the render result, not showing colors and shades but showing ToonID's instead, can be derived as follows:

- In Render Settings, check the appropriate options in Auxiliary render data

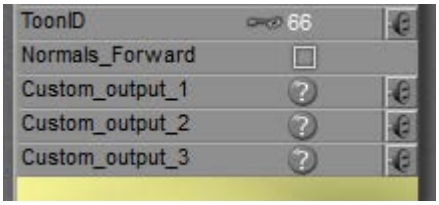


- After rendering, export the image as a Photoshop PSD file. This file will present an "ID" Layer in which each ToonID has a color of its own. This way the surface-areas can be selected easily for further modification in Photoshop (or alike).



Poser Aux render data and PSD layered export is available in Poser Pro only.

37 What are the Custom_outputs intended for?

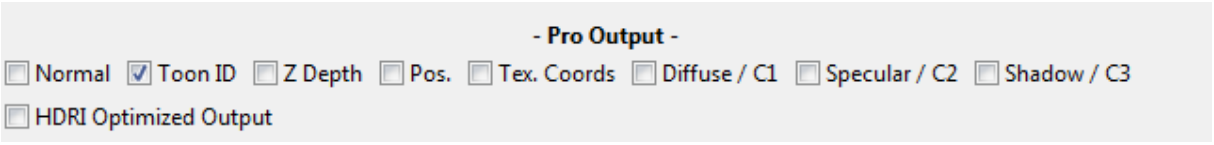


Like the ToonID (see 36 WHAT'S THE TOONID INTENDED FOR? first), checking the Custom 1, 2 and/or 3 Auxiliary render data options in Render Settings enables extra layers in the export of the render result in Photoshop PSD format. This PSD layered export is available in Poser Pro only, and so are the three Custom fields.

When no additional actions are taken,

- Custom 1 represents Diffuse
- Custom 2 represents Specular
- Custom 3 represents Shadow

When in doubt about the meaning of C1, C2 and C3, the Dimension 3D script for Render FireFly (in the Scripts > Partners menu) can help me out.



But I can take additional actions, and plug in any node tree into any of the Custom slots in which case that node tree takes over from the default meaning. It won't affect the render result at all, as the extra PSD layers are meant for supporting post processing only (masks, selections, …).

Establishing the extra info might affect render time and memory use of course and it will generate the content as I require into the appropriate PSD-layers, once I take that format for exporting the render result.

38 What's the Refraction component intended for?

What refraction is about…

When light passes a transparent material like glass or fluid, or more precise: passes a surface with different transparent materials on both sides (air/water, air/glass, water/glass, …), it bends. The angle depends on the physical properties of the material defined as Index of Refraction (IoR, see 32 WHAT'S A PROPER VALUE FOR REFLECTIONS? on values). The denser the material, the higher the value. The light bends towards the surface normal pointing towards the denser material, so from air to glass: inward, en from glass to air: outward. For a block of glass this means that the light coming out moves parallel to the light coming in, the block just shifts the scene behind the block somewhat aside.

What Poser is about…

The refraction component as such does not do anything specific. It takes any node construction plugged into it, filters it with a color and a value as discussed in 24 A POSERSURFACE MATERIAL COMPONENT, OFFERS COLOR AND VALUE. HOW DO THESE WORK TOGETHER?, and adds it to the PoserSurface set of material definitions. So I can put in an image_map, but in contrast to reflections that's not done often as the refraction needs to represent the scene behind the object, with shifts and bends according to the object shape and thickness. A proper image would be tough to construct.

The node supposed to be connected to it is Refract, from the Lighting group. This one is discussed in 60 WHAT'S THE RAYTRACE > REFRACT NODE USED FOR?. Note that although refraction mimics a full transparency of the object, the object itself remains opaque for direct lighting, and refraction is not reduced when transparency itself enters the stage as Poser Transparency affects Diffuse and Alternate_Diffuse only. One can reduce the Refraction_Value (or dim the Refraction_Color) accordingly, but the transparency from refraction, and transparency from PoserSurface are still different things.

Combining Transparency, Reflection, Refraction and Gamma Correction (from Render Settings) in one mix might seem a logical thing to do, but will raise numerous issues.

- Combining Transparency and Reflection causes long render times due to infinite internal reflections within the object;
- Balancing Reflection and Refraction requires fine instruments like Fresnel;
- Transparency and Refraction are sort of duplicating each other and
- The Gamma mechanism is well-known for distorting established balances between components.

You can find more on this in 60 WHAT'S THE RAYTRACE > REFRACT NODE USED FOR? on the refract node, in 63 WHAT'S THE RAYTRACE > FRESNEL NODE USED FOR? on the Fresnel node, in 64 HOW DO I PROPERLY COMBINE TRANSPARENCY AND RAYTRACING? on combining Transparency and Raytracing, and in 98 WHAT'S THE FUZZ ABOUT GAMMA CORRECTION? and 99 GAMMA, TRANSPARENCY AND BLENDS on Gamma Correction itself. It might make your brain twist.

39 What's the Translucence component intended for?

The Translucence component in the PoserSurface material definition is the complete equivalent of the Ambient component, except that Translucence is available through the Advanced interface only and in various cases is not exported or translated to external renderers or materials definitions, while Ambient is available in the Simple and Advanced interfaces and in far more cases is exported or translated decently to external renderers or material definitions. So for those reasons, Ambient is preferred by most people. See 07 WHAT'S THE AMBIENT COMPONENT INTENDED FOR? on the various uses of Ambient.

Why having Translucence then? Well, it's a kind of service for those who want to separate real-life object glow as it appears by led-lights of phosphorescence from faking real-life translucency (as it appears in wax candles, human skin and rice paper walls) by some sort of body glow. They can put the first effect in Ambient, and the second in Translucence. This second effect becomes obsolete though for those who use the real thing instead: sub-surface scattering as recently build into Poser.


Human skin translucency, light scattering through ear tissue.

40 What are the Alternate_Diffuse and Alternate_Specular components intended for?

Those components of the PoserSurface material definition are available in the Advanced interface only. For short: their purpose is to offer alternatives for the regular Diffuse and Specular components, as discussed in 05 WHAT'S THE DIFFUSE COMPONENT INTENDED FOR? and 06 WHAT'S THE SPECULAR / HIGHLIGHT COMPONENT INTENDED FOR?.
And although they are almost equivalent, do note one major difference between Alternate_Diffuse and Alternate_Specular: the first is affected by Transparency, the second is not. Just like regular Diffuse and Specular. When using (either of) them, do no forget to disable the regular Diffuse and/or Specular components. Just setting the Diffuse Value and/or Specular Value to 0.0 will do.

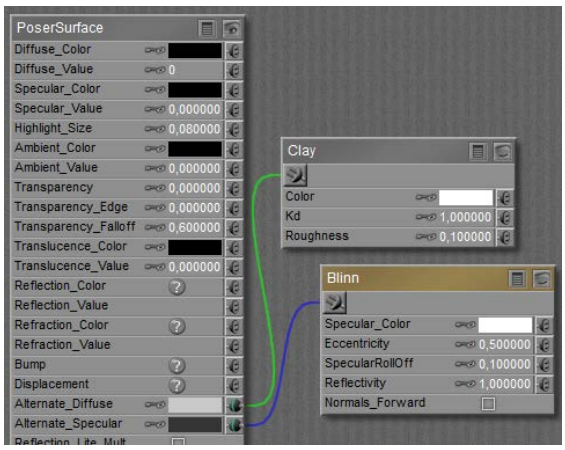
Intermediate

The PoserSurface material definition presents a regular Diffuse and Specular component, which are embedded equivalents of the diffuse and specular material tree nodes. Both nodes regulate the re-emission of received (diffuse / specular) light, and have a basic angular distribution for that build into them. See 30 CAN I GET A (SIMPLIFIED) EXPLANATION ON LAMBERT AND DIFFUSE SHADING? for details on the Lambert method, build into Diffuse.

Although materials represented this way do fine in case of hard plastics, they're not doing very well in case of other – more organic – materials. Therefore other nodes are offered, like **clay** (the usual substitute for diffuse) and **Blinn** (one of the regular alternatives for specular). These nodes need a 'neutral' slot on the PoserSurface spine to be plugged in, and both Alternate ones offer that.

Since both slots are just neutral (*) entries to the PoserSurface spine, I'm free to plug the **clay** node into Alternate Specular, or to plug the **Blinn** node into Alternate Diffuse although the suggestion is to do otherwise for the sake of readability, understandability and maintainability of the surface definition. I can use the **diffuse** and/or **specular** nodes to bring the "good old" angular distributions back into the game, and I can use the "add" nodes to combine diffuse, specular and eventually also ambient, bump, reflection, refraction and other components to build my own equivalent of the PoserSurface spine, and then plug its result into Alternate Diffuse only. Some material-defining tools (e.g. Bagginsbills' MatMagic, SnarlyGribblys' EZMat) work about that way. Transparency however can't be dealt with that way, there is no substituting node for it either.

Nevertheless, the main intention is: disable the regular Diffuse and/or Specular entries, and plug the intended alternatives into the appropriate Alternate entries to get a more realistic material definition.



(*) when nothing is plugged in, the slot will not contribute to the PoserSurface definition in any way. When something is plugged in, the slot offers an additional color filter, and delivers the (filtered) color result into the PoserSurface definition. In there, it gets added up (as discussed in 23 ARE THERE ANY PRACTICAL LIMITATIONS IN COMBINING DIFFUSE, SPECULAR, … IN A SINGLE MATERIAL?) on a color-by-color basis. This means that I might have to pay attention to overlighting conditions.

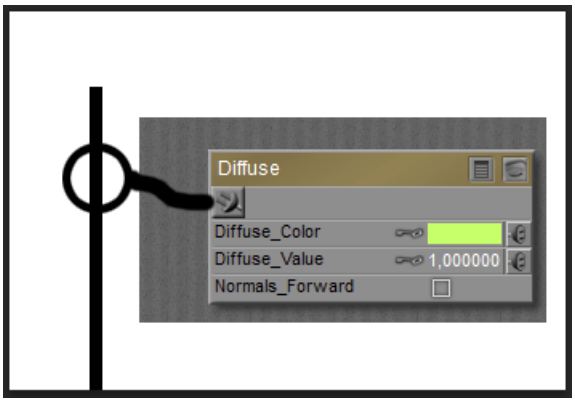
The various nodes plugged into the Alternate slots have to pick up the lighting from the scene themselves. See 25 CAN I GET A BRIEF INTRO ON NODE-TREE BUILDING IN MATERIAL ROOM? on this.

41 What's the difference between the Diffuse component, and the Diffuse node?

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

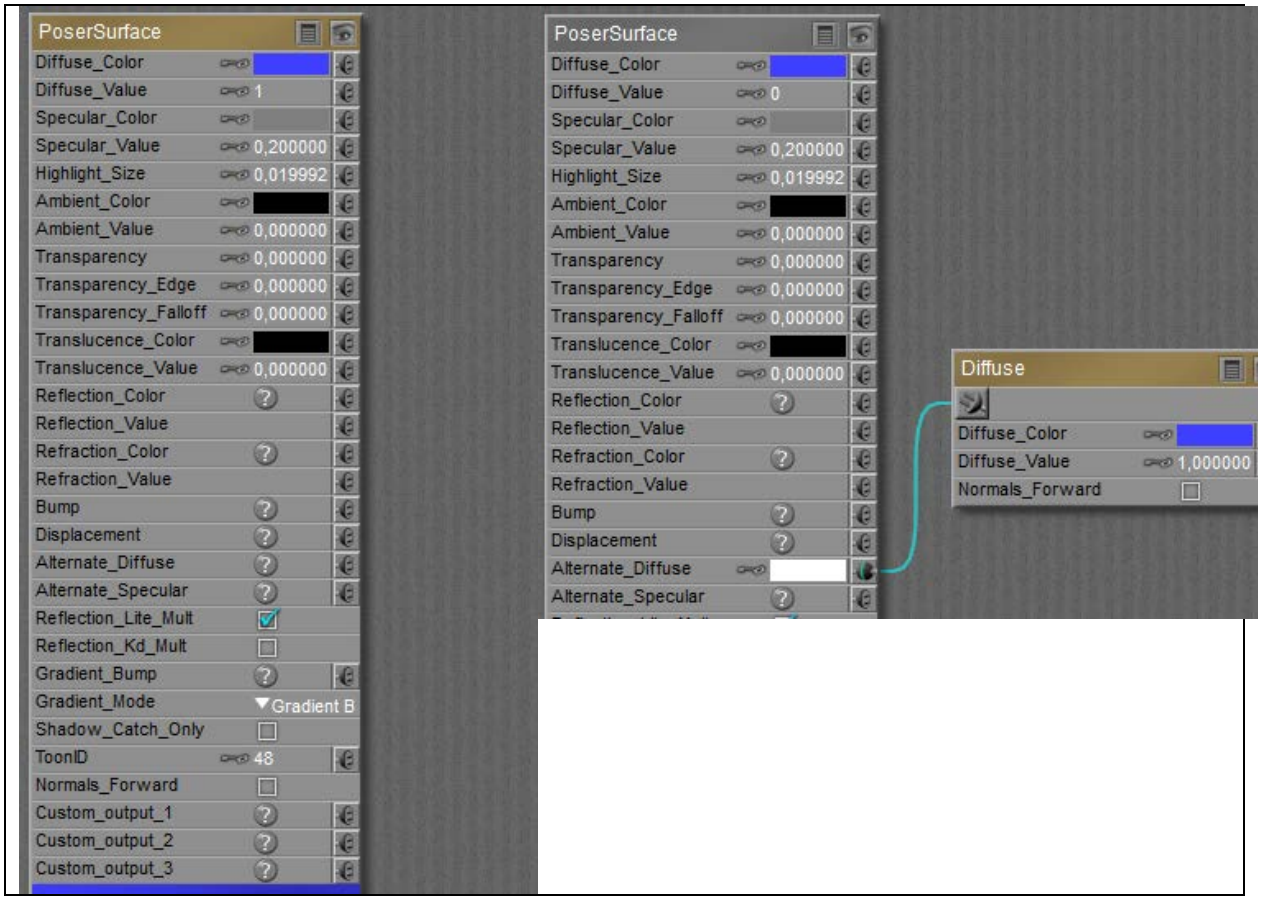
Intermediate

The Diffuse component in the PoserSurface material definition (also see 05 WHAT'S THE DIFFUSE COMPONENT INTENDED FOR?) acts as a neutral entry into the spine ***with the diffuse node already attached to it***. It embeds the material definition as represented by the diffuse node right into the PoserSurface.

is equivalent to


The separate diffuse node itself enables me to apply this definition explicitly in other occasions, like an element in a shading tree plugged into the Alternate Diffuse entry. So, when I disable (blacken out) the Diffuse component and plug the diffuse node into Alternate Diffuse (which is a neutral node by itself), I'll get the same result. See 40 WHAT ARE THE ALTERNATE_DIFFUSE AND ALTERNATE_SPECULAR COMPONENTS INTENDED FOR? for more on this.

The Diffuse node (as well as its embedded variant) responds to all sorts of diffuse light. When IDL lighting is enabled, this not only includes diffuse light from direct (point- spot- infinite) sources but also includes the ambient, reflected, diffused and specular-turned-into-highlight from surrounding objects in the scene.
The response is to re-emit the received light, independent of any angle of incidence for the incoming light rays, according to Lamberts formula (see 30 CAN I GET A (SIMPLIFIED) EXPLANATION ON LAMBERT AND DIFFUSE SHADING? for details). Plus: Color and Value work together as discussed in 24 A POSERSURFACE MATERIAL COMPONENT, OFFERS COLOR AND VALUE. HOW DO THESE WORK TOGETHER?.
This Lambert representation of material is fast to calculate, but falls short for a neat illusion of especially porous stuff like cloth, wood, plants, skin, brick, etc. Those materials might be better off with the Clay node, discussed in 42 WHAT'S THE CLAY NODE USED FOR?. Additional alternatives are offered by the various Scatter nodes, see 58 WHAT'S THE DIFFERENCE BETWEEN ALL THOSE SCATTER NODES? for details.



Left: Diffuse Color, Value (set to 1.0) and Normals_Forward embedded in the PoserSurface definition.
Right: the embedded diffuse is nullified (Value=0), the node is plugged into Alt Diffuse.
Like in the PoserSurface itself, the Diffuse node has Color, Value and Normals_Forward for parameters.

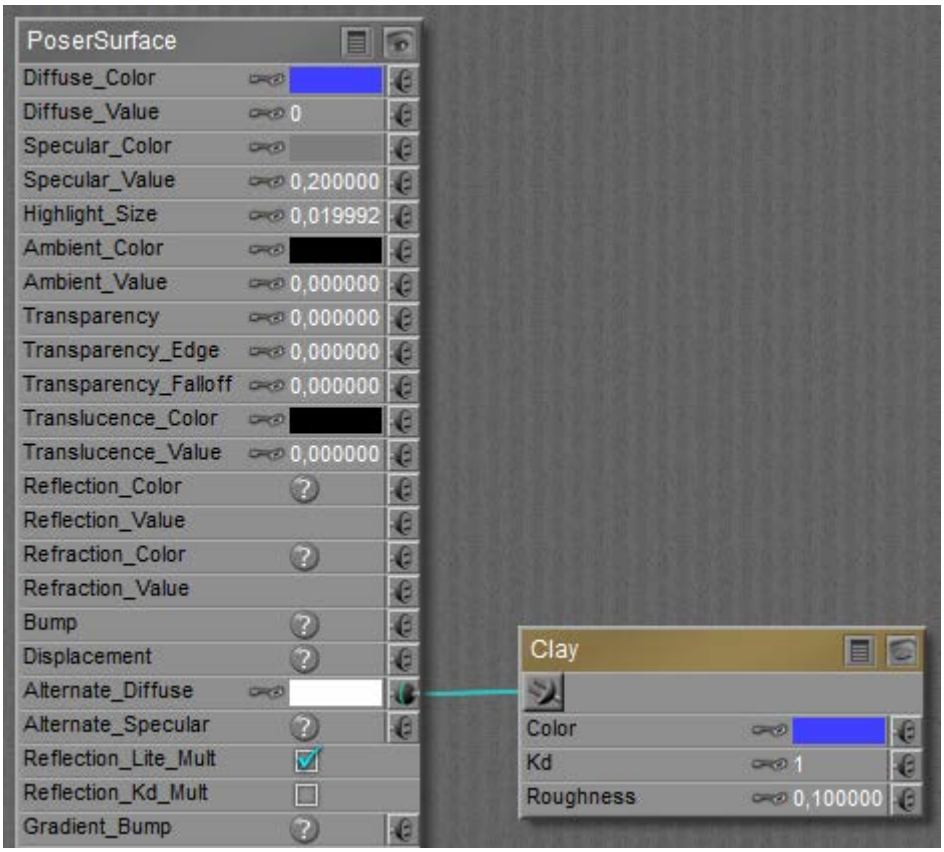
42 What's the Clay node used for?

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

Intermediate

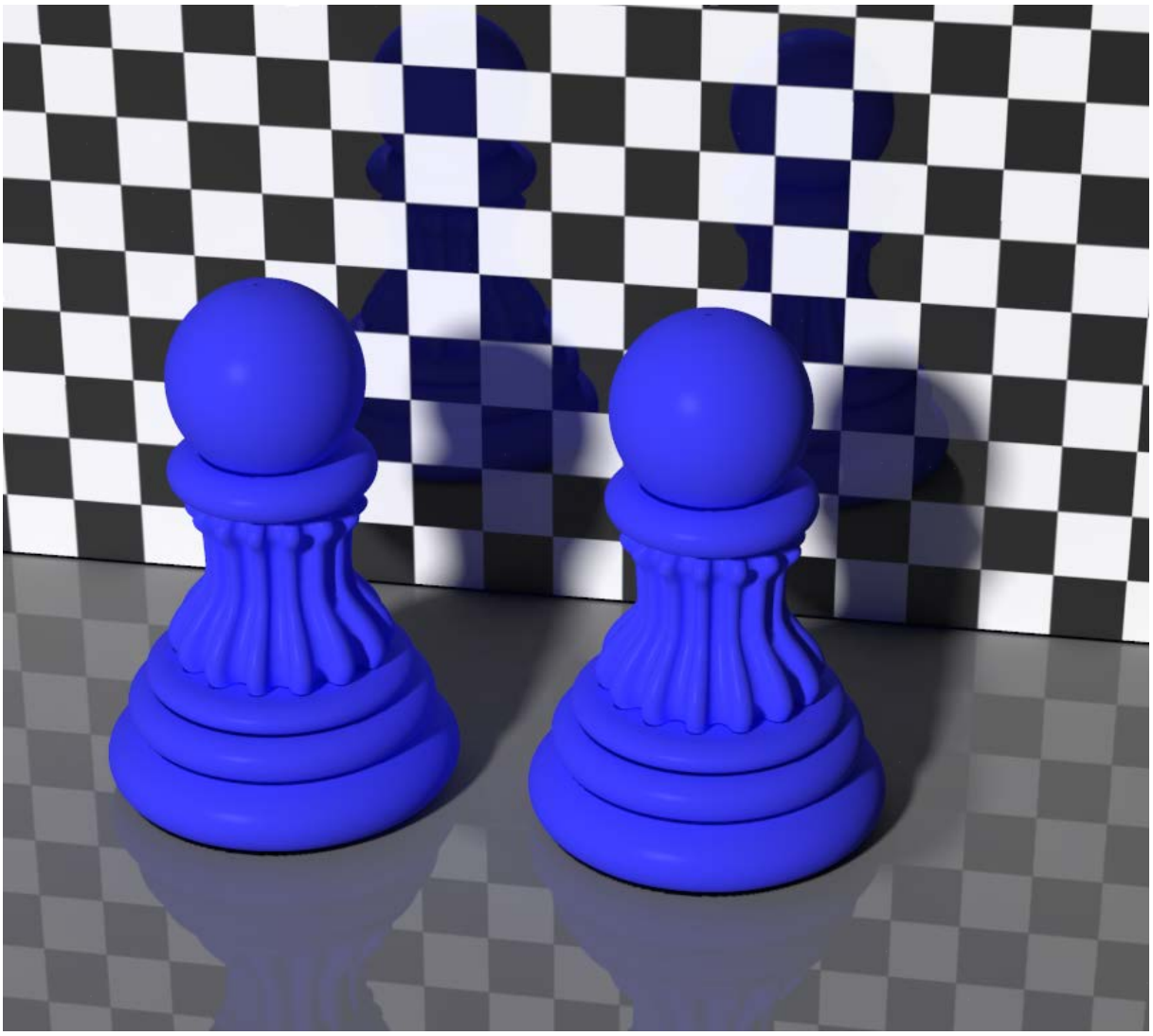
The default Diffuse property of materials offers a reasonable representation for smooth, hard materials. But for materials of a more porous nature the Clay node is a far better alternative. Think of skin, cloth, wood and all other kinds of natural stuff, brick, stone, non-glazed pottery, rubberish soft plastics, etcetera.
Note though that some materials, especially skin and soft plastics or rubbers, have got some advanced alternatives in recent Poser versions. These are the various 'scatter' nodes, discussed in 58 WHAT'S THE DIFFERENCE BETWEEN ALL THOSE SCATTER NODES? and related articles. As can be expected, such alternatives do require more parameters, and more render time.

Anyway, like the Diffuse node (see 41 WHAT'S THE DIFFERENCE BETWEEN THE DIFFUSE COMPONENT, AND THE DIFFUSE NODE?) the Clay node is meant to be plugged into the Alternate Diffuse slot (Advanced interface only), while having the Diffuse portion of the PoserSurface definition be zeroed out:

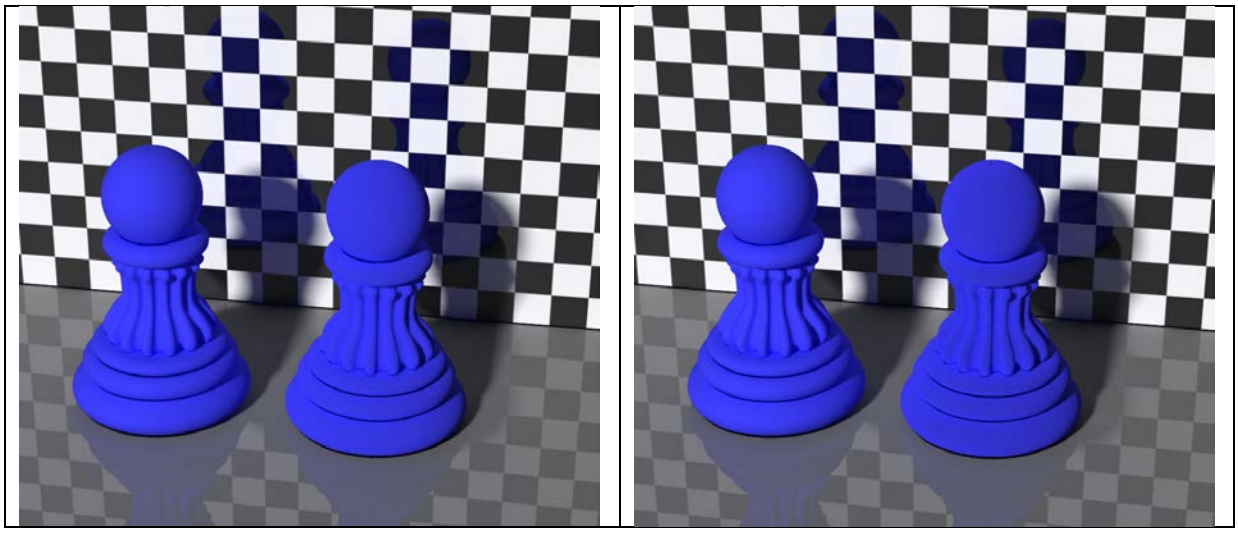


The Clay node offers **Color** and **Kd** for parameters which have the same meaning as Color and Value for Diffuse. The new kid on the block is: **Roughness**.

A low value makes the surface smooth, and **the default 0.1 turns the Clay into a Diffuse equivalent**. Lower values make no significant difference, but higher values (up to 1.0) do.

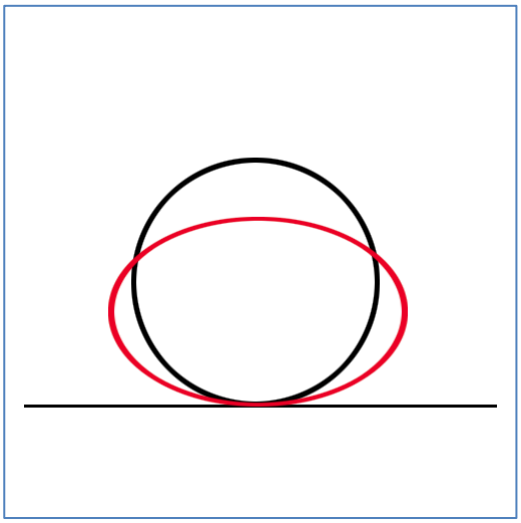


The scene above compares regular Diffuse (left) with Clay response (right) having Roughness set to 0.1. Specularity is at default value. I hardly see a difference: Diffuse is a special case of Clay.



Both two scenes above have specularity turned OFF for demonstration purposes, and use the Clay node for diffuse response. Roughness is set to 0.1 (both left pawns, representing regular Diffuse) versus 0.5 (left image, right pawn) and versus 1.0 (right image, right pawn) respectively.

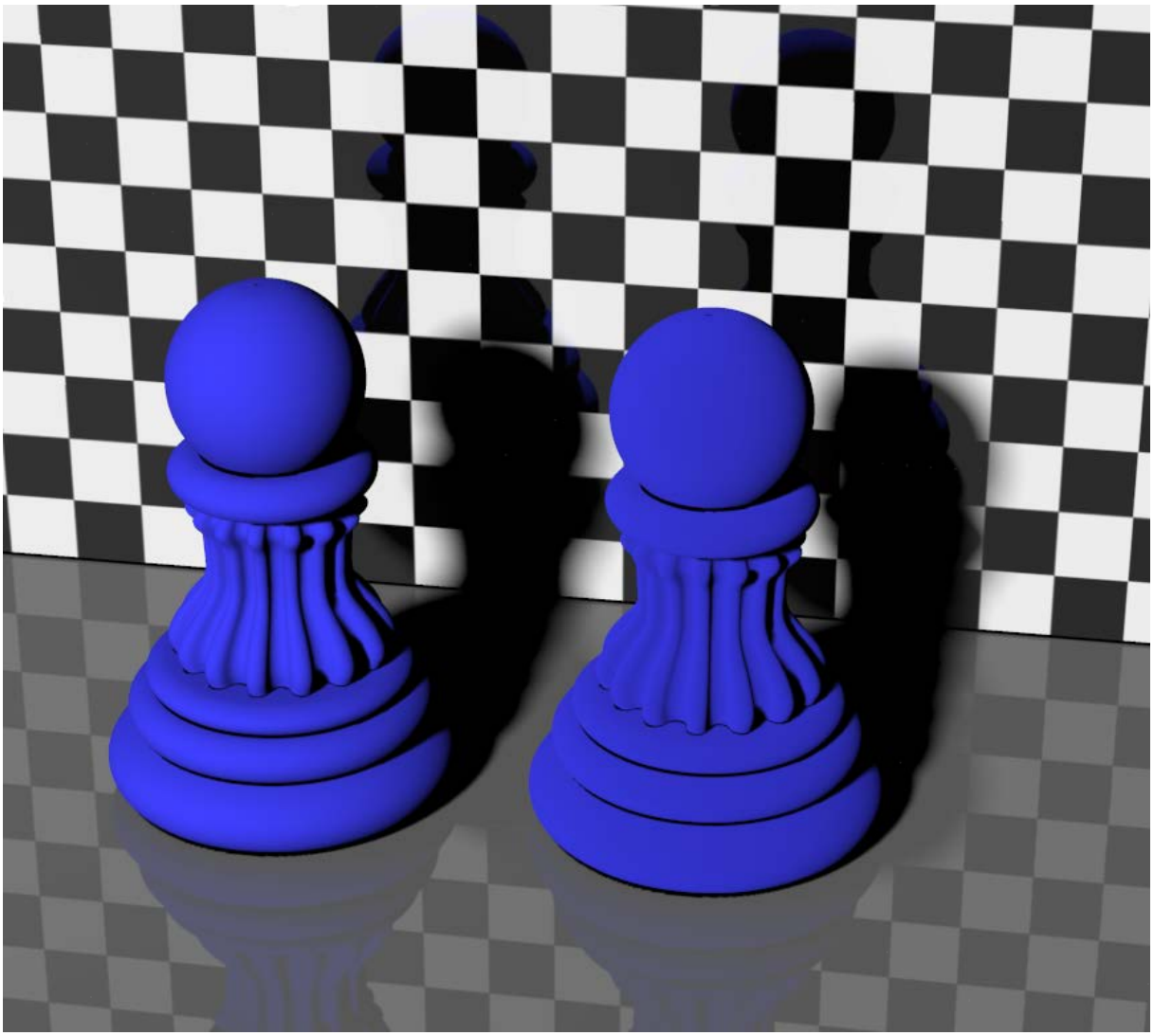
Maybe you've already noticed that the porous material, using the Clay node, re-emits less light in the forward direction (it's slightly darker *) but experiences a reduced falloff near the edges (the higher the roughness, the less darkening at the sides reveal the shape of the object). The first is the result of the pores absorbing more light, the second is because the pores make the surface more irregular. The higher the Roughness, the more obvious this effect.



The graph shows this in another way.
The black circle represents the light intensity from classic Lambert diffusion. Lots of light perpendicular from the surface, no light parallel to it, and a cosine fall-off for angles in between.
The red curve represents the Clay effect: less light perpendicular compared to diffuse, but less intensity-falloff at the edges.

Surfaces with small pores (soft plastics, skin, plant leaves) will benefit from low roughness values, say 0.2 to 0.4 while surfaces with noticeable pores (brick, pottery) will benefit from larger values (0.7 to 0.9).

(*) since IDL light scenes will have the light coming from about all directions, the effect on light distribution – as being darker in the forward direction – will be less noticeable. Rendering with direct light only makes this Clay effect far stronger.

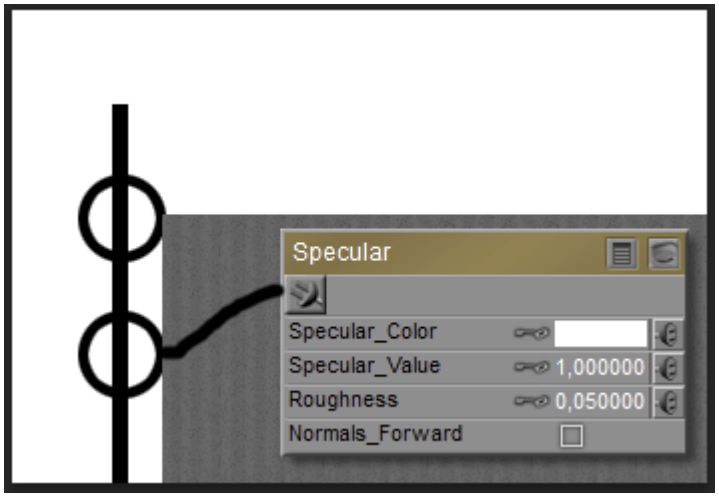


In the image above (with no specularity), the left pawn represents regular diffusion or clay with Roughness set to 0.1 while the right pawn represents clay with Roughness 1.0; just like the image before. But now, IDL lighting is OFF. Now the difference between them is much larger. This implies that when I render my scene under IDL conditions, it might become questionable whether it pays off to implement the clay node instead of regular diffuse.

43 What's the difference between the Specular component, and the Specular node?

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

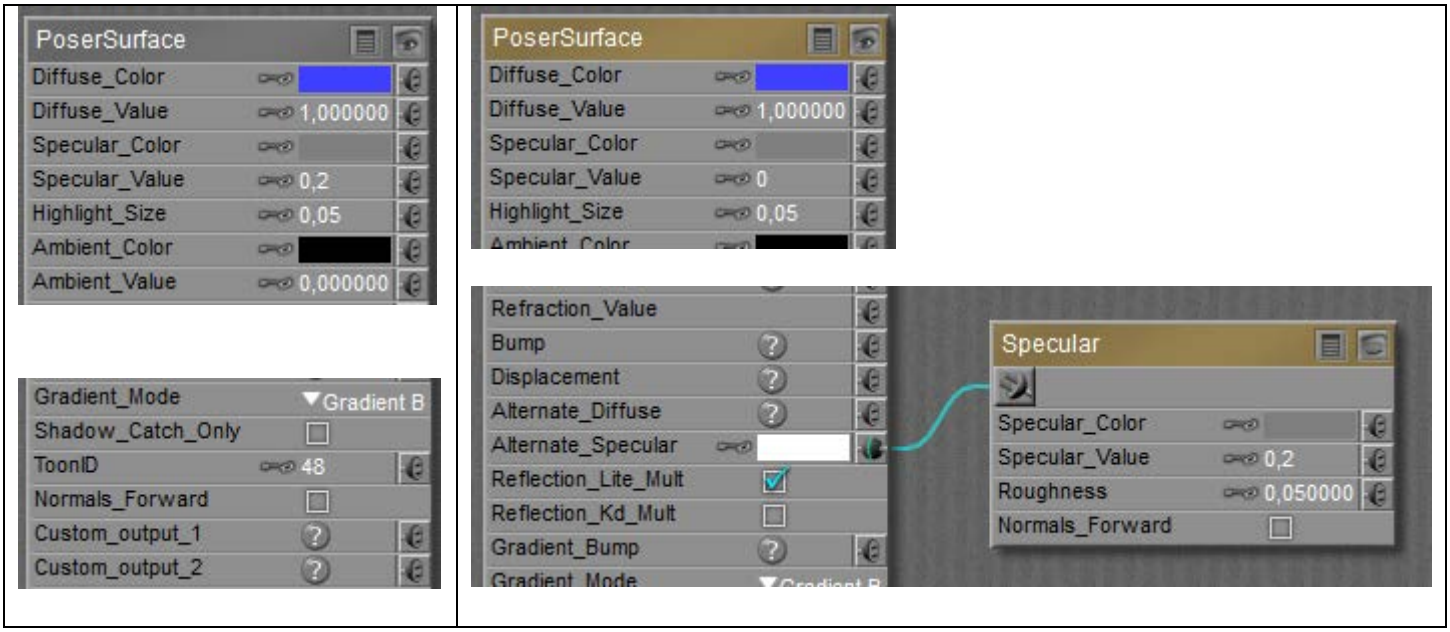
Intermediate



The Specular component in the PoserSurface material definition (also see 06 WHAT'S THE SPECULAR / HIGHLIGHT COMPONENT INTENDED FOR?) acts as a neutral entry into the spine with the specular node already attached to it. This embeds the material definition as represented by the specular node into the PoserSurface.

The specular node itself enables me to apply this definition explicitly in other occasions, like an element in a shading tree plugged into the Alternate Specular entry. So, when I disable (blacken out) the Specular component and plug the specular node into Alternate Specular (which is a neutral node by itself), I'll get the same result. See 40 WHAT ARE THE ALTERNATE_DIFFUSE AND ALTERNATE_SPECULAR COMPONENTS INTENDED FOR? for more on this.

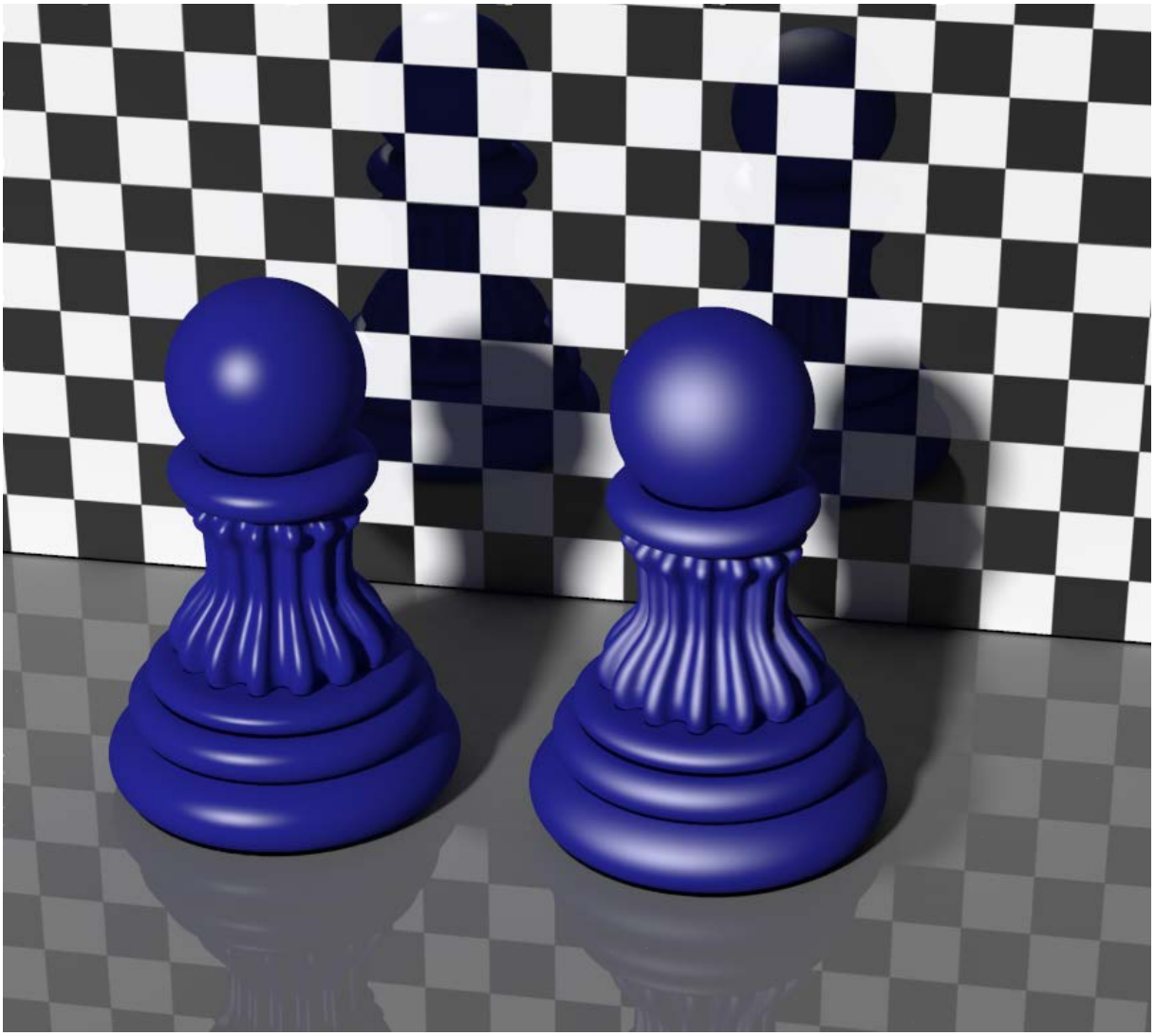
On Specularity node parameters



Left: Specular Color, Value (set to 0.2), Highlight Size (0.05) and Normals_Forward embedded in the PoserSurface definition.
Right: the embedded diffuse is nullified (Value=0), the node is plugged into Alt Specular.

Like the PoserSurface component itself, the Specular node has **Color**, **Value** and **Normals_Forward** for parameters. **Highlight Size** as used in the PoserSurface and **Roughness** as used in the node refer to the same, just different words for the same concept. The same value will render the same result. The smoother the surface, the smaller the roughness value, the sharper, stronger and smaller the

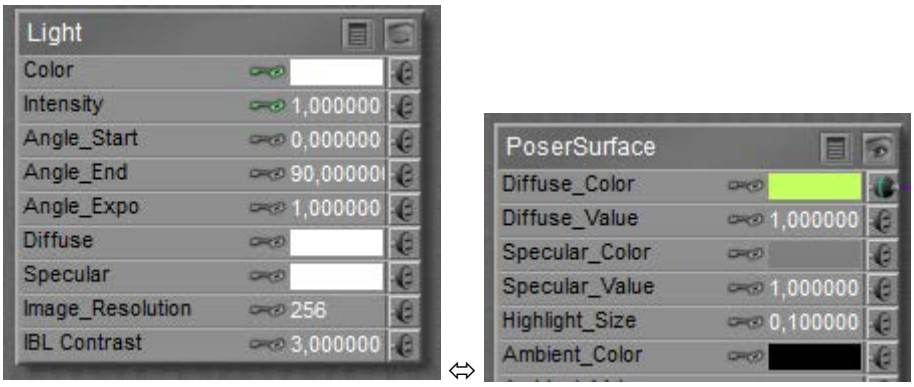
highlight will be.
Note however that Highlight Size and Roughness come with different default values.



Left: Highlight size 0.05, right: highlight size 0.25.

On highlights and specularity

The Specular node (as well as its embedded variant, as well as all its alternatives) responds to specular light, which refers to specular light from direct (point- spot- infinite) sources only.



Nor IBL direct lights, nor IDL lighting setups do supply any specular lighting, and therefore cannot create highlights in the render.

Note that any form of specular lighting, handled by the specular nodes or any of its alternatives (Blinn, …) actually represents the reflection of the direct lights in the scene themselves onto the object surface. The reflective properties of a PoserSurface only bounce the (diffuse) light from surrounding objects in the scene, not from lights. As reflected light itself is considered diffuse and not specular, reflections of lights in mirrors, metal objects or anything else will **not** create highlights on other objects by themselves.

Also note – with respect to IDL and reflections – that the resulting highlights from specular lighting will be considered diffuse light themselves. So they will contribute in IDL lighting the scene, they will occur in reflections, and again - they will **not** create highlights on other objects.

So, the Specular node (as well as its embedded variant, as well as all its alternatives) responds to specular light. The response is to re-emit the received light, independent of any angle of incidence for the incoming light rays, in the form of highlights on the surface. Plus: Color and Value work together as discussed in 24 A PoserSurface material component, offers Color and Value. How do these work together?.

Variations

As Poser is not raytracing to derive the specular highlights and as blurred highlights are not properly described by the sharp “angle in equals angle out” concept as is the case with decent mirrors, one has to define another rule for re-distributing the bounced light. And though the specular principle calculates and renders pretty fast, it produces quite unrealistic results and makes everything look like a bad representation of hard plastic.
This is why Poser offers alternatives, like Blinn (44 What’s the Blinn node used for?) or Anisotropic (46 What’s the Anisotropic node used for?) for microscopically grooved surfaces.

> *A decent run-down on various specular reflection models is here:*
> *http://en.wikipedia.org/wiki/Specular_highlight*
>
> *There is a confusing nomenclature in Poser regarding "Phong". There is a Phong node, but it does not implement what is now called the Phong Reflection Model. It implements something that Mr. Phong used in the 70's which has now been discarded because computers are so much faster - we can use better models.*
> *With some subtle differences and variation, we can make these connections:*
>
> - *The "Phong distribution" explained in that Wikipedia article is what in Poser is called the Specular node.*
> - *The "Ward anisotropic distribution" is our Anisotropic node.*
> - *The "Cook-Torrance model" is our Blinn node. (As far as I can tell, Blinn took the Cook-Torrance model and improved upon it - resulting in the Blinn-Phong reflection model. Or something like that.)*
>
> *The Glossy node is none of these. It is a clever little cheat to deal with the fact that our CG lights are effectively point sources. A point light has no dimension, and a perfect glossy specular reflection of it would have no dimension, either. The Glossy node ignores that little fact. It simulates what you'd get from a glossy surface if the light actually had some finite width to it.*
>
> *by Bagginsbill, Renderosity forum, 2009*

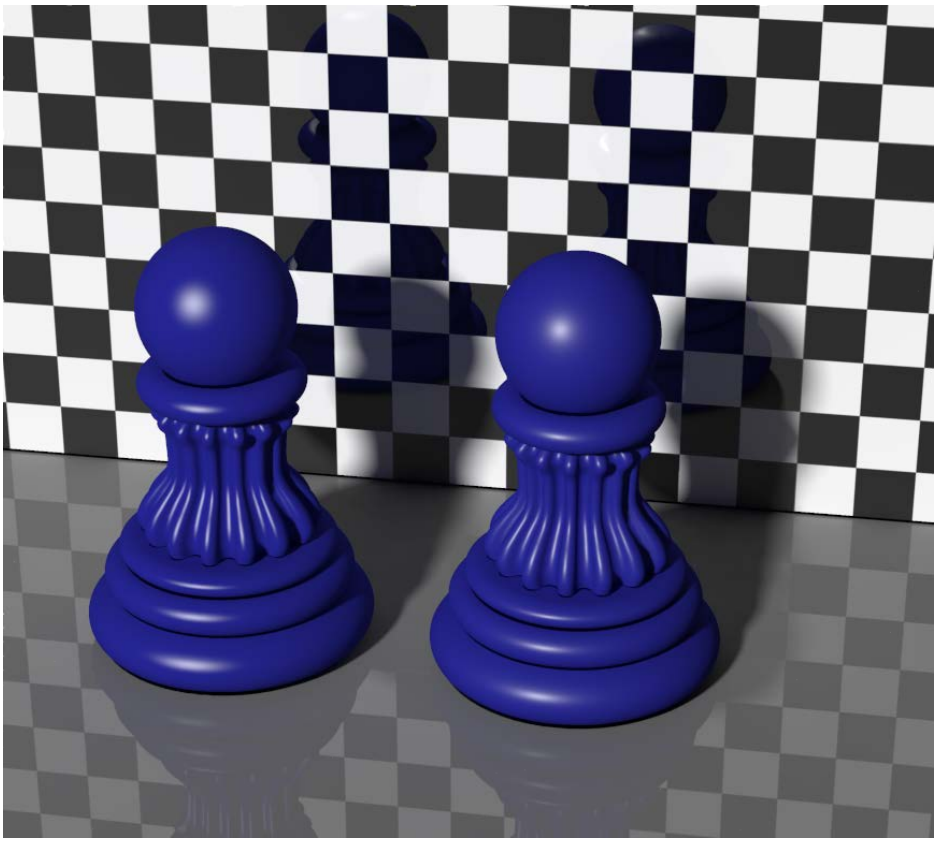
44 What's the Blinn node used for?

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

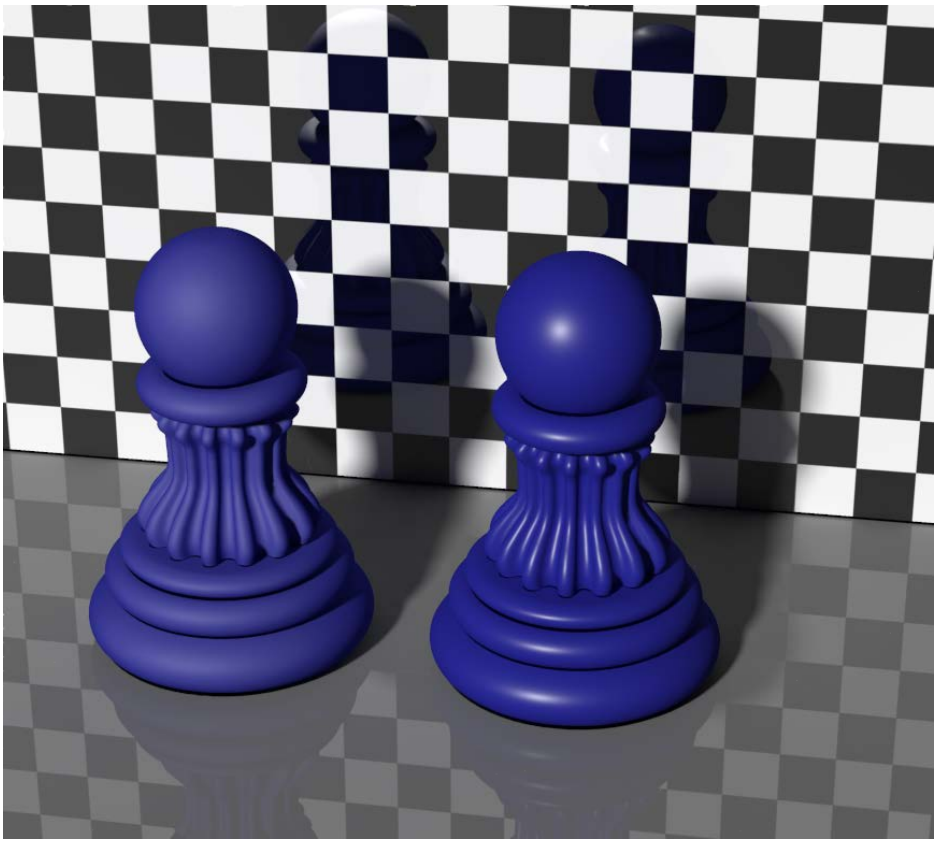
Intermediate

All nodes in the Specular group respond to specular light, which refers to specular light from direct (point- spot- infinite) sources only. The regular 'specular' node itself, either separate or embedded in the PoserSurface definition directly (see 43 WHAT'S THE DIFFERENCE BETWEEN THE SPECULAR COMPONENT, AND THE SPECULAR NODE?) is fast to calculate, but falls short for a neat illusion of about any real stuff except hard plastic.

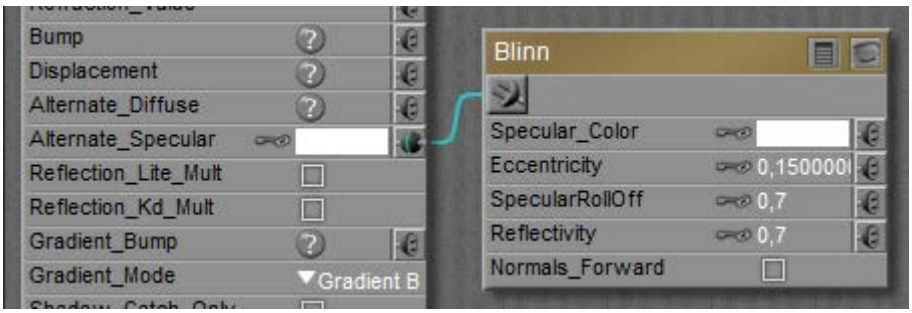
This is why Poser offers alternatives, like **Blinn**, or Anisotropic (46 WHAT'S THE ANISOTROPIC NODE USED FOR?) for microscopically grooved surfaces. Effectively Blinn relates to Specular as Clay relates to Diffuse (see 42 WHAT'S THE CLAY NODE USED FOR?): it's the improved version for generic purposes.
In other words: when I do need Clay (or other improvements, like Scatter) to represent surface roughness, then I do need Blinn for the according specularity. Just nullify or black out the regular Specular slot in PoserSurface, and plug the Lighting > Specular > Blinn node into Alternate_Specular.



Above: the left pawn has regular specularity (highlight size 0.05) while the right pawn has Blinn. Parameters at 0.15 / 0.7 / 0.7 make a sort of matching specular strength, but note the strong center of the highlight in Blinn.



Above: the right pawn has parameters at 0.15 / 0.7 / 0.7 as before, now the left pawn at 0.5 / 0.1 / 1.0 representing a porous, soft, diffusing surface. There still is some resemblance of highlight, though.
So the question is: what do the parameters mean, and how do they relate to regular specularity?



Well, **Specular_Color** and **Normals_Forward** are the same, while **Reflectivity** sort of matches Specular_Value. It serves the same purpose at least, though I need different values for a similar effect. When Eccentricity is low (< 0.1) the highlight is sharp like regular specular, and when it's 0.2 or up the represented stuff looks mildly porous like the roughness of clay. Values will be quite different though, and differ from the regular Highlight_Size as well, so it's a matter of experimenting what works best in my scene. The Specular RollOff also affects the impression of porosity.

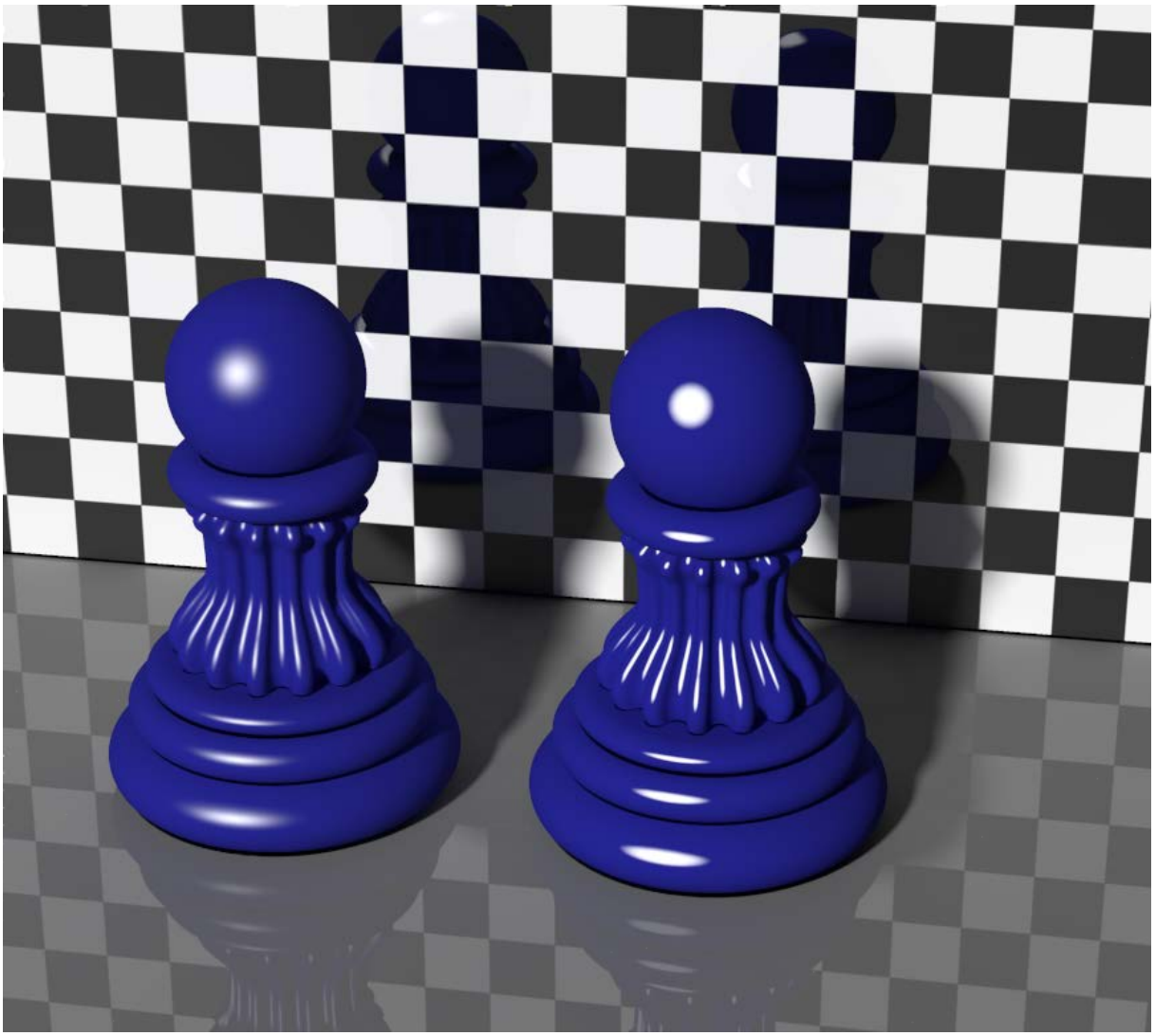
45 What's the Specular > Glossy node used for?

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

Intermediate

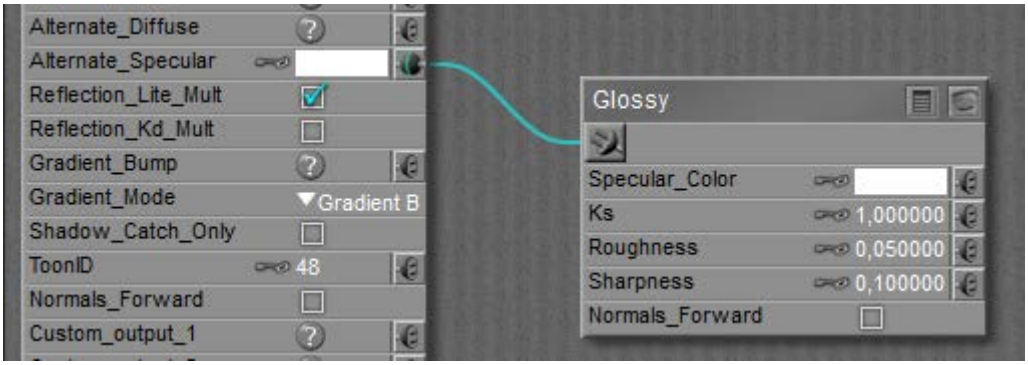
All nodes in the Specular group respond to specular light, which refers to specular light from direct (point- spot- infinite) sources only. The regular 'specular' node itself, either separate or embedded in the PoserSurface definition directly (see 43 WHAT'S THE DIFFERENCE BETWEEN THE SPECULAR COMPONENT, AND THE SPECULAR NODE?) is fast to calculate, but falls short for a neat illusion of about any real stuff except hard plastic.

This is why Poser offers alternatives, like Blinn (44 WHAT'S THE BLINN NODE USED FOR?), or Anisotropic (46 WHAT'S THE ANISOTROPIC NODE USED FOR?) for microscopically grooved surfaces. In this series, Glossy can be considered as somewhat outdated, and available for compatibility reasons mainly. Glossy was introduced as an improvement over the classic Phong specularity, and added the suggestion that light sources have a physical size (while in Poser, they have not). So Glossy produces quite intense midsections in the round highlights; nice for point- and spotlights but incorrect for infinite lights unless these represent a sun.



Above: the left pawn shows regular specular, right pawn shows glossy, both at their default settings.

Glossy makes sharp highlights with eventually a softening edge. In the image above the highlights are about equal size. The left pawn is regular specular, color white, value 1.0 and highlight size 0.05. The right pawn has the same color, the **Ks** value matches Specular Value and **Roughness** matches Highlight size, while **Normals_Forward** matches the same option in PoserSurface.



This leaves **Sharpness** as an extra parameter. It makes the width of the blur at the edges of the highlight. High values – up to 1.0 – produce sharp edges but frankly, values over 0.1 start to make quite an unnatural impression.

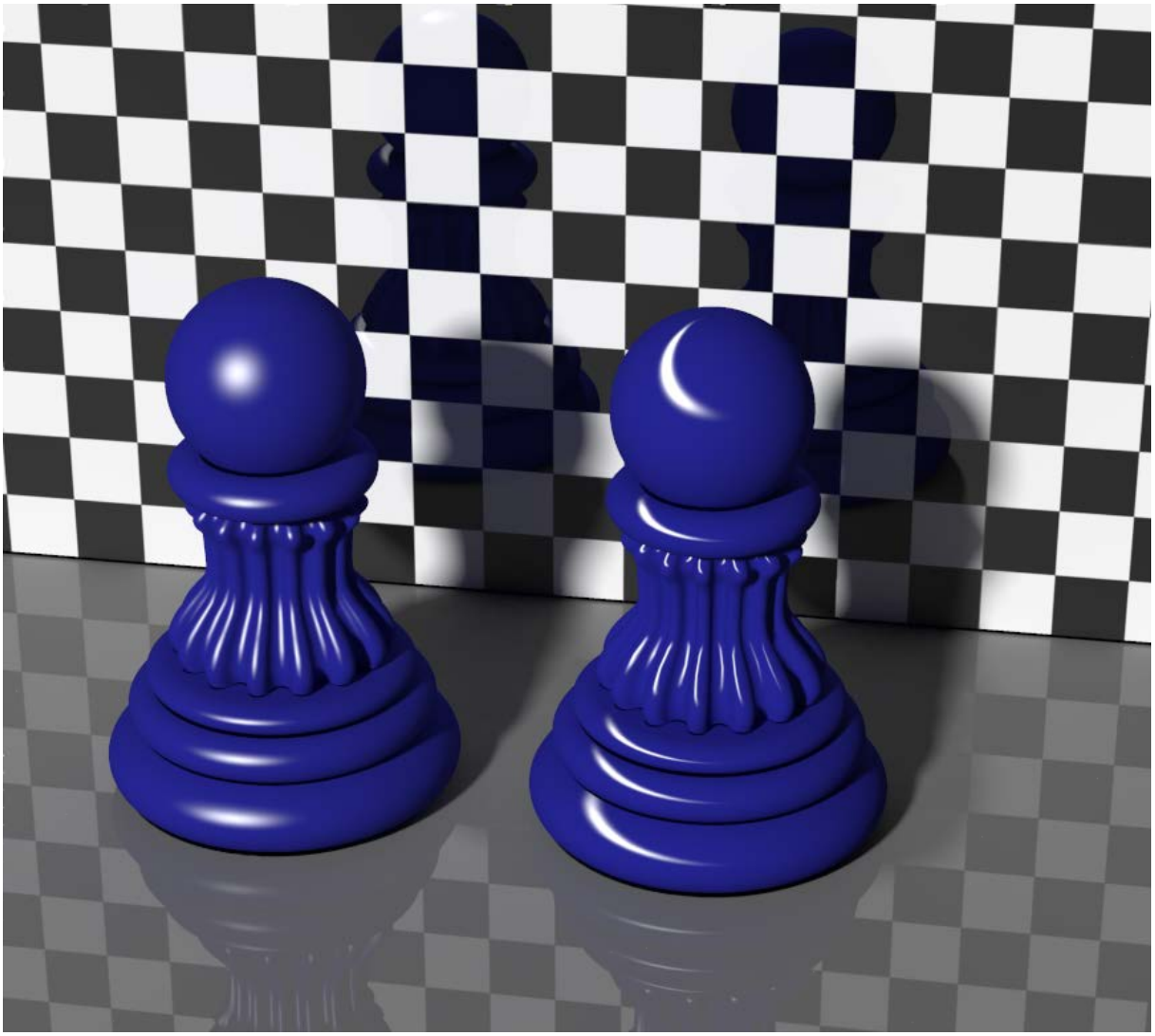
46 What's the Anisotropic node used for?

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

Intermediate

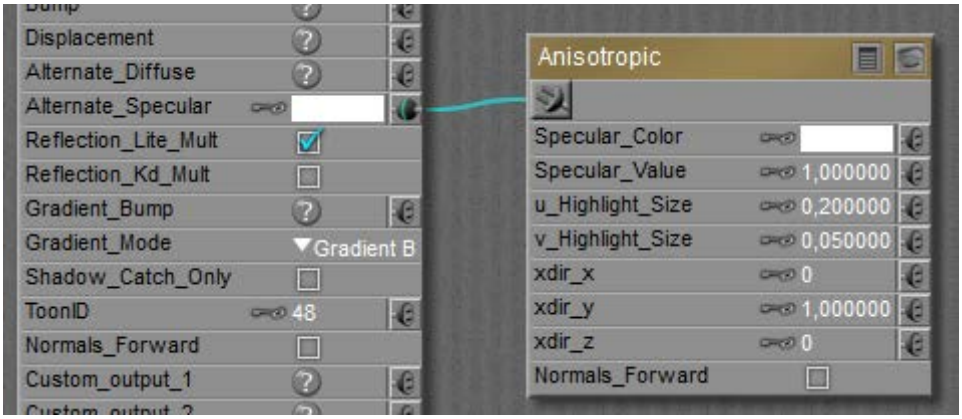
All nodes in the Specular group respond to specular light, which refers to specular light from direct (point- spot- infinite) sources only. The regular 'specular' node itself, either separate or embedded in the PoserSurface definition directly (see 43 WHAT'S THE DIFFERENCE BETWEEN THE SPECULAR COMPONENT, AND THE SPECULAR NODE?) is fast to calculate, but falls short for a neat illusion of about any real stuff except hard plastic.

This is why Poser offers alternatives, like Blinn (44 WHAT'S THE BLINN NODE USED FOR?). These specular effects make round highlights, similar in all directions. Iso-tropic (iso=same), so to say. **An-isotropic** (=non-iso-tropic) therefore introduces some directionality. This is meant for brushed metals, and for other materials with a microscopically 'grooved' surface, like brushed metals and the good old vinyl music disks.



The left pawn shows regular specularity, the right pawn shows anisotropic.

In the parameters, we recognize the **Color**, **Value** and **Highlight** size and in the regular **Specular** setup, and those do mean the same thing. The highlight size however is distinguished for U and V direction, that's relative to the object surface layout.



The node also offers the possibility to distinguish for X, Y, Z directions but up till now different values have not produced different results.

47 What's the Diffuse > probeLight node used for?

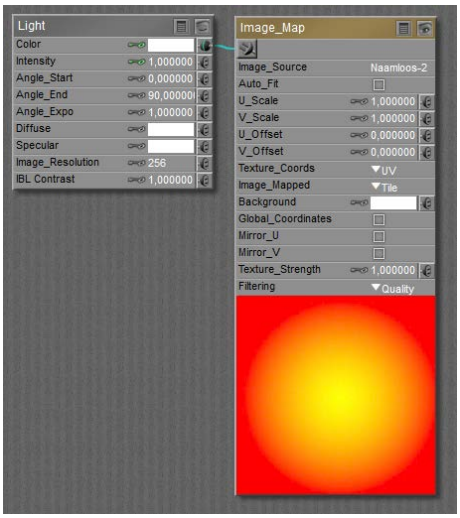
Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

Intermediate

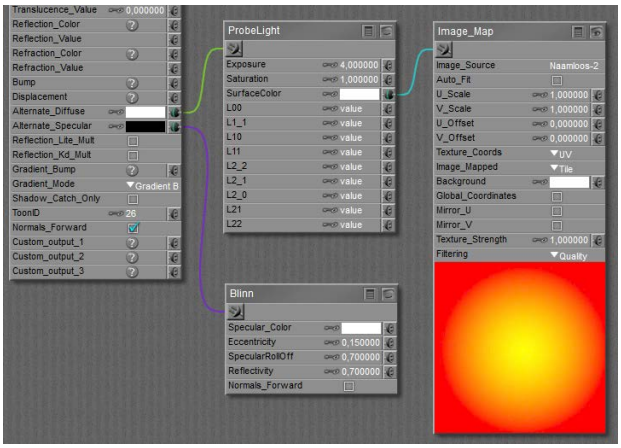
The probeLight node is one for very specific purposes, related to IBL (image based) scene lighting, as discussed in 97 How are Image Based Lighting, probeLight objects and IDL Sky Domes related?. The idea is as follows:

In the industry, when images are shot for IBL scene lighting (see http://my.smithmicro.com/tutorials/1850.html for example), shiny metal balls or 'probes' are used. Photographs of those make good candidates for the images attached to IBL lamps, lighting the scenes.

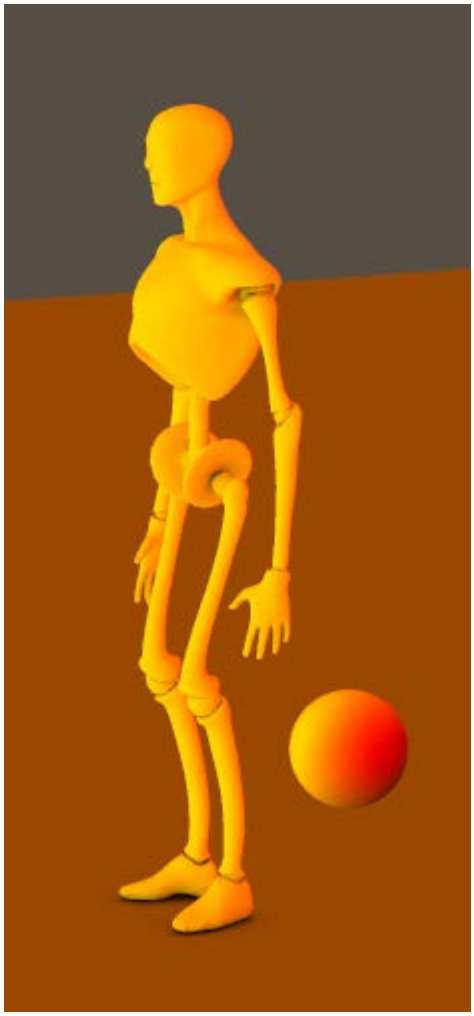
But what if I want such a probe itself in my scene? Then I take a ball, and use the probeLight node to wrap the same IBL image around it, the same way as it's mapped in the IBL lighting itself.



relates to



I even can wrap the image around a point light, but do note that in such a case the light ray will be directed in a direction opposite to the IBL lighting. Someone might get disoriented then. So, for balls as well as point lights, I should take care of their orientation (that is, their yRotate transform value). Some need a 90° or 180° twist.



Andy is lit by the IBL light shown above, the ball also has the probeLight material attached to it. Andy looks towards the scene Front.

Now, what are the probeLight parameters about?

Next to the **SurfaceColor** which just filters its (image map) input like usually for such color swatches, the **Exposure** and **Saturation** adjust the image at hand, so I can match is with the lighting levels from the IBL lamp in the scene.
All the other, L00...L22 parameters, are (undocumented) vectors which all squash and stretch the image in various directions, when being mapped onto the object at hand. This is meant to adjust for imperfections. Not in the object, which usually is a perfect round sphere in virtual 3D space. But for imperfections in the image, which might not have been shot from such a perfect sphere in real space.

Note: when using a ball to serve as a probe in the scene, the lowres Poser ball is recommended over the hires one. The latter appears hard to orient easily, while the first maps correctly immediately.

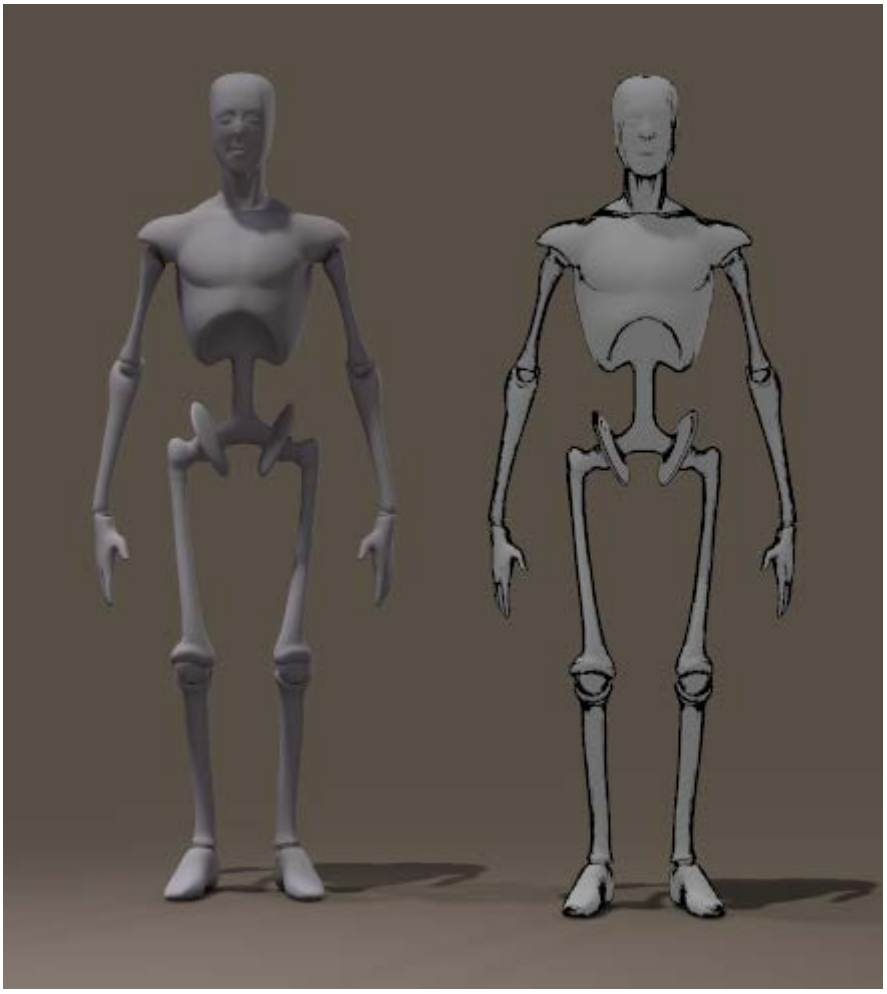
48 What's the Diffuse > Toon node used for?

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

Intermediate

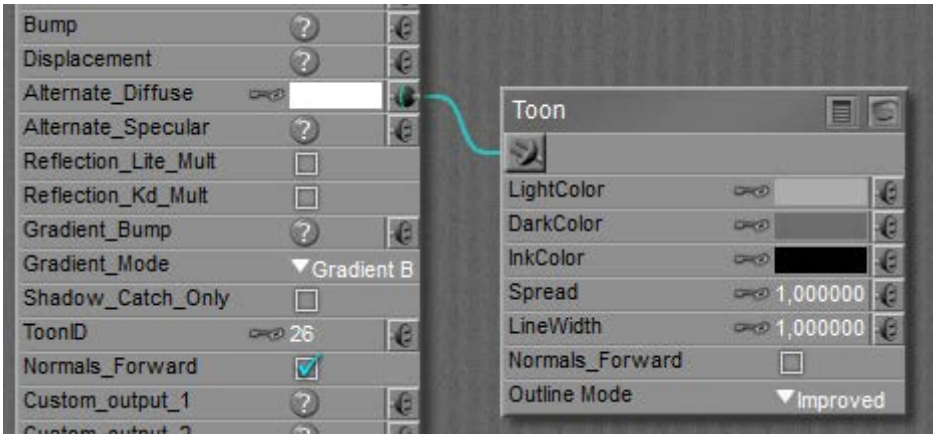
In cartoon style, images have their colors reduced and have the contours of object and object elements lined out. Poser once even had a Toon Renderer to support this but it dealt only with images as a whole. That effect can be obtained with about any reasonable image processing tool in post, nowadays.

So Poser now offers a Toon node, which translates a regular diffuse light distribution pattern to a reduced color scheme, plus the required line-out (inking) as well. This enables me to mix regular, photo real and cartoon elements in one scene, or even animation.



Diffuse Andy (left) and Cartoon Andy (right).

The node parameters are quite obvious:



The surface fades from **LightColor** to **BlackColor** like a regular surface would fade from bright to dark, but in a toon surface the pace is determined by the **Spread** value. If 1.0 then LightColor is very dominant, and I'll see various surface and shape details. If low (say 0.1) DarkColor is dominantly present, and surface and shape details will hardly show.

The object elements are outlined in **InkColor**, with a specified **LineWidth** in a way determined by **Outline Mode**. Just experiment a bit to find out what settings fit your needs best.

Normals_Forward is like the similar switch in PoserSurface and in any node from the diffuse or specular group: when the surface elements have their normal (dictated by the vertex order in the mesh definition) in the wrong direction, light bounces might be calculated erroneously, and dark spots on the surface will result. Check the box to make Poser fix the issue.

49 What's the Specular > Phong node used for?

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

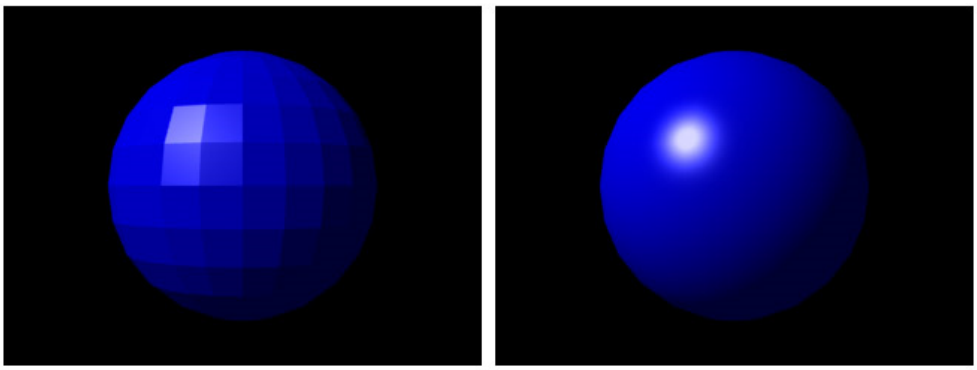
Intermediate

All nodes in the Specular group respond to specular light, which refers to specular light from direct (point- spot- infinite) sources only. The regular 'specular' node itself, either separate or embedded in the PoserSurface definition directly (SEE 43 WHAT'S THE DIFFERENCE BETWEEN THE SPECULAR COMPONENT, AND THE SPECULAR NODE?) is fast to calculate, but falls short for a neat illusion of about any real stuff except hard plastic.

This is why Poser offers alternatives, like Blinn (44 WHAT'S THE BLINN NODE USED FOR?), or Anisotropic (46 WHAT'S THE ANISOTROPIC NODE USED FOR?) for microscopically grooved surfaces. Phong is considered completely outdated.

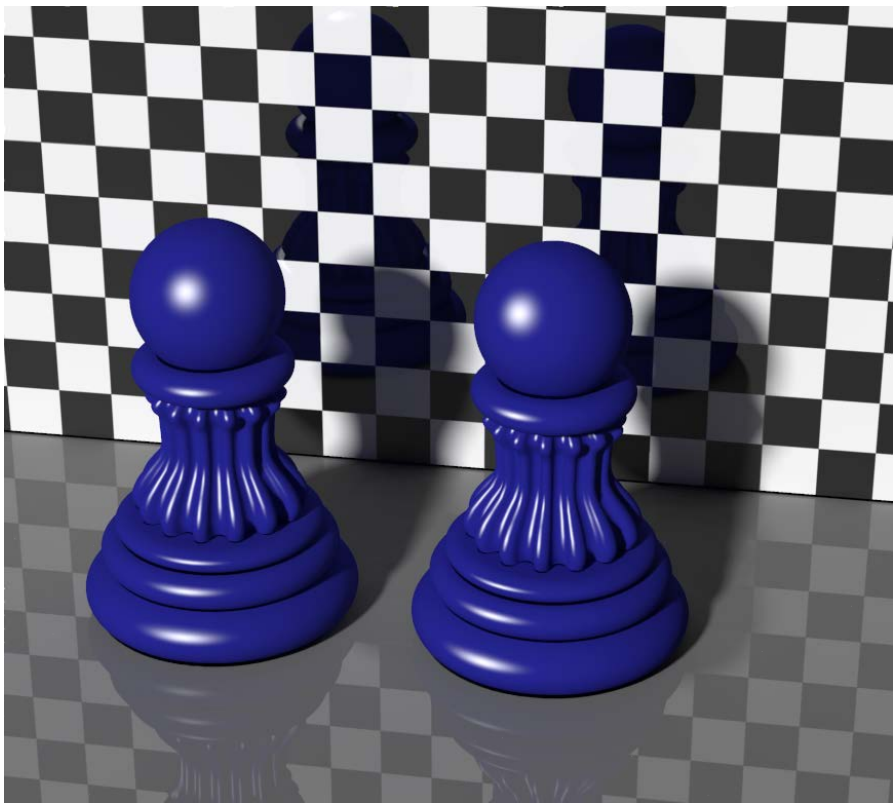
Some background

B.T. Phong introduced (in 1973) a method to make neat highlights and smooth surfaces out of a surface which consisted of small flat elements, as our 3D objects do.

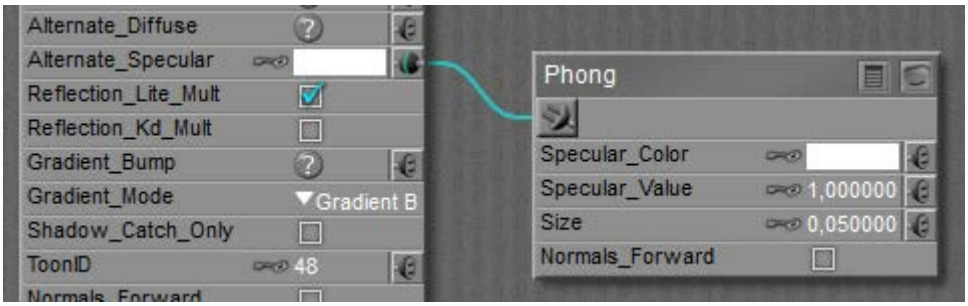


FLAT SHADING PHONG SHADING

As his method was improved by Blinn, and as the results in Poser hardly differ from the regular specularity (left pawn, vs right pawn showing Phong):



the method – and the node – can be considered outdated. The parameters have the same meaning as the Specular ones.



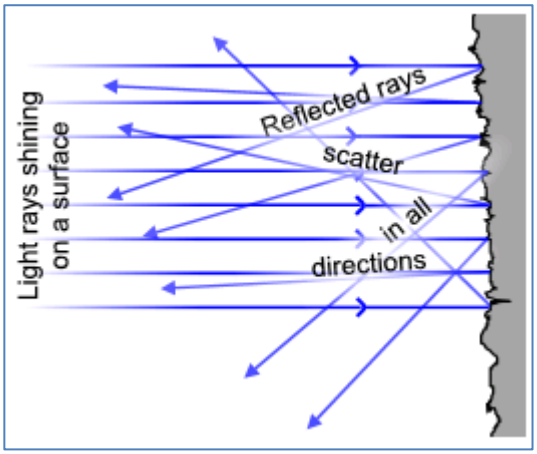
50 What's the Specular > Ks Microfacet node used for?

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

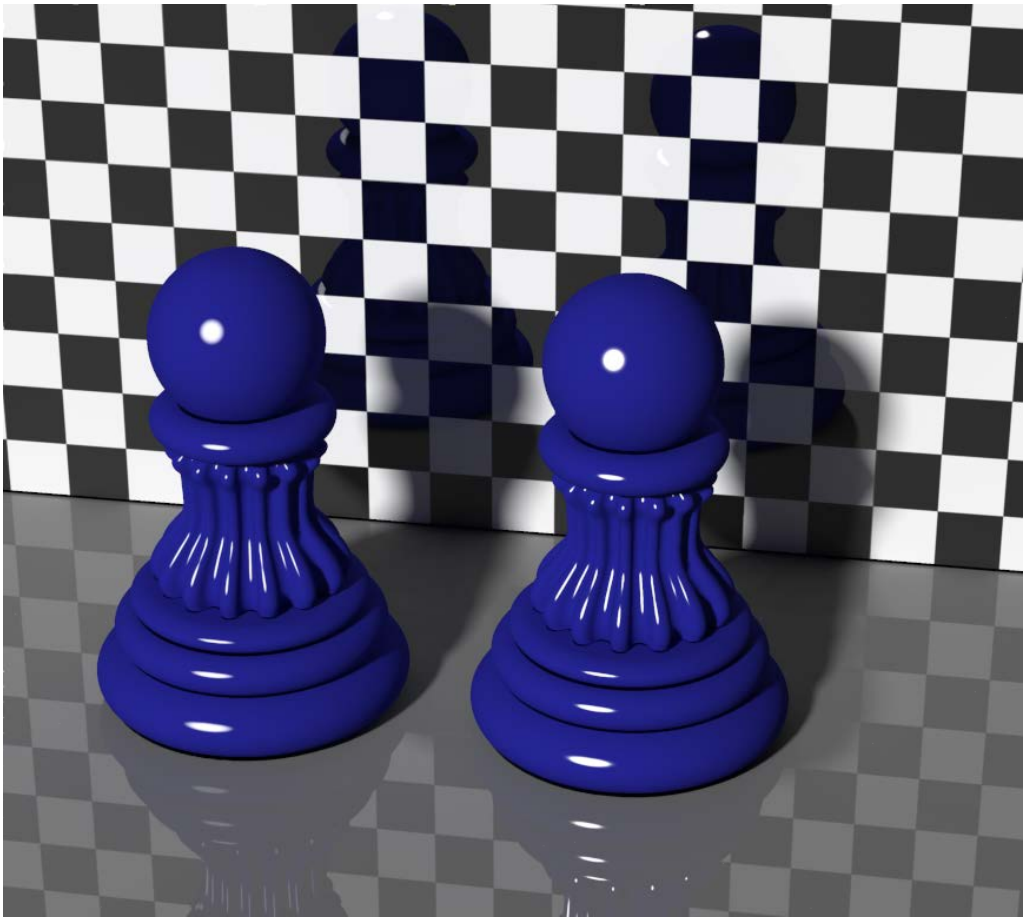
Intermediate

All nodes in the Specular group respond to specular light, which refers to specular light from direct (point- spot- infinite) sources only. The regular 'specular' node itself, either separate or embedded in the PoserSurface definition directly (see 43 WHAT'S THE DIFFERENCE BETWEEN THE SPECULAR COMPONENT, AND THE SPECULAR NODE?) is fast to calculate, but falls short for a neat illusion of about any real stuff except hard plastic. And even then Poser offers an alternative: Glossy for hard, smooth surfaces.

This is why Poser offers alternatives, like Blinn (44 WHAT'S THE BLINN NODE USED FOR?), or Anisotropic (46 WHAT'S THE ANISOTROPIC NODE USED FOR?) for microscopically grooved surfaces "with a direction". Generally, Blinn is the preferred node to use, but it's too soft for the really hard and shiny surfaces. Especially since not all hard, shiny surfaces are smooth. Some have a cellular structure, where each element is smooth and shiny, but the surface as a whole is more diffusive.



Recent Poser versions introduce the Ks-microfacet node for this. As can be seen (right pawn), the highlights are small and sharp as are those from glossy (left pawn).



So, like glossy was the improved version of Phong for hard and shiny stuff, ks microfacet seems the improved version of glossy, or the improvement over Blinn for hard and shiny stuff.
What's the difference with Glossy then?

Well, when I give the example a closer look, it shows that the glossy pawn (left) is fuzzy, blurred all over the highlight while the ks-microfacet one (right) is quite bold in the middle of the highlight, and becomes blurred at the edges only. To me, the latter has a stronger impression of smooth hard plastic or a hard blank lacquer than the first.
Another difference is at the parameter values: although the highlights have about equal sizes, the glossy pawn (left) needs a roughness 0,02 (or: 2%) while the ks-microfacet one (right) needs a roughness 0,05 (5%) for it. This implies that the ks-microfacet node can represent much smoother and harder materials than the glossy node, as it has more 'room to maneuver' for nuances left before it reaches the ultimate 0.0 value. (At roughness <0.001 I won't get a highlight of any size any more).

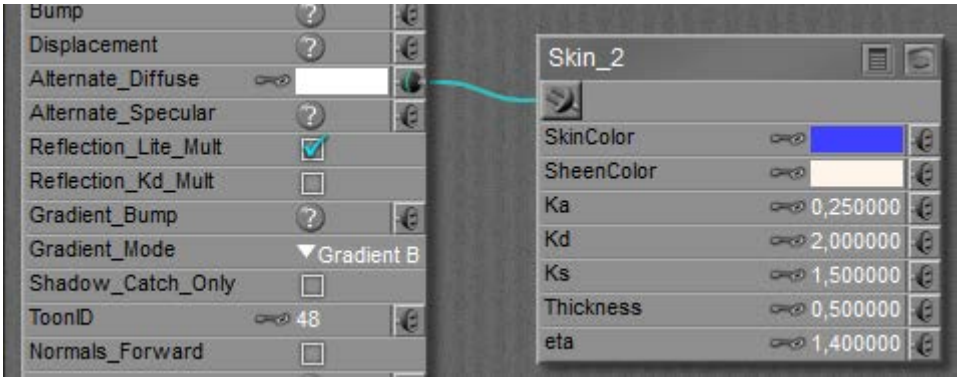
51 What's the Special > Skin node used for?

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

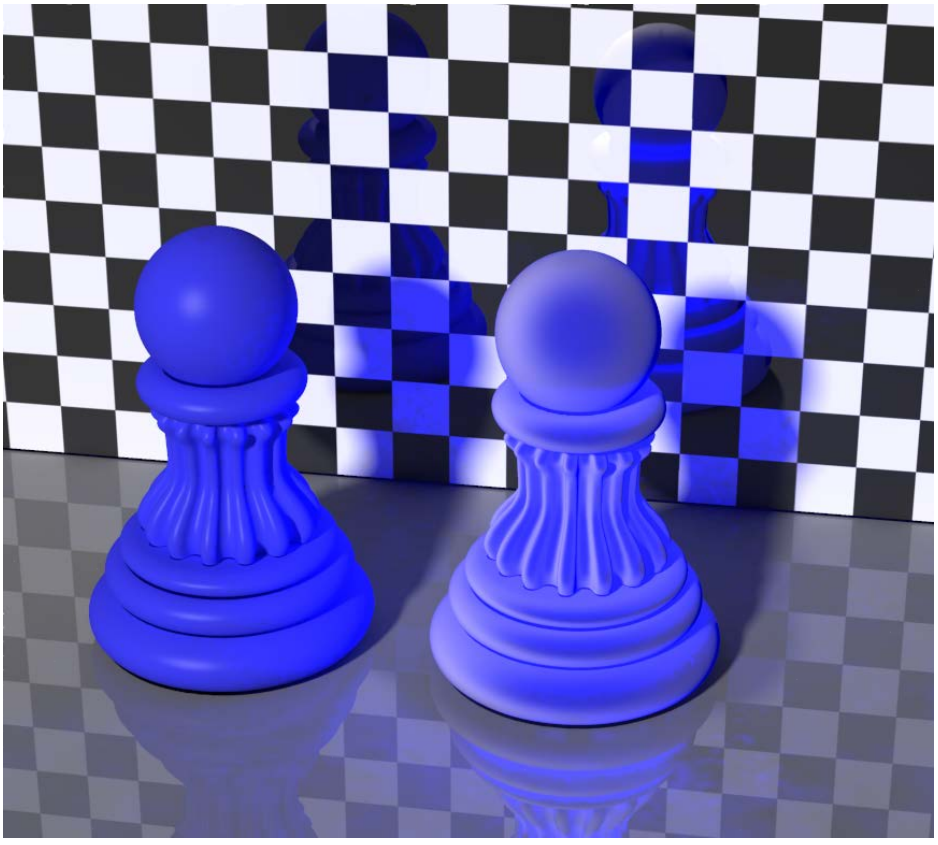
Intermediate

Skin is an "all in one" material node, which caters for diffuse, specular and ambient portions of the definition. So it responds to diffuse light (direct and indirect), specular light (direct only) and also generates some lighting of its own (glow). Just black-out (or nullify) the regular Diffuse, Specular and other properties of the PoserSurface, and plug the node into Alternate Diffuse.

The **SkinColor** is dominant when looking straight onto the surface, the **SheenColor** becomes dominant at skew angles towards the camera. The idea is that light is diffused from slightly below the surface. At skew angles that light travels a longer distance through the surface layer, and picks up more color from within that layer. SkinColor therefor represents the color at the outer side, SheenColor represents the colorization within the surface layer by the scattering.
The specular and ambient colors of the material are derived from them.

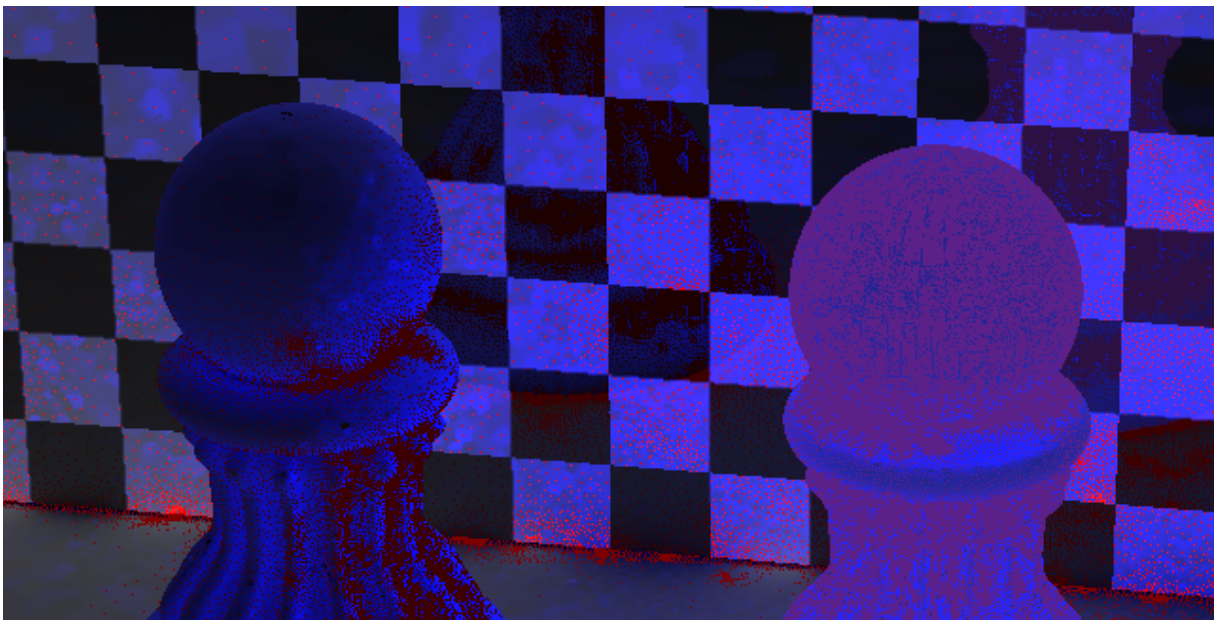


Kd is jargon for Diffuse value, **Ks** for specular value and **Ka** for ambient value.



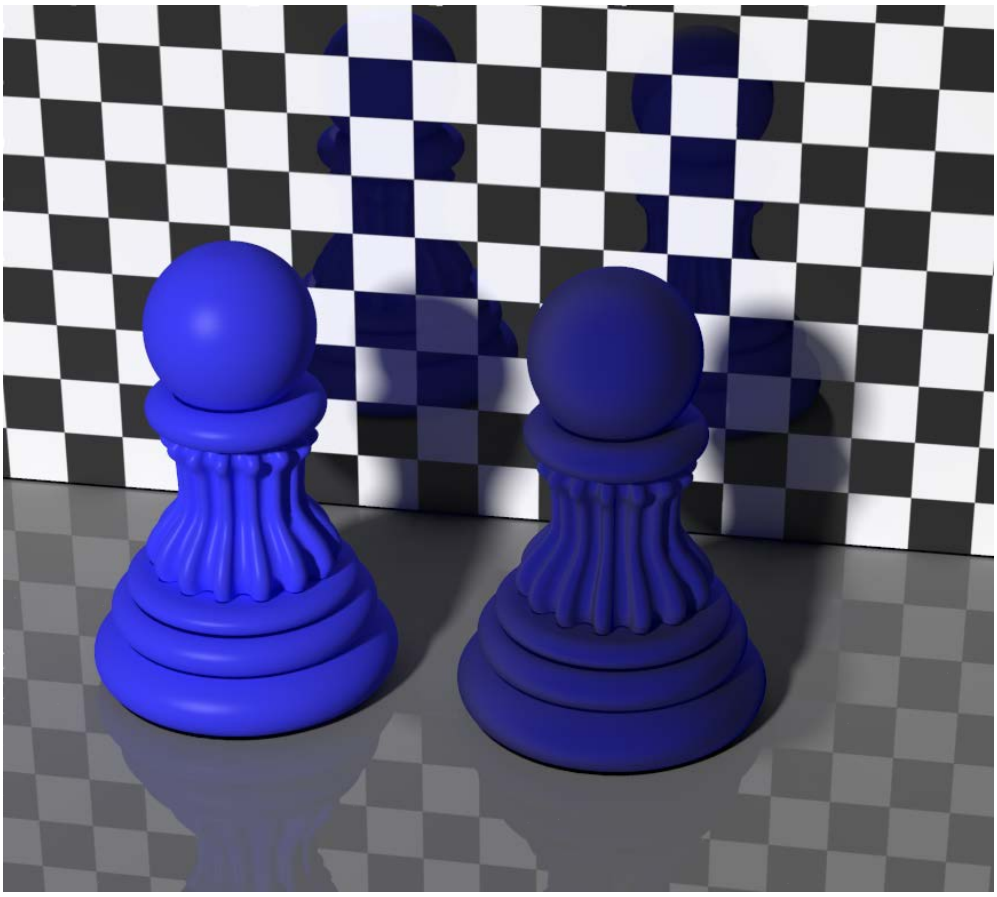
The left pawn shows a regular diffuse and specular use, the right one shows skin at default settings (except for the color adjustment to blue).

Note: with this node, precalculating Indirect Light takes quite some time, and the surface gets an enormous amount of ‘red dots’. RAM usage goes up accordingly (over 2Gb for a single pawn at 85% irradiance cache), so this node certainly is a ‘render killer’ and should be used on high performance machines (and patient users) only.



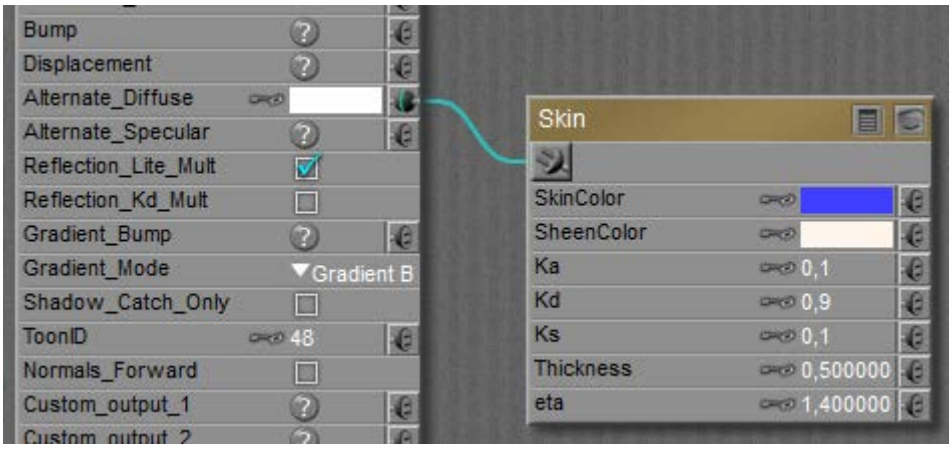
Left pawn using regular diffuse and specular, right pawn using skin node at default values.

As the left pawn had Diffuse value 0.9, Specular Value 0.1 (and 0.0 ambient), let’s use those values instead of the default ones.

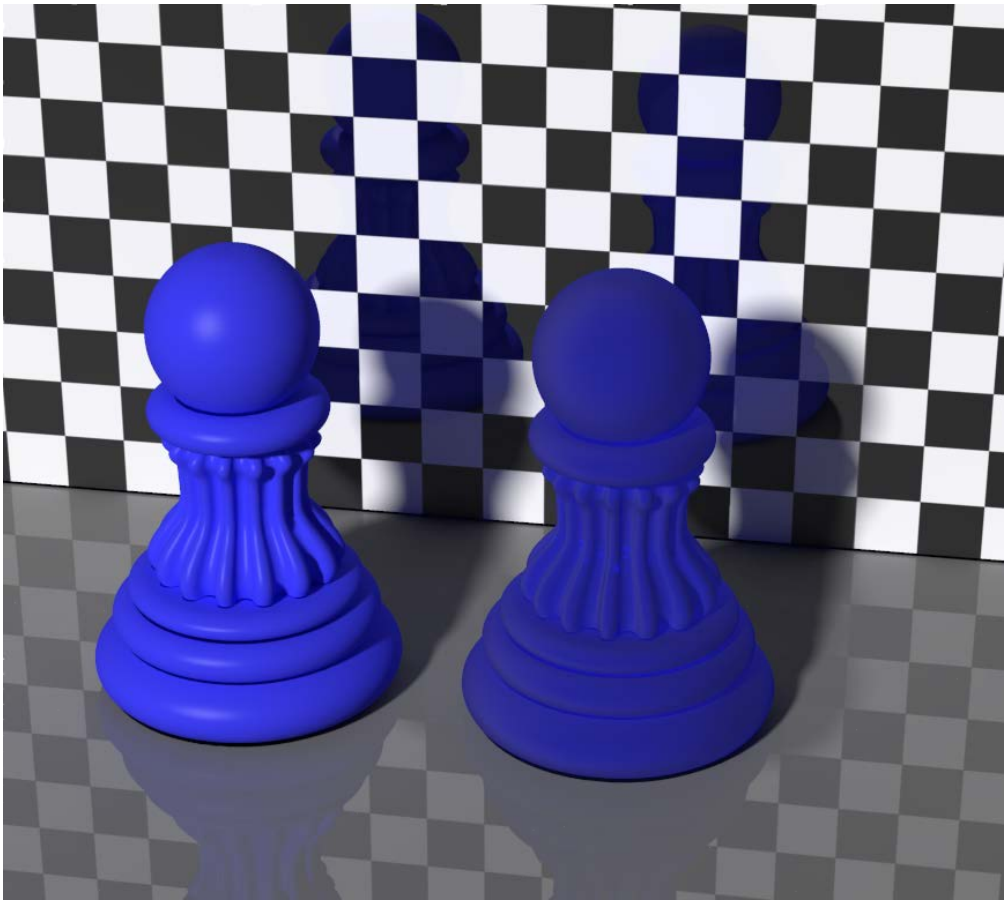


In the first place, the lower values for ambient and diffuse produce less Indirect Light, and renders much faster than the default settings.
Second, the result looks like velvet, without any noticeable highlights. So, the Kd and Ks do behave different from their Diffuse Value and Specular Value equivalents.
This effectively means that I'm on my own finding the right mix of settings.

Let's set ambient to 0.1, just to add a bit.



Now the result looks like glowing velvet:



See 58 WHAT'S THE DIFFERENCE BETWEEN ALL THOSE SCATTER NODES? on a comparison of the various skin and scattering nodes.

52 What's the Special > Subsurface Skin node used for?

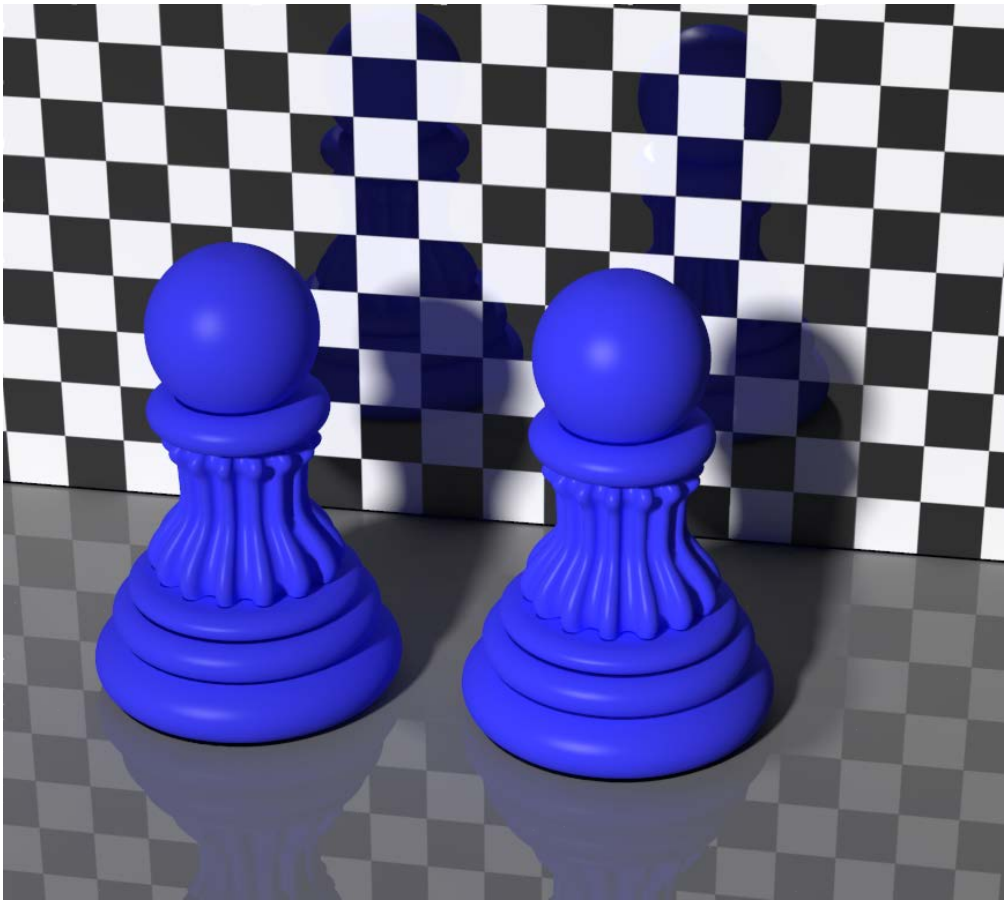
Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

Intermediate

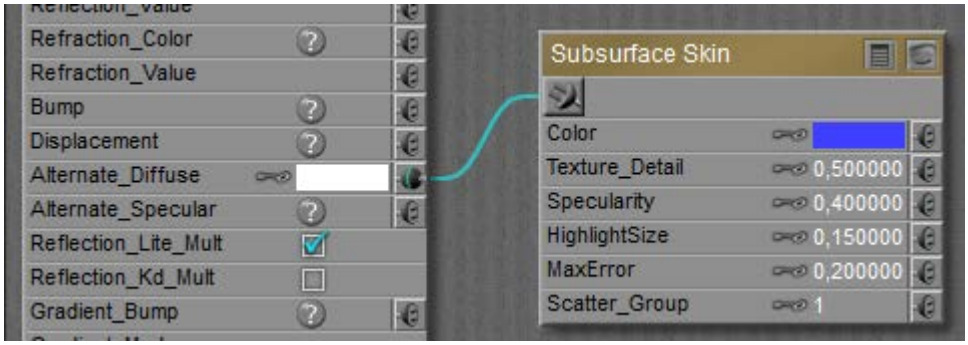
In "old days" or "classical" Poser solutions, regular Diffuse (with image maps), Specular (sometimes with special maps too), and some Ambient were combined to mimic the presentation of realistic skin. Modern Poser versions (8 and Pro 2012, and up) offer additional surface scattering. See 58 WHAT'S THE DIFFERENCE BETWEEN ALL THOSE SCATTER NODES? on a comparison of the various nodes.

For short: CustomScatter is the most generic, does require the SubSurface Scattering (SSS) option in the Render Settings, and provides direct access to all parameters of the mechanism. Subsurface Skin – discussed here – then is a derivative of CustomScatter, dedicated to (human) skin.

Like all special nodes, the Subsurface Skin node also provides diffuse, specular and eventually more as an all-in-one offer. Just black-out (or nullify) the regular Diffuse, Specular and other properties of the PoserSurface, and plug the node into Alternate Diffuse.



The left pawn shows regular diffuse and specular (default settings), the right pawn shows subsurface skin, as in:



To obtain a highlight of about the same size and intensity as the regular (left) material setting, the node needs about three times the regular values

for Specularity and Highlight Size. But since the material supports some scattering under the skin, it will produce a more washed out color overall.

Some notes, mainly on parameters:

- **Texture Detail**: this value is meaningful when a texture map is applied. At 1.0 the texture is applied at the outside of the skin layer, making the texture quite crisp but reducing the skin scattering. At 0.0 the texture is applied at the inner side of the skin layer, blurring the texture by the scattering. The default 0.5 is offered as a starting point.
- **MaxError**: 1.0 offers speedy rendering at reduced quality, while 0.0 offers max quality at the cost of long render times. The default 0.2 is offered as a starting point.
- **Specularity**: according to the manual, 1.0 matches human skin reflectivity. This is something different indeed from the regular Specular value where values exceeding 0.2 match shiny hard plastics. As said above, the Specularity and Highlight Size in this node need to be set three times as high as the settings in the regular Specular component to match the effect.
- This node requires the SSS in Render Settings being switched ON.

53 What's the Special > Velvet node used for?

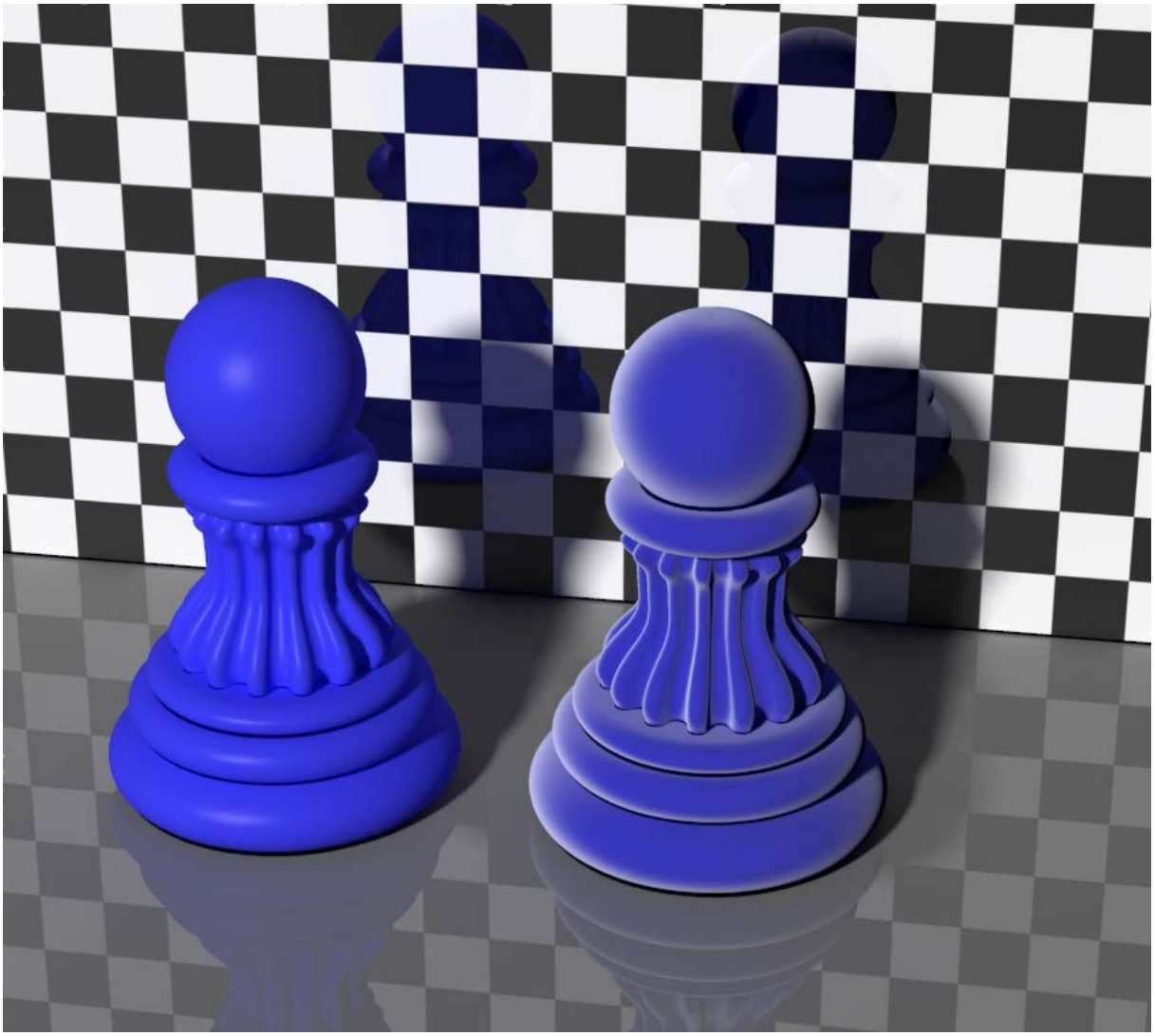
Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

Intermediate

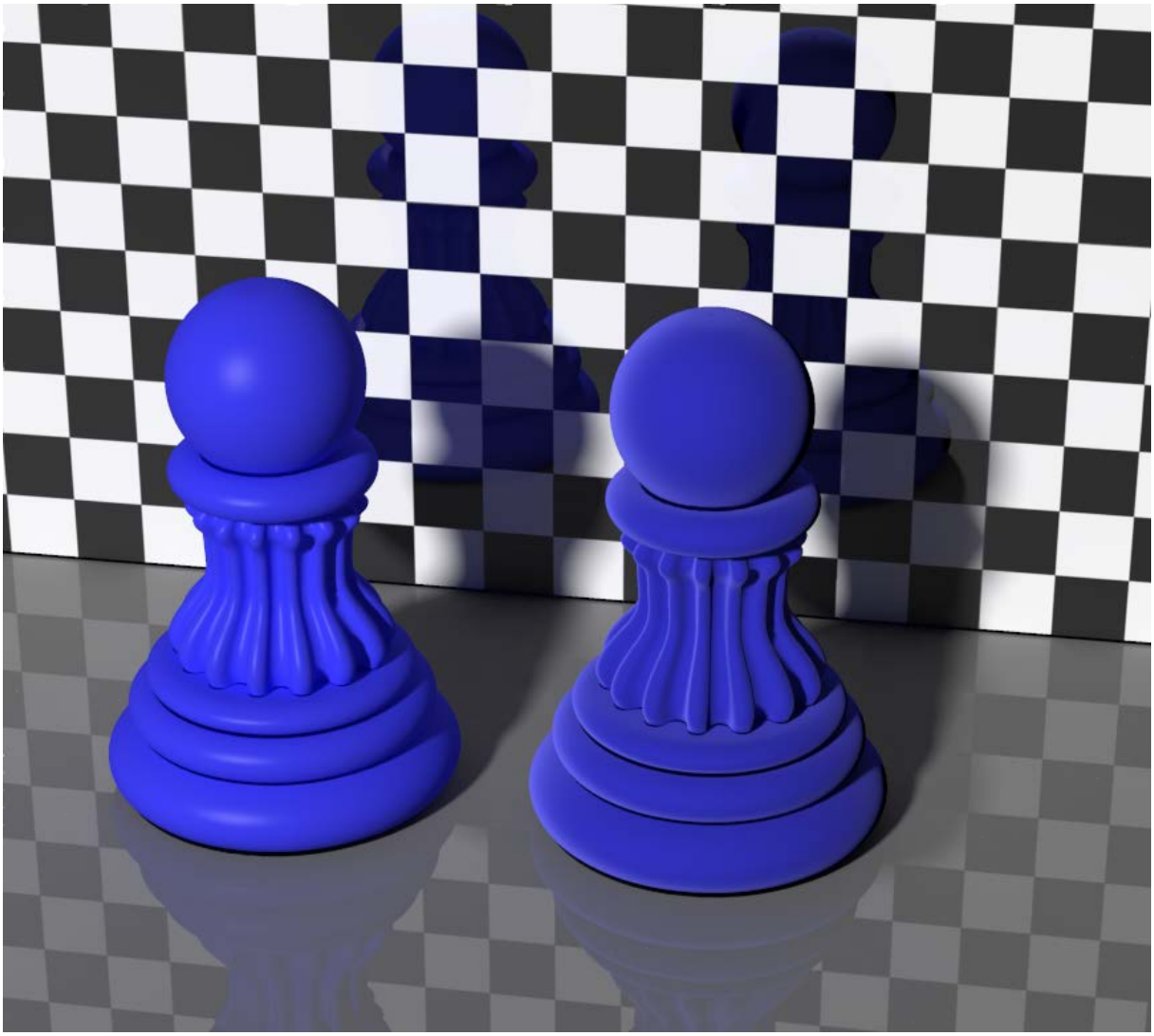
Real velvet is some 'hairy' cloth, with short shaved strands standing out. As a result, one looks into the bare ends of the strands when looking perpendicular to the cloth, without any specularity, and with deep dark color. When looking at it sideways however, one looks at the sides of the strands which do have some specularity, and usually have a lighter color. That's the 'Sheen'.

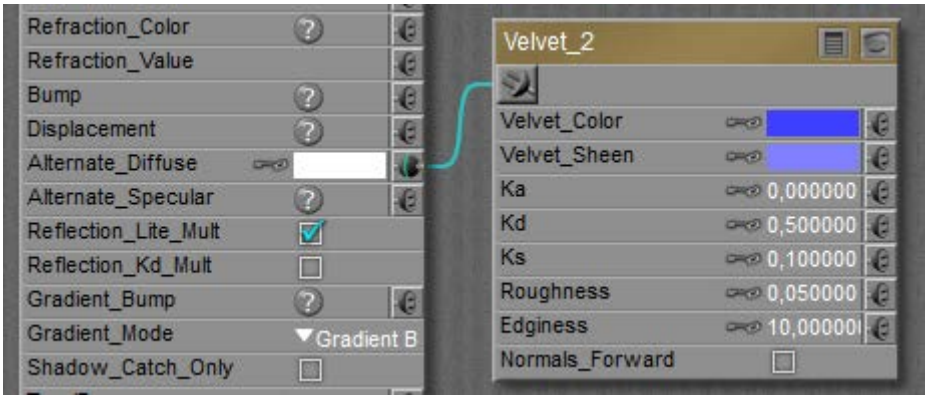
Note that the strands – which stand perpendicular to the cloth – that point towards the camera have the deep red color.

To illustrate this, just consider the extreme case of blue strands and a white sheen (right pawn, the left pawn shows regular diffuse and specular):

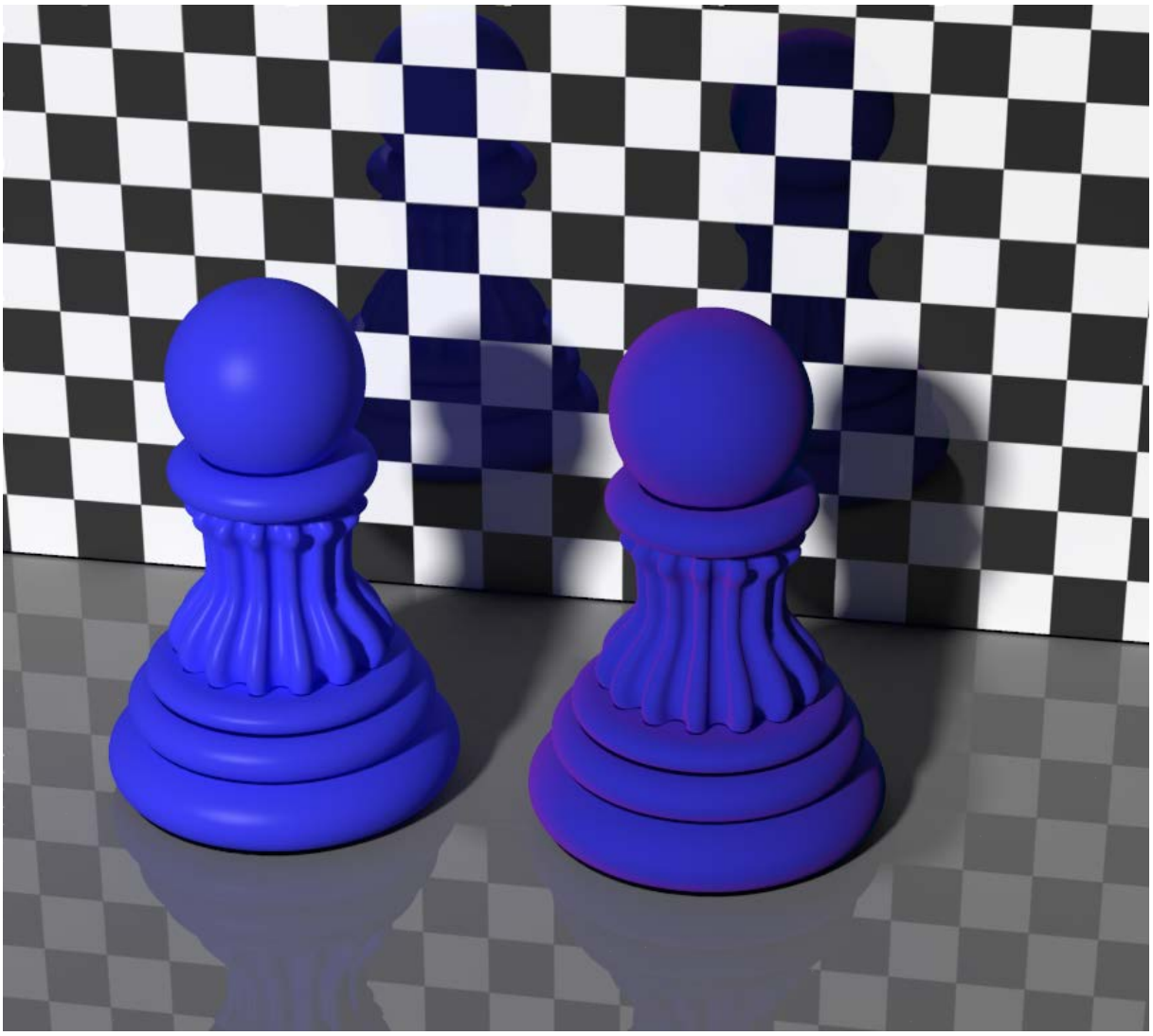


Although a just paler shade of the (Velvet_)Color makes a better sheen:





Note that darker or off-color sheens can make interesting stuff, which however are not realistic velvets. But strands with a bluish inside core and a dark purple outside coat can make:



Like all special nodes, the Velvet node also provides diffuse, specular and eventually more as an all-in-one offer. Just black-out (or nullify) the regular Diffuse, Specular and other properties of the PoserSurface, and plug the node into Alternate Diffuse.

Kd, **Ks** and **Ka** represent the regular Diffuse_Value, Specular_Value and Ambient_Value and present any required mixture of those components. **Roughness** is the direct equivalent of Highlight_Size in PoserSurface, or Roughness in the various specular nodes. Its value is similar as well.
Normals_Forward is the same as in PoserSurface, and in all diffuse and specular nodes.

The only new one is **Edginess**. The default (around 10) value makes realistic velvets. Low values (say 1.0) make very short-haired lightweight velvets, for which the Velvet_Sheen is quite dominant. High values (say 100) make long-haired heavy-weight velvets, for which the Velvet_Color itself is by far the dominant one.
Compare cloth for T-shirts (lightweight) and for theatre curtains (heavy weight). Both extremes will lose the velvet effect somewhat.

54 What's the Special > Hair node used for?

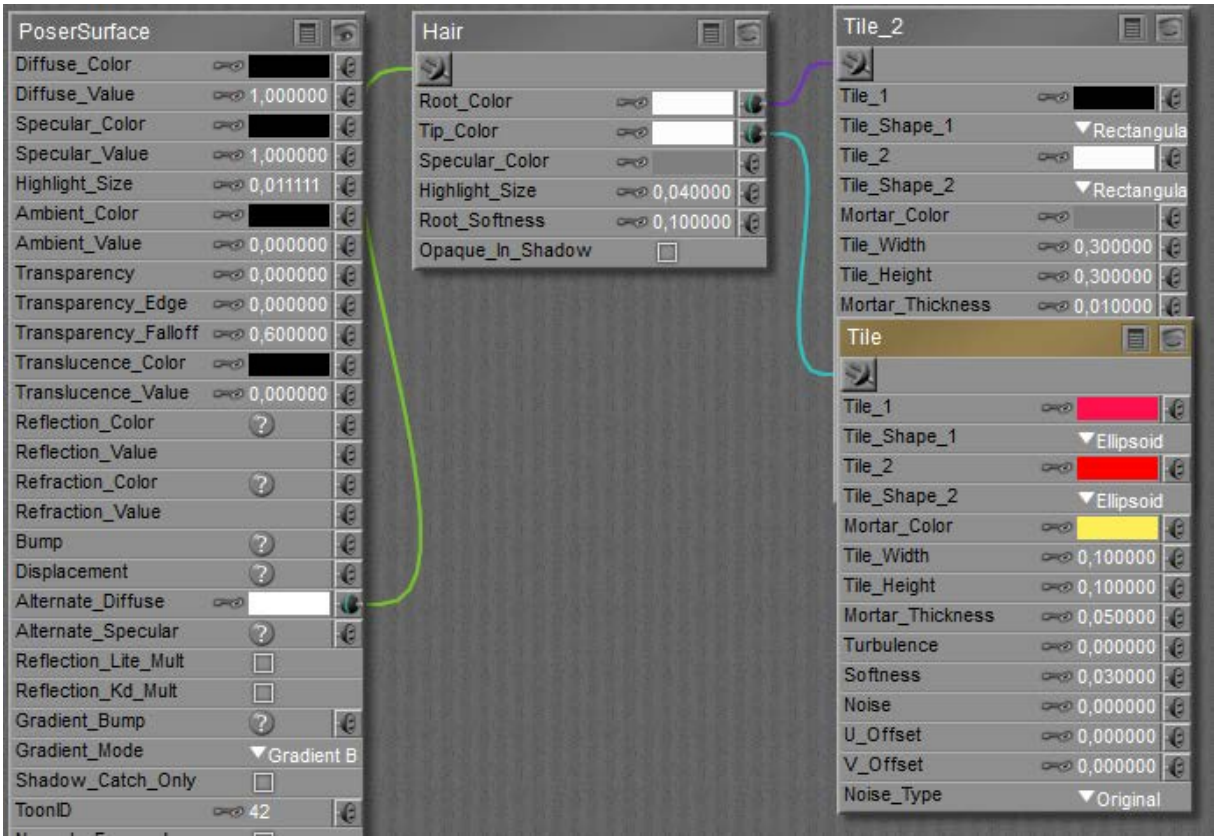
Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

Intermediate

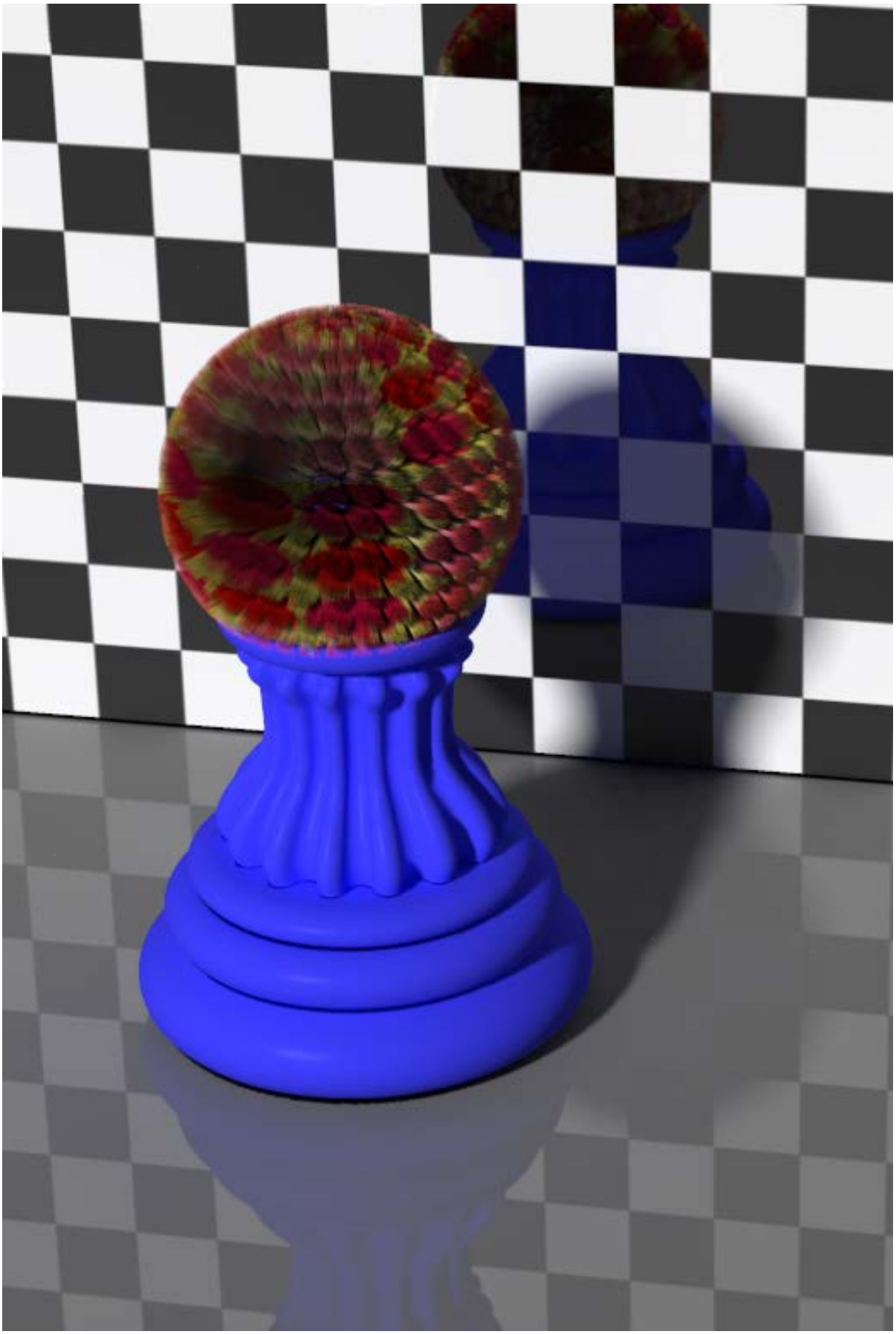
Like all special nodes, the Hair node also provides diffuse, specular and eventually more as an all-in-one offer. Just black-out (or nullify) the regular Diffuse, Specular and other properties of the PoserSurface, and plug the node into Alternate Diffuse.

But besides that, the Hair node is special in its own right. It's not just a surface cover, it's a full 3-dimensional U-V-and-W mapped coloring device. And… it's specially designed for the hair strands which result from applying the features in the Hair Room.

The (diffuse) coloring works as follows: one can apply a UV surface color at the hair-root level. A color, an image map, a 2D or 3D coloring node, whatever. So each strand gets its own color at the root. And one can apply a UV surface color at the other end, the hair-tip level. Then, Poser will interpolate the color, as in a gradient with set start and end colors, along the hair strand itself.

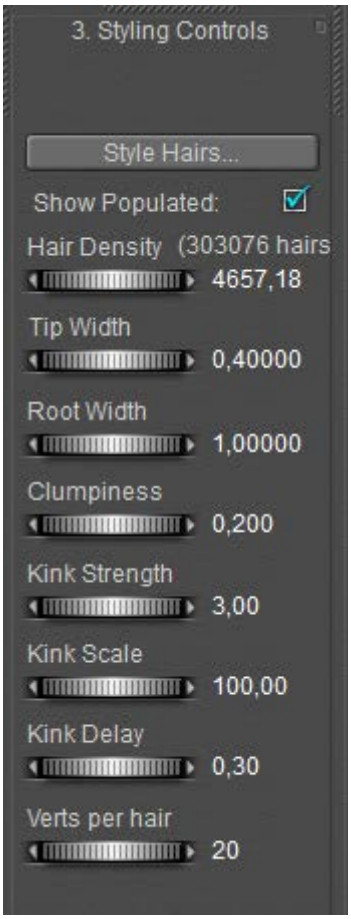


In the resulting image, the mid-part has a black tile at the root while the surrounding areas have a white tile underneath. So in some cases the color grades from black to red/yellow, and in the other cases from white to red/yellow. The black/white tiles are three times as large as the red/yellow ones.



Some notes.

- Hair – with any kind of coloring – is a challenge for rendering, due to its vast amount of vertices in the object. You might not realize this, but when you've got 300.000 hairs of 20 elements (verts per hair, see the screen grab) each, then you end up with 6.000.000 vertices. For comparison: a Poser figure like Vicky has about 60.000 vertices, so the hair object is 100 times as large.
  As a result, rendering stranded hair takes long render times and an enormous amount of memory. Like the image above: it took 90 minutes and 5.5 Gb ram to get rendered on quite a high-end machine.



- Under IDL conditions, each hair-element is considered to be a light source. This of course will grow out of hand pretty fast (as in the example above, I'll have 6 million of them!), and it will require even more memory and render time. I do switch Light-emitter OFF!

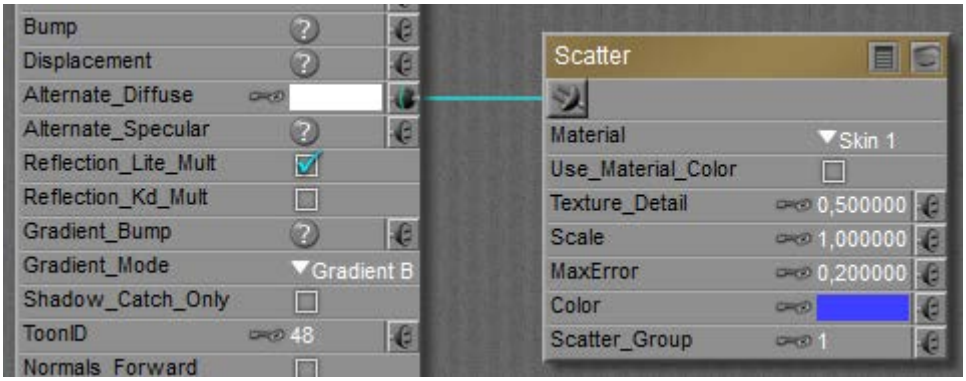
55 What's the Special >Scatter node used for?

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

Intermediate

Like all special nodes, the Scatter node also provides diffuse, specular and eventually more as an all-in-one offer. Just black-out (or nullify) the regular Diffuse, Specular and other properties of the PoserSurface, and plug the node into Alternate Diffuse. This Scatter node does not use the subsurface scattering as referred to in the Render Options. It's offered from Poser 8 / Pro 2012 on.

The node represents the scattering of light within the outer layer of a real-life material. It takes the thickness, the internal structure as well as the natural color of the material into account, like ketchup is darker red, apple is pale green and skin is pinkish.

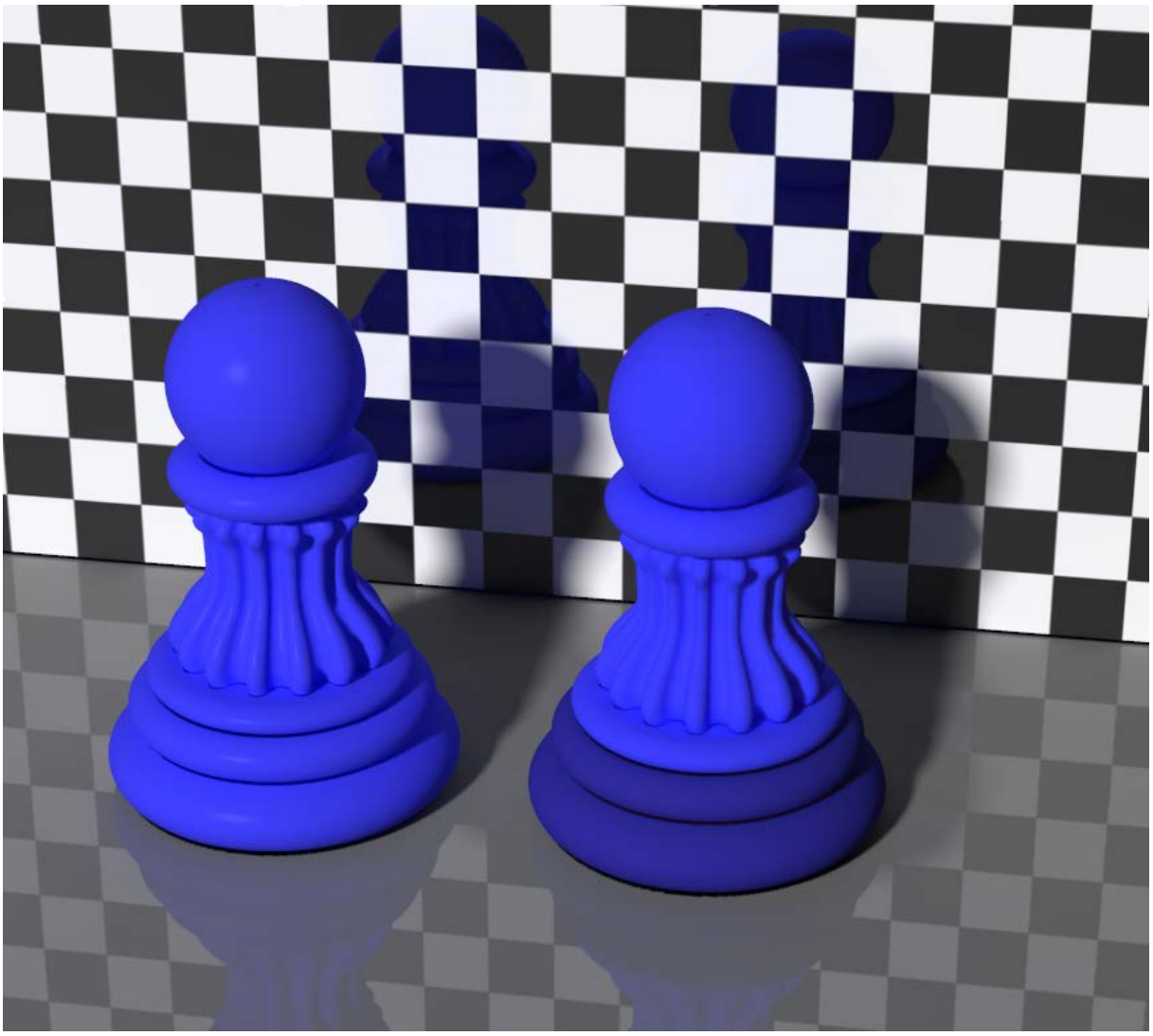


The internal structure of the scattering layer can be altered by picking a different **Material** from the list. The thickness of the layer can be altered by changing the **Scale** parameter, so when Apple (waterish in a cellular structure) is the substance of choice, but the surface layer of my stuff is twice as thick as a real-life apple (or the object is an apple, but at twice the regular size) then Scale should be set to 2.

MaxError is a speed vs quality setting, large values (up till 1.0) give high speed and lower quality, low settings give higher quality (more scatter bounces within the layer, I guess), at the cost of longer render times.

As in the regular (diffuse) case, the object surface can be tinted with color, image map, another texture node, etcetera. This surface color is multiplied (filtered) by the color of the scattering. If I don't want this latter effect, switch OFF the **Use_Material_Color** option. I might want the structure of Ketchup, the layer thickness of Ketchup but not the reddening from Ketchup.

In the example below the right pawn has the Scatter surface, with the Use_Material_Color option OFF except for the base where the pinkish skin color is merged into the main color.



The second issue is: do I want the surface texture on top / at the outside of the object, or at the inner part of the skin layer. In the first (outside) case, details will be crisp but are not scattered themselves, in the second (inside) case the details are scattered as well, but the texture is blurred because of that. This is handled by the **Texture_Detail** option: 1.0 means “outside” (all details available), 0.0 means “inside” and the default setting is halfway 0.5.

When the surface texture already offers an object color (as is usual the case), and I want to make use of the scatter color as well to improve skin tones etc., then some portion of the skincolor should be removed from the surface texture to avoid “double counting” of the coloring. The Poser Manual, chapter Material Room Nodes, sector Scatter tells in detail how to accomplish this.

See 58 WHAT’S THE DIFFERENCE BETWEEN ALL THOSE SCATTER NODES? on a comparison of the various skin and scattering nodes.

56 What's the Special >CustomScatter node used for?

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

Intermediate

Like all special nodes, the CustomScatter node also provides diffuse, specular and eventually more as an all-in-one offer. Just black-out (or nullify) the regular Diffuse, Specular and other properties of the PoserSurface, and plug the node into Alternate Diffuse.

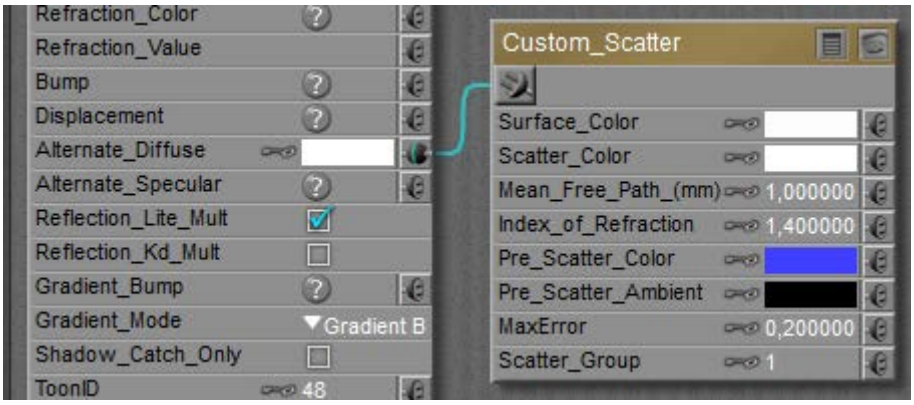
This CustomScatter node provides the raw parameters for Subsurface Scattering, as are addressed indirectly by all presets from the dedicated nodes. It does use the subsurface scattering as referred to in the Render Options. It's offered from Poser 8 / Pro 2012 on.

The node takes the **Pre-Scatter** texture (might be an image map) and eventually some **Pre-Scatter Ambient** as well, scatters that around within the object surface layer while filtering it with **the Scatter Color** (white is: no additional coloring), and then the light leaves the surface layer while being filtered by the **Surface Color** parameter, So the Pre-Scatter Texture will get blurred while the Surface Color texture will remain sharp.

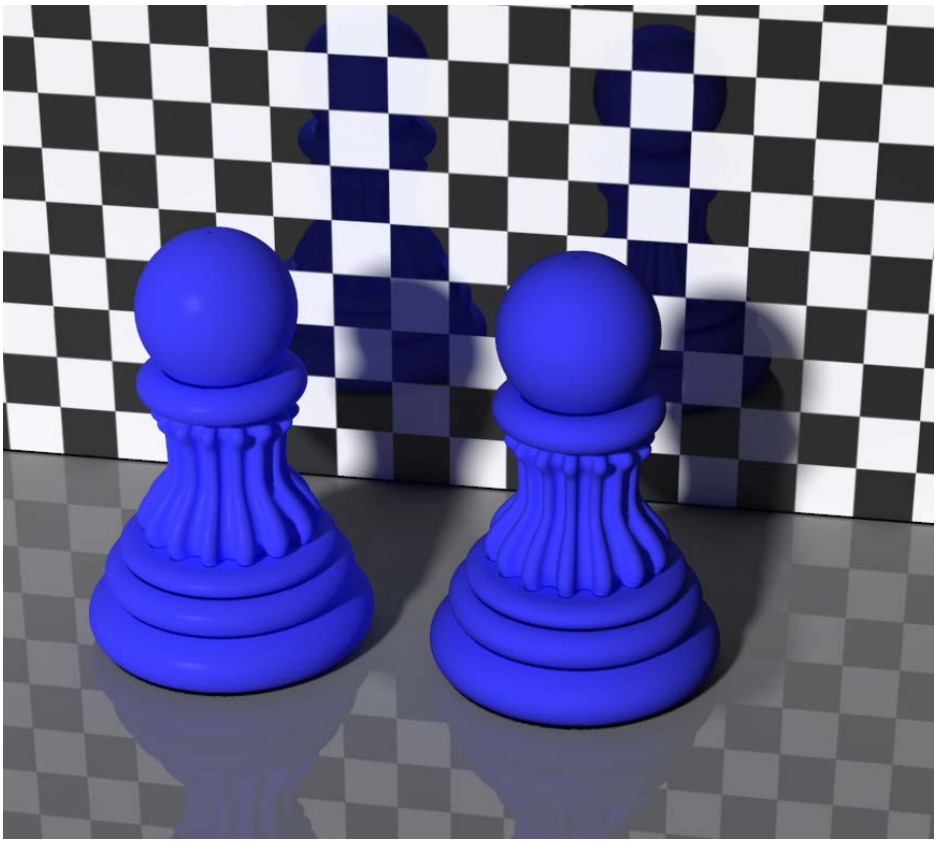
MaxError is a speed vs quality setting, large values (up till 1.0) give high speed and lower quality, low settings give higher quality (more scatter bounces within the layer, I guess), at the cost of longer render times.

The scattering itself is defined by

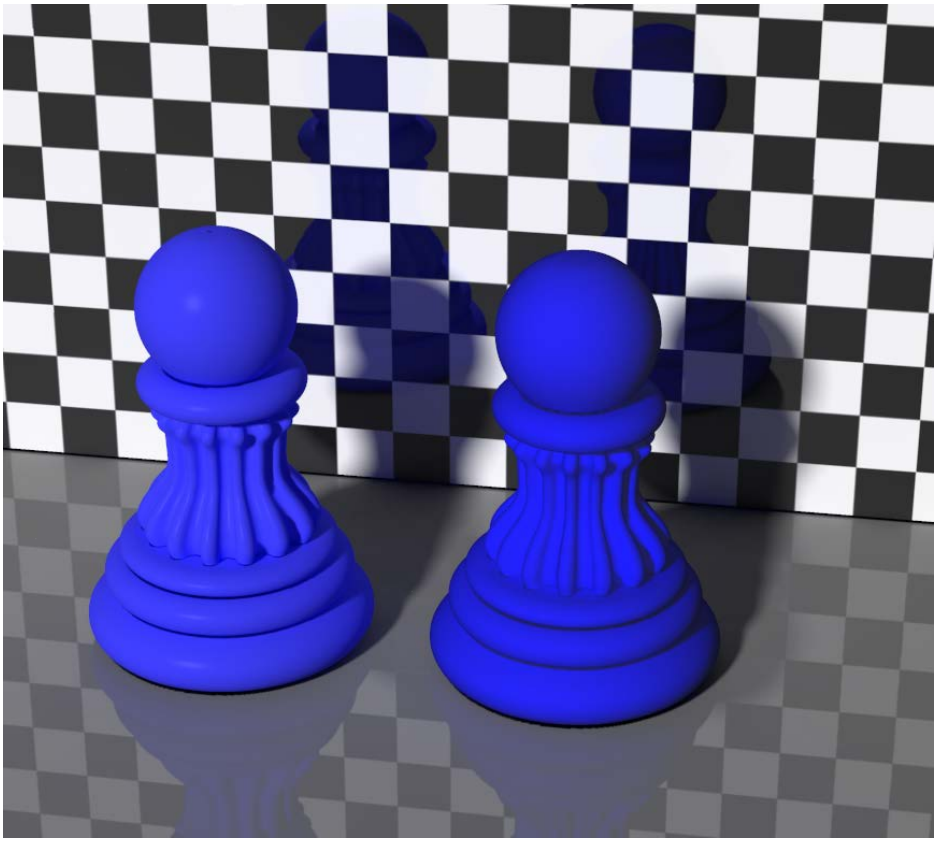
- **Main Free Path**, in mm, the distance a scattered light ray can travel before being scattered again. In a cellular structure like skin, the cel-size might be a good start. This is about 0.15 mm for human skin, and say twice as much for some plant leaves. For suspension fluids (milk, ketchup) smaller values will apply.
  This value is also a measure for the skin layer thickness, and as milk and ketchup don't have any in real life it's my artistic taste only which really counts in the end.
- IOR or: **Index of Refraction**, is determined by the stuff the skin layer is made of. The default 1.4 is fine for troubled water, while 1.8 is the value for glassy stones (as marble). I tend to stay in the 1.2 … 2.0 range to keep a sense of realism.



In the following render, the left pawn shows regular diffuse and specular. The right pawn shows Custom_Scatter with the settings as above, while having the object color in the Pre_Scatter_Color swatch.



By assigning the object color to the Surface Color swatch too, the color filtering will be done again when the scattered ray leaves the object. This intensifies the colors significantly:



Next to all this, see 58 WHAT'S THE DIFFERENCE BETWEEN ALL THOSE SCATTER NODES? on a comparison of the various skin and scattering nodes.

57 What's the Special > FastScatter node used for?

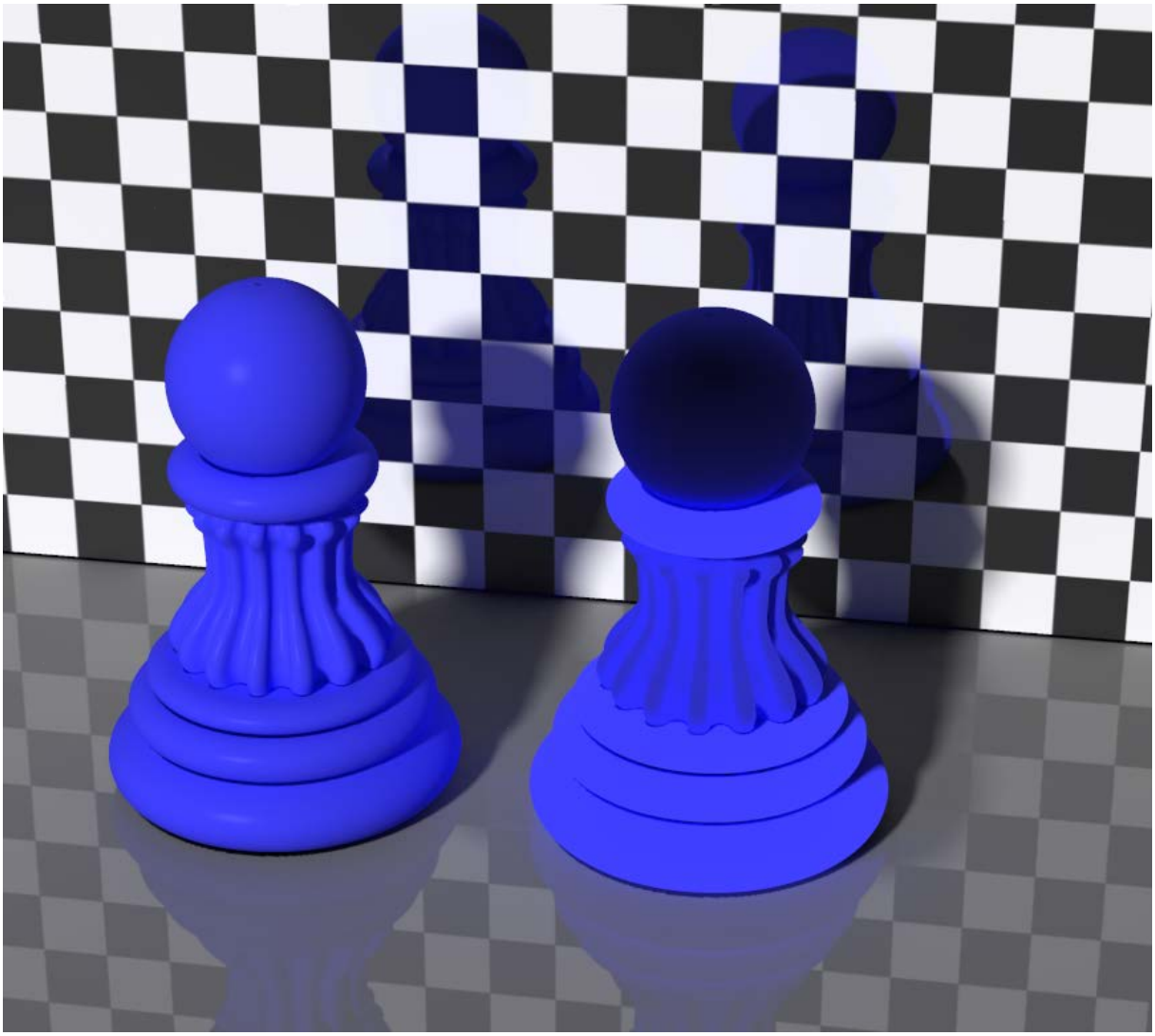
Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

Intermediate

Like all special nodes, the FastScatter node also provides diffuse, specular and eventually more as an all-in-one offer. Just black-out (or nullify) the regular Diffuse, Specular and other properties of the PoserSurface, and plug the node into Alternate Diffuse.

FastScatter is an early, simple – and according to the manual outdated – way of representing SubSurface Scattering. It does not use the subsurface scattering as referred to in the Render Options.

Besides a single **Color** entry, the choice is for **None** (demonstrated in the right pawn base in the image below), **Light** (pawn column) or **Strong** (pawn head) material. The left pawn in the render shows regular diffuse and specular, for comparison.



See 58 WHAT'S THE DIFFERENCE BETWEEN ALL THOSE SCATTER NODES? on a comparison of the various skin and scattering nodes.

58 What's the difference between all those Scatter nodes?

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

Intermediate

Poser Material Room offers the following nodes

- Skin, see 51 WHAT'S THE SPECIAL > SKIN NODE USED FOR?
- Subsurface Skin, see 52 WHAT'S THE SPECIAL > SUBSURFACE SKIN NODE USED FOR? (Poser 8 / Pro 2012 and up)
- Scatter, see 55 WHAT'S THE SPECIAL >SCATTER NODE USED FOR? (Poser 8 / Pro 2012 and up)
- CustomScatter, see 56 WHAT'S THE SPECIAL >CUSTOMSCATTER NODE USED FOR? (Poser 8 / Pro 2012 and up)
- FastScatter, see 57 WHAT'S THE SPECIAL > FASTSCATTER NODE USED FOR?

addressing scattering of light within an object surface (skin) layer.

From these, **CustomScatter** is the most generic, does require the SubSurface Scattering (SSS) option in the Render Settings, and provides direct access to all parameters of the mechanism. Which, by the way, are nice words telling that I'm on my own finding the appropriate values for them.
As it actually deals with the light rays scattering within the surface layer, it takes the most resources and render time. And gives the 'best', most manageable results in return.

Subsurface Skin then is a derivative of CustomScatter, dedicated to (human) skin.

Scatter on the other hand offers the possibility to pick a preset, based on something similar to what I want to accomplish. It does not use the SubSurface Scattering option in the Render Settings, and so it takes (far) less time and resources for rendering, but the results might be somewhat less realistic. They are less manageable anyway, but is does resemble scattering to some extent.

The ones just mentioned above are available in Poser 8 / Pro 2012 and up. The ones mentioned below are available in earlier versions as well.

Skin itself is just a combination of regular material properties, dedicated to (human) skin. It presents the diffuse, specular and ambient properties of a Poser material in a balanced way and mimics the waxiness of skin in a classical way, as used in the earlier versions of Poser. It does not use the subsurface scattering as referred to in the Render Options.

FastScatter is an early, and simple – and according to the manual outdated – way of representing SubSurface Scattering. It does not use the subsurface scattering as referred to in the Render Options.

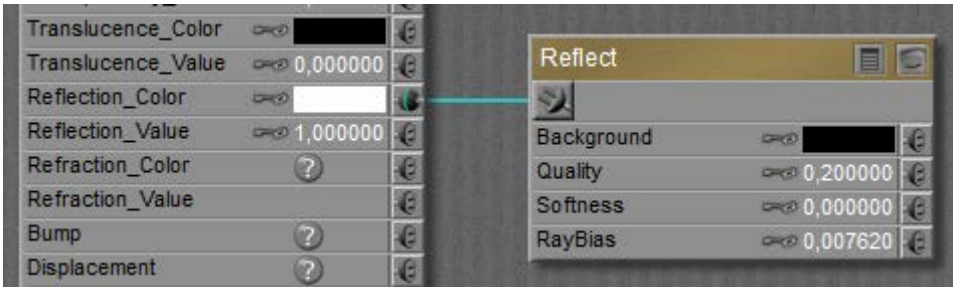
59 What's the Raytrace > Reflect node used for?

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

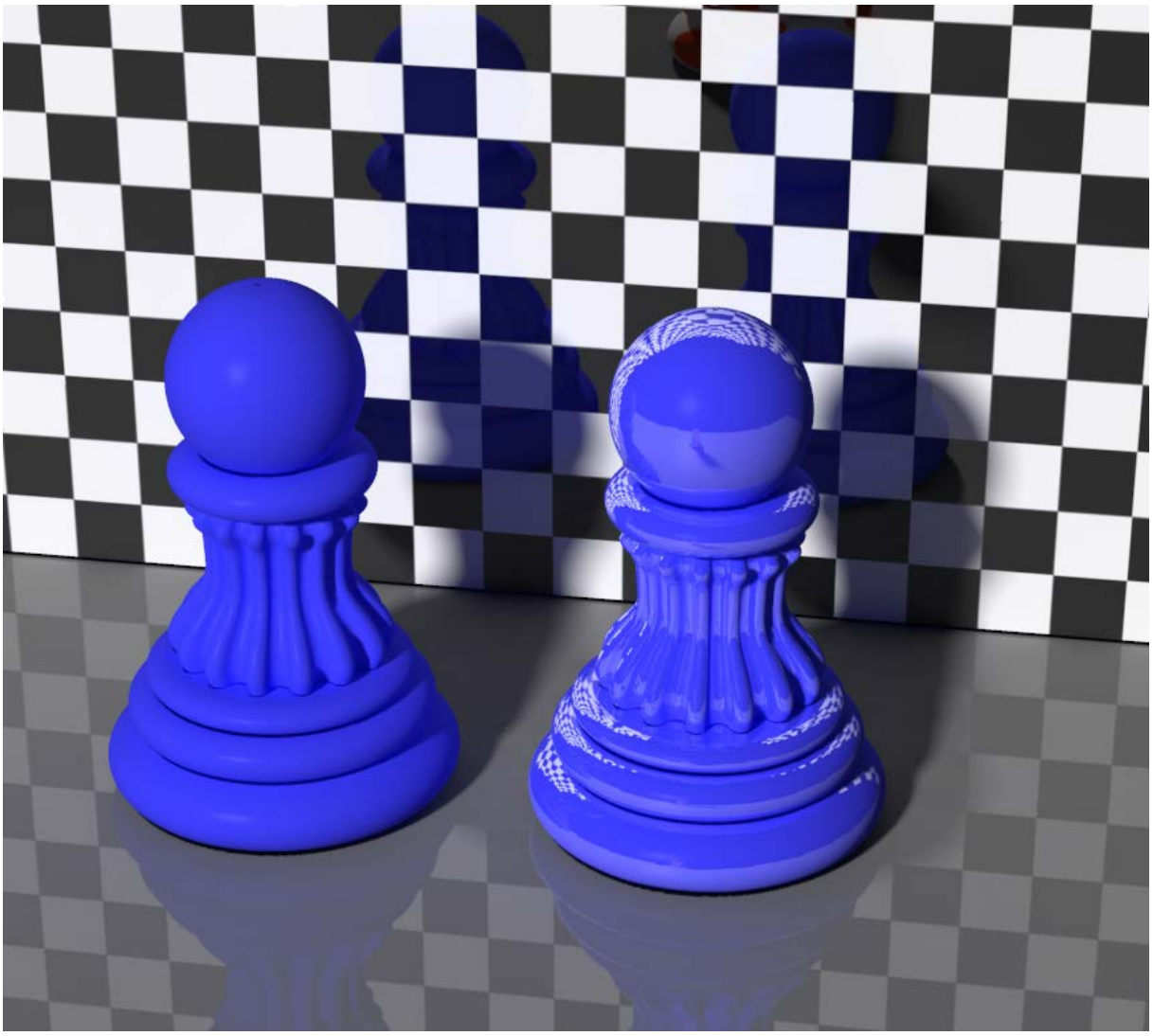
Intermediate

This Reflect node adds accurate, raytraced reflections to the PoserSurface material definition. It requires Raytracing in Render Settings to be switched ON. The quality of the result depends on the 'Number of Bounces' set in Render Settings as well. This number is a maximum value, when Poser does not need them it won't use them, but if the number of bounces for a light ray exceeds this limit, this light ray is killed. This might speed up the rendering while it also might introduce artifacts (black spots) in the result.

Do note that Reflection only handles the light from objects in the surrounding scene. It does not cater for the rays from direct light sources (spot, point, etc. lights). This part is dealt with by Specularity. Reflection and Specularity are two portions of the same natural phenomenon: reflectivity. These two material properties should be balanced with respect to that, but since they have no parameters in common, that's mainly a matter of taste and intuition. Effectively, I'm balancing the intensities a) of the direct lights and b) of the light bouncing from the objects, with respect to reflection.



Of course there must be something to reflect in the scene at the first place, and this sort of requires that my scene is build out far beyond and behind the scope of the camera. I probably don't want to do that, and Background offers to possibility to supply an additional background color or image or texture node instead. In the screen grab above nothing is supplied, but turn the color swatch to white or grey and plug in the appropriate node instead, and the job is done.



The left pawn shows regular diffuse and specularity, the right pawn has reflection added to this.

The **Quality** parameter offers a tradeoff between speed and result; high values require longer render times but present crispier results. **Softness** increases the blur of the reflected image, representing irregularities, impurities and even minor movements and vibrations in the reflecting surface.

RayBias is a feature to address a special issue, regarding the reflections on image-based displaced surfaces. RayBias is a distance, in the Poser units as set in my User Interface preferences. The default is 0.3 inches or 0.007620 meters as in the screen grab above.
Poser takes the reflecting surface, moves it for this distance outwards from the object, then it calculates the reflections, and then it projects those reflections onto the surface of the object itself.

This way, all irregularities of the reflecting surface which are smaller than this RayBias distance are disregarded; usually these are caused by the details of displacement maps. It speeds up reflection calculations considerably, but it does introduce artifacts and less accurate reflections as a downside. For that reason, one should not have this value larger than the amount of displacement in the same PoserSurface material definition. The 0.3" (0.76cm) default value is quite a lot, actually.

Reflection Color (in the PoserSurface column) usually works with white, unless I'm representing a metal. Metals have colored reflections, like Gold is yellow and Copper is red. The **Reflection Value** (PoserSurface as well) represents the reflectivity of the surface, and despite my experiences in nature, this value is quite low for most materials, except metals. See 09 WHAT'S THE REFLECTION COMPONENT INTENDED FOR?, 31 WHAT'S A PROPER COLOR FOR REFLECTIONS? and 32 WHAT'S A PROPER VALUE FOR

REFLECTIONS? for more on this, including detailed colors and reflectivity values for various kinds of real-life stuff.

One note of caution: the final amount of reflectivity will be made up from the combination of Reflection_Color * Reflection_Value. To represent low reflectivity, I can either use a dark color, or a low value. But I should not do both! This is especially relevant for colored metals, where reflectivity becomes somewhat reduced simply but not using white in the color swatch.

Raytraced reflections are realistic, detailed, and therefore time and resource consuming. When I don't need this level of accuracy, reflections can use a simple image map, or as a compromise, a Spherical Environment Reflection Map. See 65 WHAT'S THE ENVIRONMENT MAP > SPHERICAL MAP NODE USED FOR? for details.

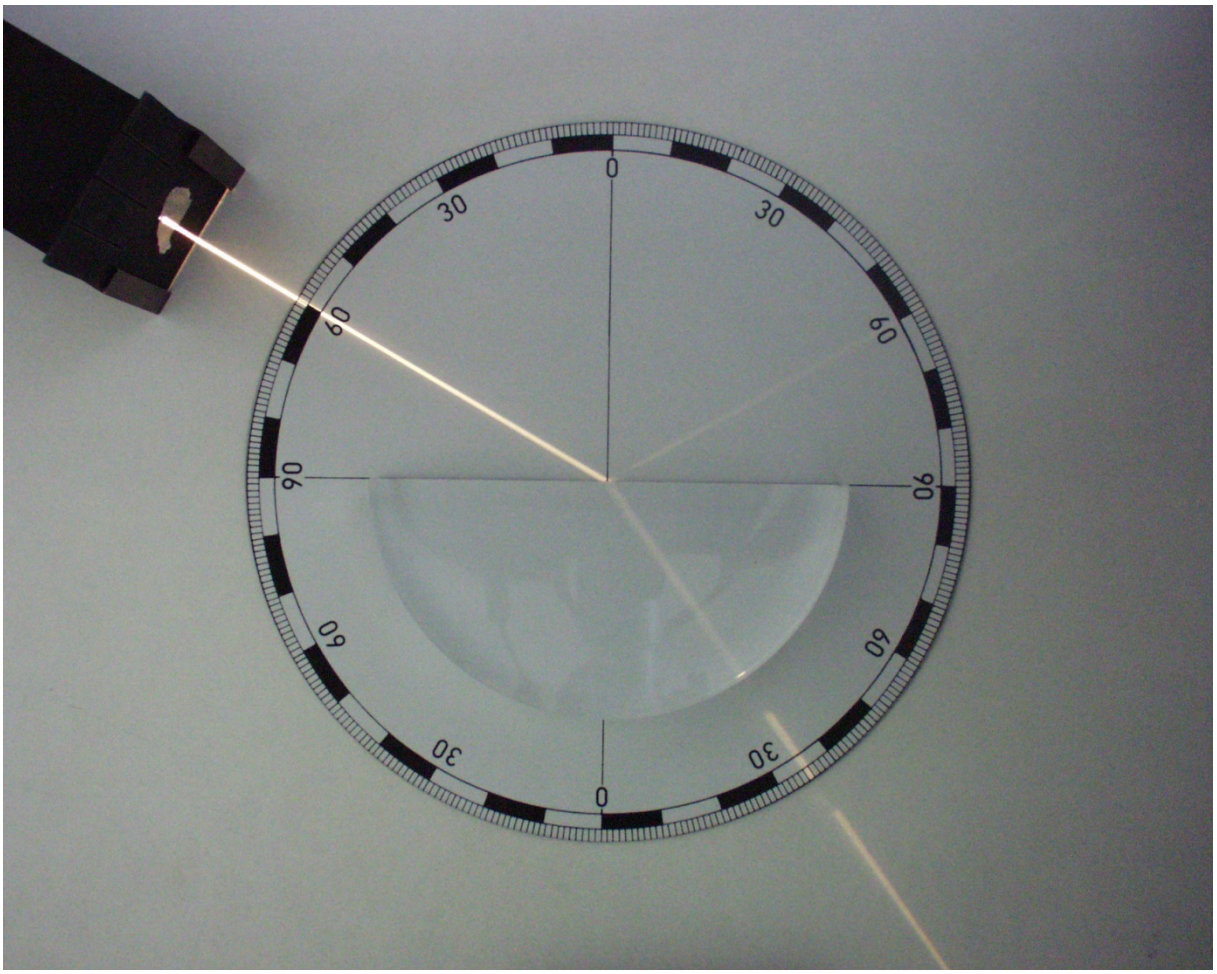
The raytraced reflections as presented by the Reflect node are not extremely photo-realistically accurate however. In real life, the amount of reflection depends somewhat on the angle the surface is looked at by the camera. Skew angles increase reflectivity. This is known as the "Fresnel effect". It's mainly visible for fluids and glasses.

60 What's the Raytrace > Refract node used for?

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

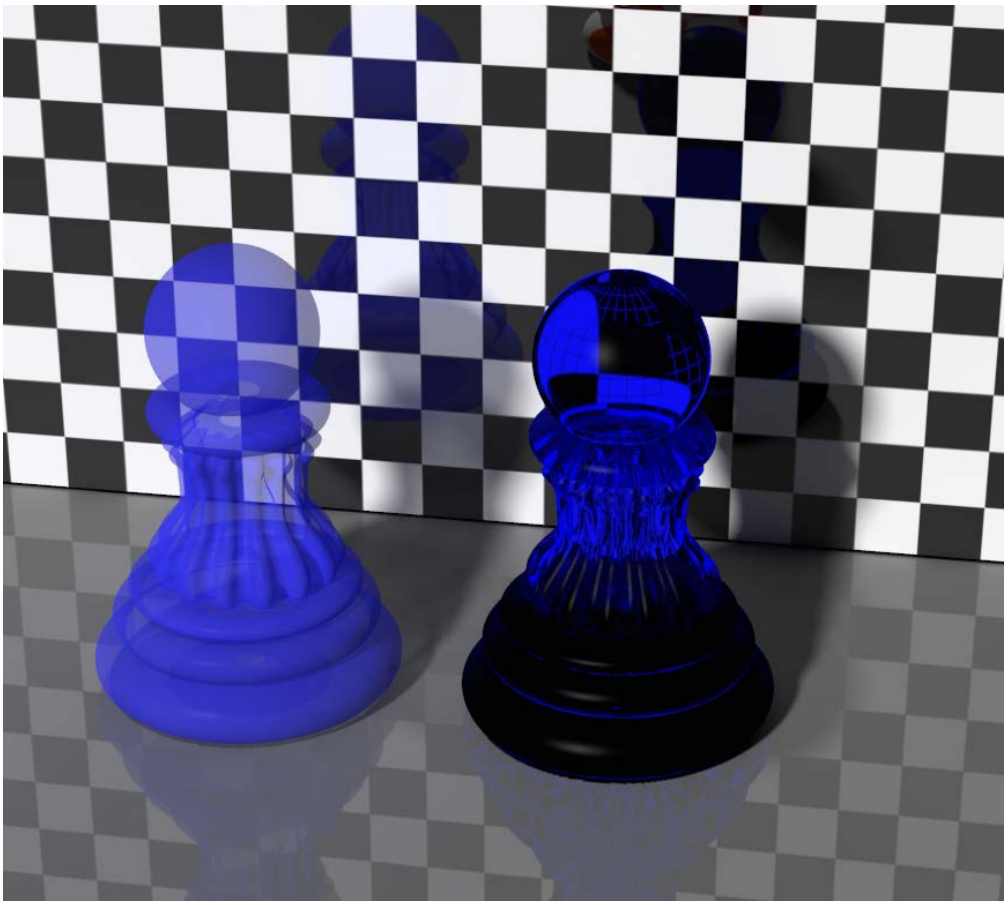
Intermediate

Abstractly spoken, any object surface is the separation of two volumes: the 'inside' part and the 'outside' part relative to the object. When both parts are transparent to some extent, a light ray can pass through the surface. And when both parts consist of different stuff, with a different 'optical density' (like air and water, water and glass, …) then the light ray gets bend at the surface because the speed of light is different at both sides.

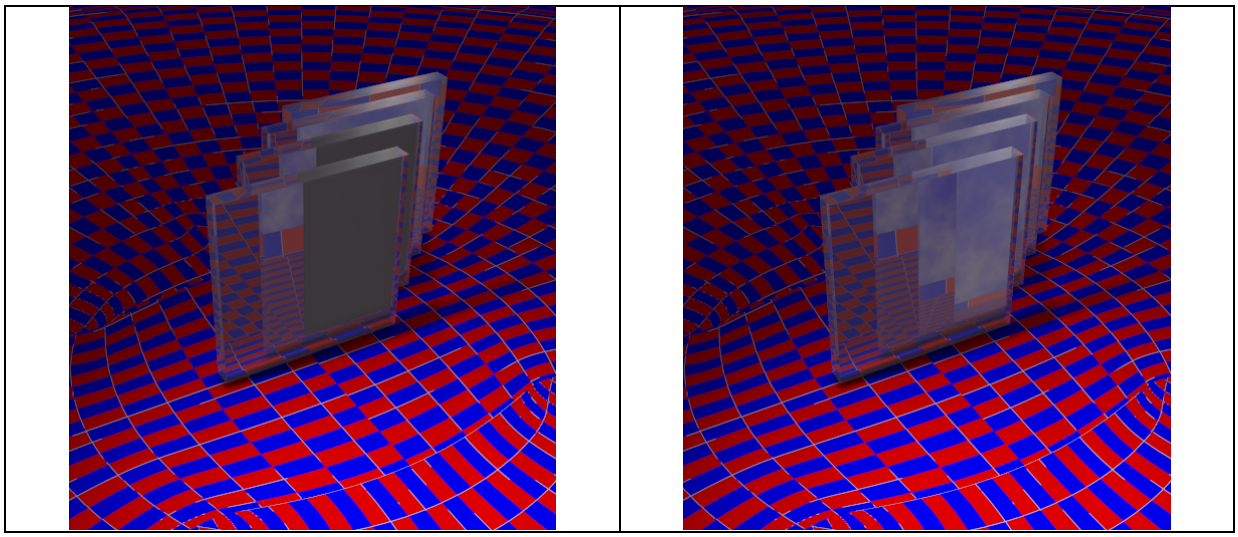


This is 'refraction', and by inventing an **Index of Refraction** (IOR), being 1.0 for vacuum, each material can have its own value defining its optical properties. In other words, the IOR defines the amount of refraction, the extent of light ray-bending, at the surface.

The image below illustrates transparency without refraction (left pawn) and refraction without transparency (right pawn). Without refraction the background image passes through undeformed, with refraction the pawn shows a glass-like behavior. As I can see, Poser refraction takes care of transparency all by itself (*).



This Refract node adds accurate (*), raytraced refraction to the PoserSurface material definition. It requires Raytracing in Render Settings to be switched ON. The quality of the result depends on the 'Number of Bounces' set in Render Settings as well. Passing through a surfaces counts as a bounce, so entering and leaving a transparent object requires two bounces.
The number set is a maximum value, when Poser does not need them it won't use them, but if the number of bounces for a light ray exceeds this limit, the light ray is killed. This might speed up the rendering while it also might introduce artifacts (black spots) in the result. The tradeoff is mine, but as nature has an infinite number of bounces, the max value is the best when I can afford it.



Left: Raytrace bounces set to 4, while 4 objects require 8 bounces.
Right: When the value is increased to 8 or more, all objects and surfaces can be passed.

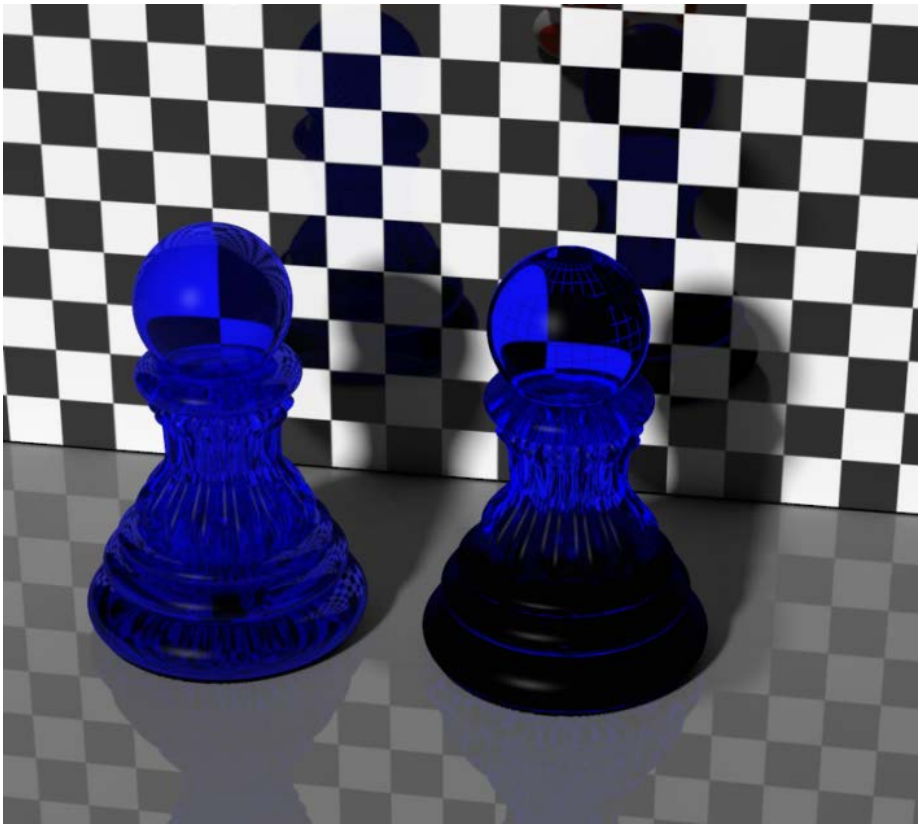
(*) Notes:

- Refraction only handles the light from objects in the surrounding scene. It does not cater for the rays from direct light sources (spot, point, etc. lights), these will not get bend at all. Refraction does not work for light passing through an object shining onto another object: refractive objects behave opaque to direct light and make deep shadows accordingly. But Transparency behaves as expected, as discussed in 08 WHAT'S THE TRANSPARENCY COMPONENT INTENDED FOR? (basic) and 64 HOW DO I PROPERLY COMBINE TRANSPARENCY AND RAYTRACING? (advanced).
  In other words: basically Poser transparency and refraction are supposed NOT to be mixed, but in various cases you'll need to. Mixing them will introduce a lot of issues, including a serious slowdown of rendering. All this is discussed in 64 HOW DO I PROPERLY COMBINE TRANSPARENCY AND RAYTRACING?.
- At the moment (Poser versions up till Poser 10 / Pro 2014 SR3) raytraced refraction is in error, as the ray leaving the object towards the camera is bend the wrong way. In real life, a light ray should bent 'forward' when entering the object, 'backward' again when leaving he object, and as a result it should continue its journey parallel to its original path but just shifted in space. Currently in Poser the ray bends in the same direction twice.
  It's said to be repaired in Service Release SR4.

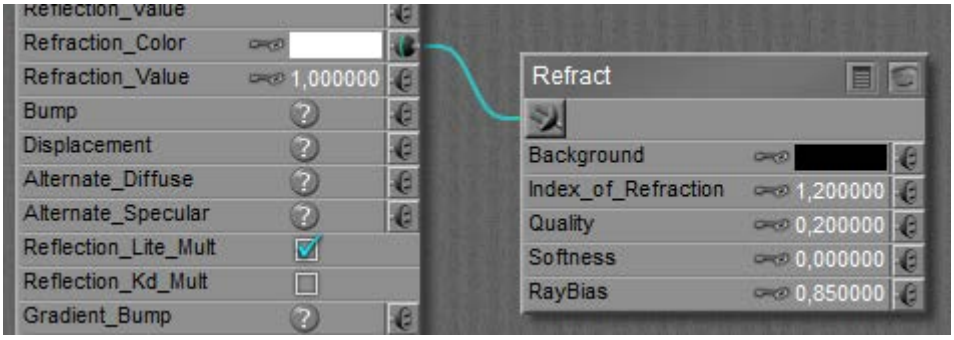
Practical use

I want to use refraction when a material represents liquid or glass, but in real life such transparent materials are reflective as well. And actually, those two phenomena are in a complex balance: the more refractive a material, the more reflective it will be too. On top of that, real life transparency and reflectivity both depend on the angle the camera looks at the surface. The skewer the angle, the more reflective the surface becomes, and the less light is left to pass through. That makes the surface less transparent (and not: less refractive, the bends will be the same).
These combined issues are knows as the "Fresnel effect", and supported in Poser by the Fresnel node (see 63 WHAT'S THE RAYTRACE > FRESNEL NODE USED FOR?) and the Fresnel_Blend node (see 81 WHAT ARE THE MATH NODES USED FOR?).

The right pawn still shows refraction as before, the left pawn shows Fresnel. Especially the upper edge of the left pawn clearly shows that Fresnel is not only transmitting the wall at its back, but also is reflecting it. Those reflections are missing on the right pawn.



About the Refract node parameters:



The **Index Of Refraction** is discussed above. Values for various materials can be found in 32 WHAT'S A PROPER VALUE FOR REFLECTIONS?.

Quality offers a tradeoff between speed and result; high values require longer render times but present crispier results. **Softness** increases the blur of the refracted image, representing irregularities, impurities and even minor movements and vibrations in the refracting surface.

RayBias is a special feature to address a special issue, regarding the refraction on image-based displaced surfaces. RayBias is a distance, in the Poser units as set in my User Interface preferences. The default is 0.3 inch or 0.007620 meters as in the screen grab above.
Poser takes the refracting surface, moves it for this distance outwards from the object, then it calculates the refractions, and then it projects those refractions onto the surface of the object itself. This way, all irregularities of the refracting surface which are smaller than this distance are disregarded, usually these are caused by the details of displacement maps. It speeds up refraction calculations considerably, but it does introduce artifacts and less accurate refractions as a downside. For that reason, one should not have this value larger than the amount of displacement in the same PoserSurface material definition. The 0.3" (0.76cm) default value is quite a lot, actually.

Refraction Color (in the PoserSurface column) gives color to the material, like red to wine or sapphire. But do note that this is a surface effect only, like a colored plastic cover around the object. Poser cannot do volumetrics, the wine will be equally red whatever way I look at it, and the glass will be equally red whatever the thickness.

The contribution of refraction to the total surface definition will be made up from the combination of Refraction_Color * **Refraction_Value**. This does not hold for the refractive effect itself, which is determined by the IoR.
To represent a dark colored material, I can either use a dark color in the swatch, or a low value. Note that Color will be affected by Gamma Correction, the Value will not so 80% White and 100% Value will behave different from 100% White and 80% Value under GC render conditions. For that reason, it's recommended to leave Value at 1.0 and put all adjustments into the color swatch.

Rendering

Raytraced refractions and reflections are realistic, detailed, and although Poser performs them quite efficient they are time and resource consuming at render time.
As a consequence, one should be careful not to put too many raytracing intensive challenges into one scene. InDirect Lighting (IDL) is such a challenge, having a lot of reflective and/or refractive surfaces in one scene is a challenge, having reflections and refraction (and especially Fresnel) on a complex surface is a challenge, and having Max Bounces (and the IDL Quality options) set high in

Render Settings make a challenge as well.
Take the Refraction vs Fresnel image shown above. It's IDL, and reflective wall and floor, and quite high values in Render Settings. Rendering on a fast machine took 3.5 hours. Poser does have its limitations.
There are no meaningful fast alternatives for refraction, like we have image-maps or environmental maps for reflections. Sorry for that.

On top of all those things, Poser raytracing is designed to work with a completely non-transparent surface; refraction caters for (full) transparency on its own. While doing so, raytracing works for objects in the scene only, it cannot handle direct light, nor the shadows or specularity thereof. Mixing raytracing with transparency however might produce unexpected or even erroneous results, while also taking render time to infinity. See 64 HOW DO I PROPERLY COMBINE TRANSPARENCY AND RAYTRACING? on details.

61 What's the Raytrace > Ambient Occlusion node used for?

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

Intermediate

When a direct light shines on an object, the light will get diffused, reflected and transmitted from that object surface. The object will cast a shadow to another object, or to parts of the same object, behind it. Shadow maps are a fast but limited way to derive such direct shadows at rendering, raytraced shadows are slower to derive but more accurate. Those shadows can be blurred somewhat to represent the size of the light source. And they can get a reduced shadow intensity to represent the ambient lighting levels from the environment. But the latter is just a trick to compensate for Posers inabilities to handle such effects from direct lighting only.

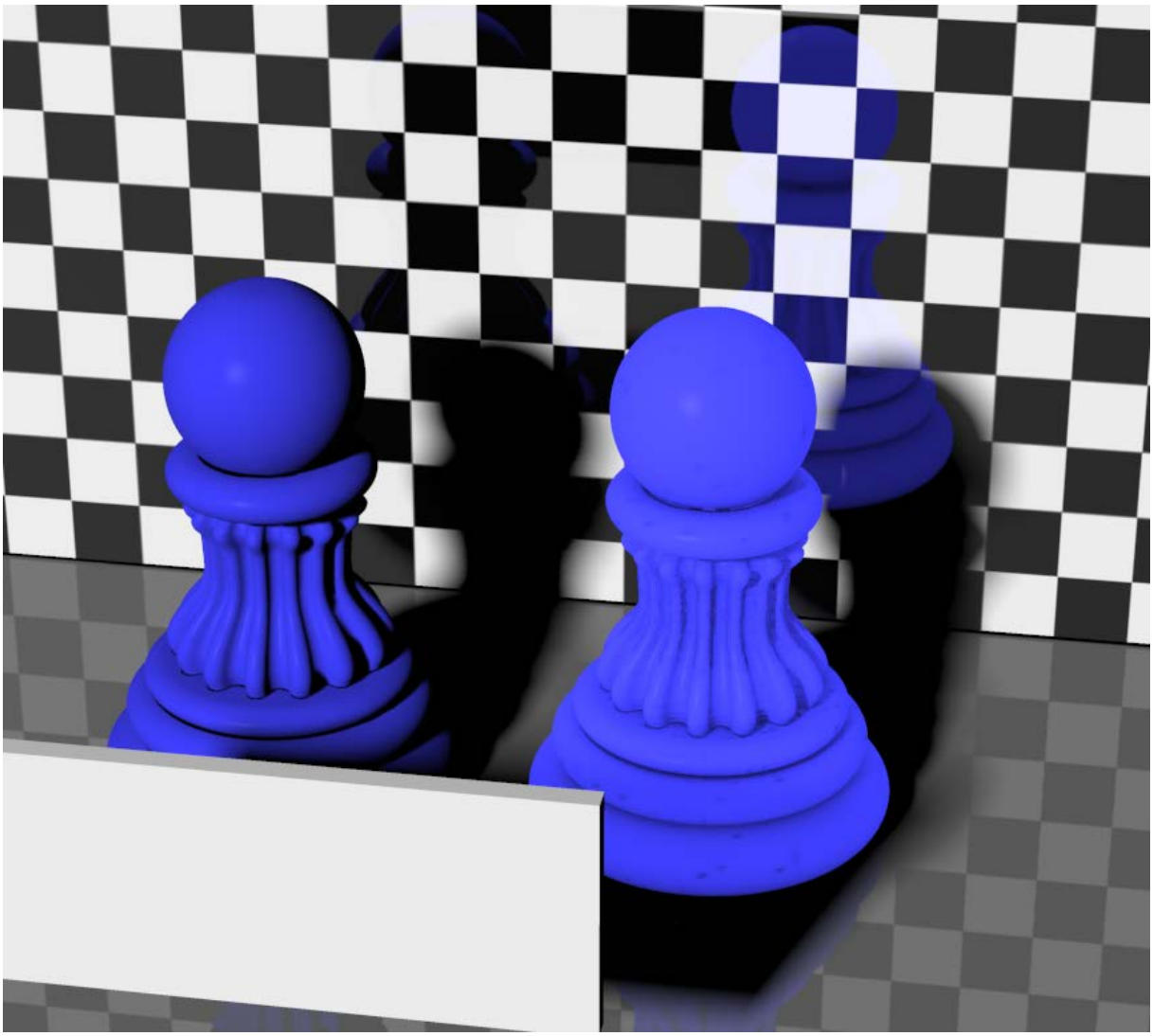
In the meantime, the industry developed an approach to resolve the issue: IBL or Image Based Lighting. A point source in the scene radiates light according to an image map, but in an inverse way: the light rays are not leaving the light to the outside, but travel from the outside towards the light as if a huge sky dome surrounding the scene emits this environmental light. Using a dome object with the image onto it enhanced the illusion of such a surrounding environment, whether it's a sky or a shop interior.

Although that increased the quality of the lighting it did not answer a similar requirement for shadowing. As a result, Ambient Occlusion (AO for short) was born to get better shadows as well. Better as in: reduced by environmental lighting. That was at least some step forward from the (ab)use of Ambient for adjusting the lighting levels in an object specific way.

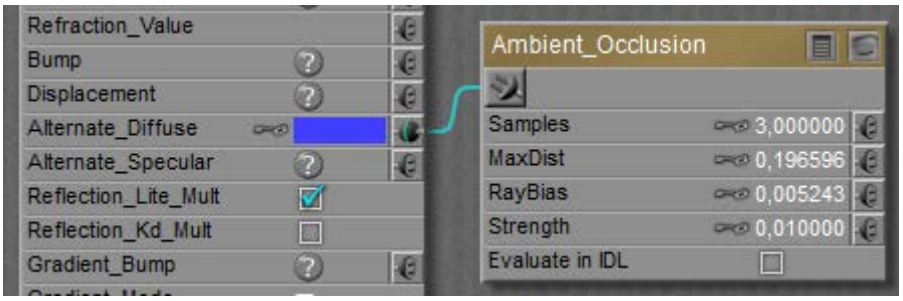
First, AO was implemented in Poser as a surface property which made the response to light dedicated to specific objects in the scene. Again, as an improvement over the use of Ambient for this. Then AO was implemented as a property of the light itself, effectively making the surface property obsolete. And after all, the entire IBL+AO concept was replaced by Indirect Lighting.

So, the Ambient Occlusion surface node is meant for scenes without IDL, without IBL, lit by direct point- spot- or infinite lights only, without AO enabled for those lights.
When the scene gets lit by IBL (which has no shadows and no highlights) and some direct lights as well, AO should be enabled for those lights and the AO materials node should be dropped.
When the scene gets lit by IDL, it should not have IBL lights or any AO materials nodes. Direct lights can give some support (like flashing outdoors, or representing a sun shining in) and their AO properties can help to improve on their shadows even more but should be used with care.

For short: the AO node can be considered outdated, and is available for compatibility only. Use the AO properties of lights instead.



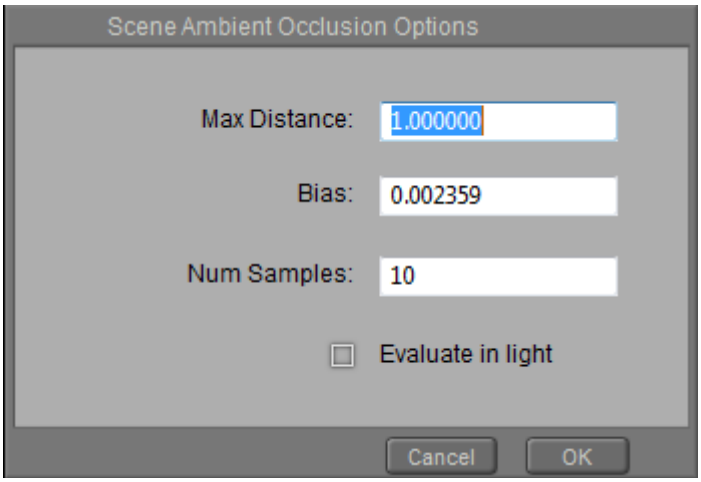
The left pawn shows regular diffuse, the right pawn has a (very limited) amount of AO assigned to the surface. It behaves like it glows a bit, and it hardly picks up shadows when an object blocks the light.



The surface having Ambient Occlusion shoots out rays to scan the neighborhood. **Samples** determines the ray-density (quality), **MaxDist** the maximum length of them. So shadows from surfaces further away are not dealt with. When the surface has displacement maps, **RayBias** will make the first portion of them ignored to avoid time-consuming raytracing on the tiny details. Both MaxWidth and RayBias are in the units I've set in Global Preferences (which is Meters in my case). Altering my preferences will affect the values shown, but of course not the effect itself. When taking values from other sources, I might have to convert for unit differences.

By default, AO is dealing with direct light only which is exactly what it's made for, but by ticking the **Evaluate in IDL**, that can be changed.

As said, instead of using the node on a surface, the use of the light property is preferred.



When clicking the [Scene AO Options] button the generic AO settings unfold in which I recognize all the other AO options from the node. Except that Strength is a per light setting, and the other options are the same for all. Like in IDL lighting which replaces all IBL, AO and more, the effects are the same for all objects in the scene now, as it should be.

62 What's the Raytrace > Gather node used for?

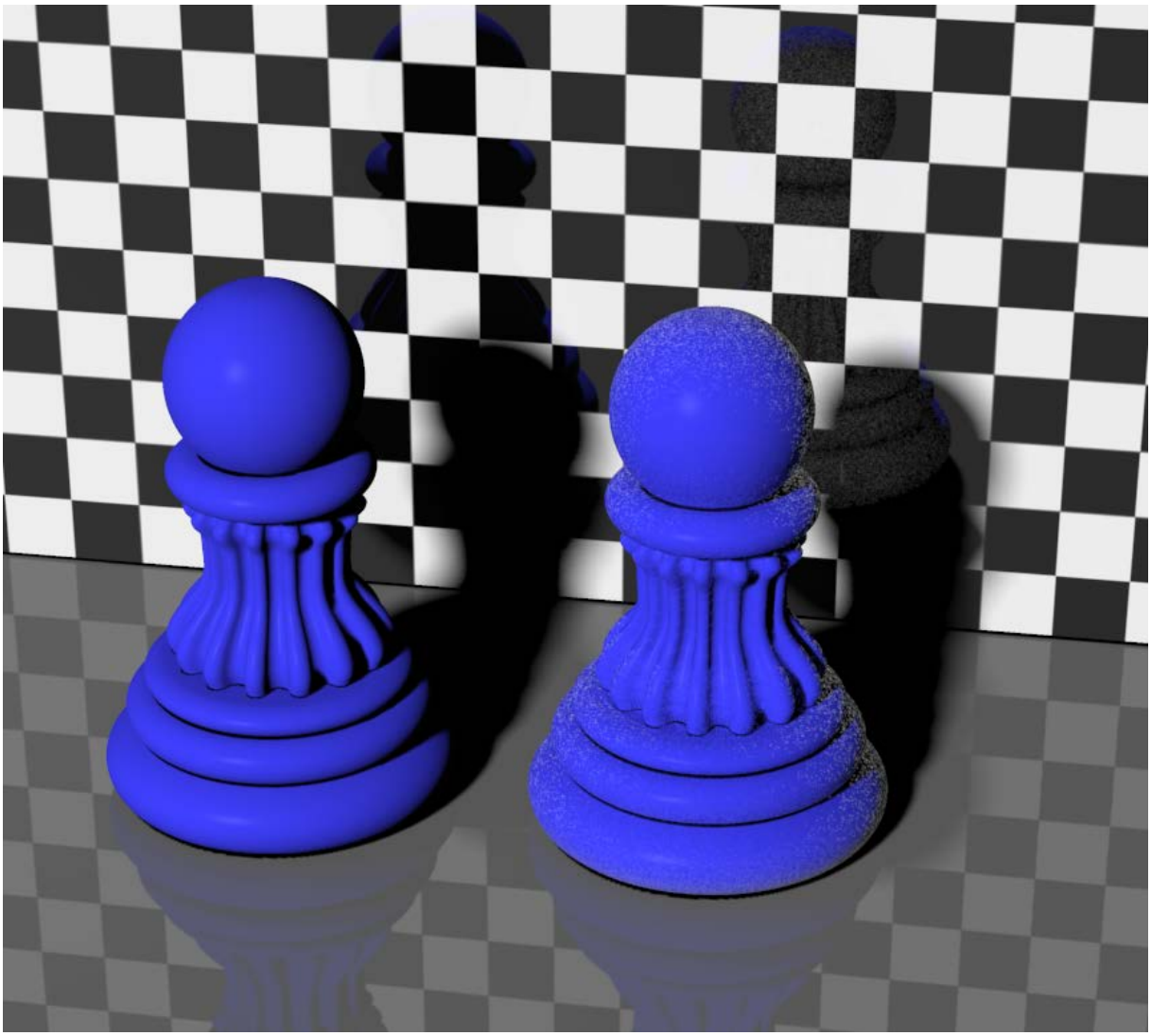
Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

Intermediate

While Ambient Occlusion (see 61 WHAT'S THE RAYTRACE > AMBIENT OCCLUSION NODE USED FOR?) was introduced to handle the softening of shadows due to environmental lighting levels, Gather was introduced to handle radiosity or color bleeding which resulted from being positioned next to a colorful object. It's the red shine a white wall picks up from a bright red ball close to it.

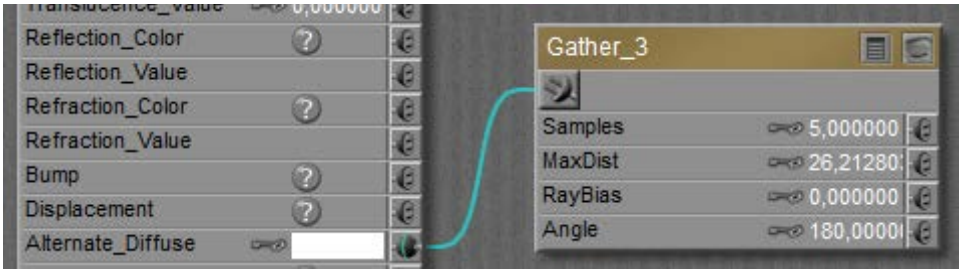
Later on, Indirect Lighting (IDL) took care of this as well. So Gather is meant to support scenes with IBL and direct lighting only, and should be dropped when IDL is switched ON in Render Settings.

In the image below, the left pawn shows regular diffuse while the right pawn offers Gather as well, in an exaggerated way to illustrate the effects. The surface tries to attract light from surrounding objects and wants to bleed their color onto its own surface.



The surface having Gather shoots out rays to scan the neighborhood. **Samples** determines the ray-density (quality), **MaxDist** the maximum length of them. So colors from surfaces further away are not dealt with. When the surface has displacement maps, **RayBias** will make the first portion of them ignored to avoid time-consuming raytracing on the tiny details. Both MaxWidth and RayBias are in

the units I've set in Global Preferences (which is Meters in my case). Altering my preferences will affect the values shown, but of course not the effect itself. When taking values from other sources, I might have to convert for unit differences.



In finding surfaces to pick up colors from, each surface element of the object looks around in outward direction (following its surface normal). **Angle** limits the directions in which it's doing so, 180 means all outward directions.

63 What's the Raytrace > Fresnel node used for?

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

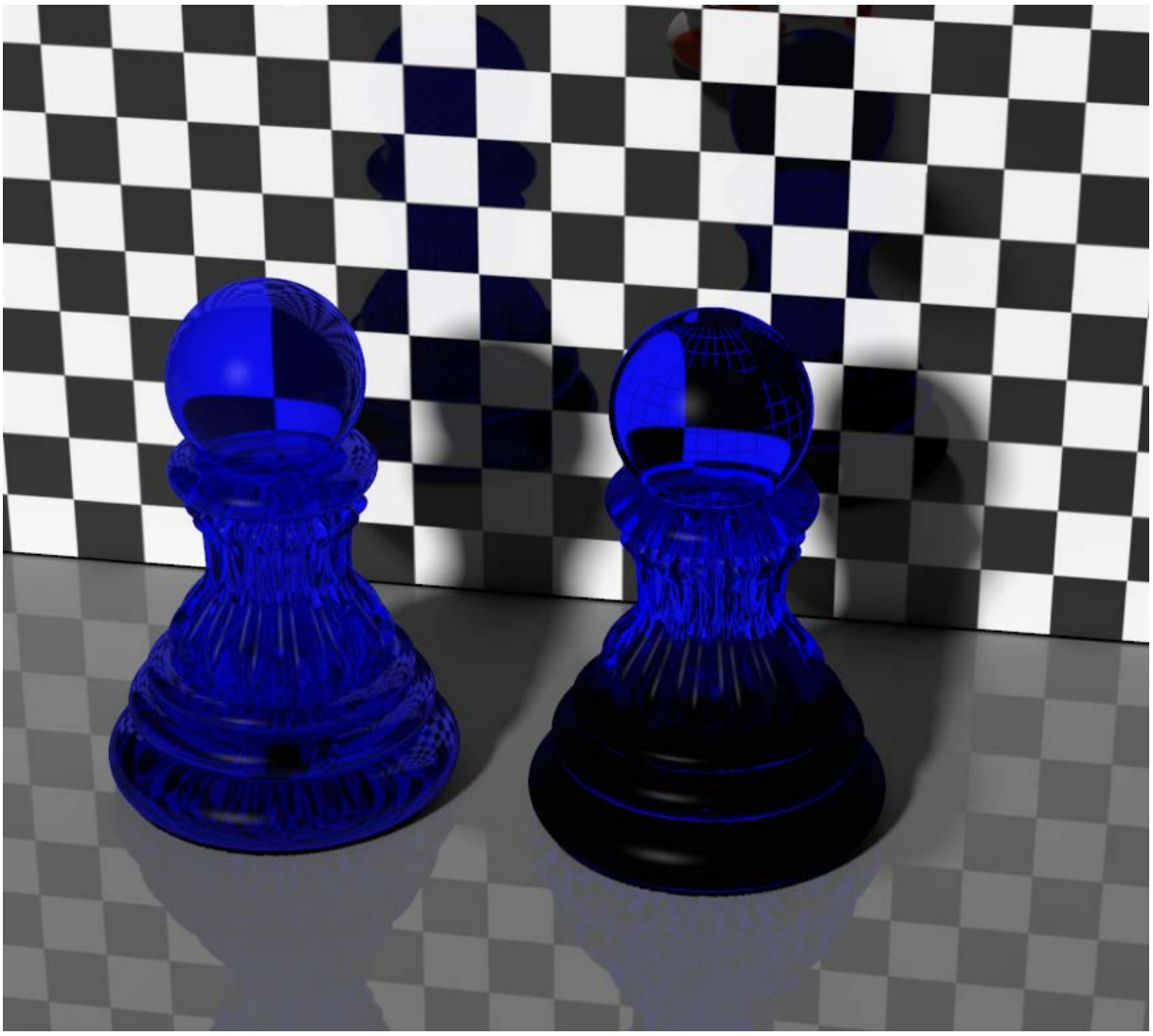
Intermediate

Fresnel is the elaborated combination of reflection with refraction (transparency included), as shown in nature by transparent materials like glass and liquid. Those materials show low reflection and high transparency at angles perpendicular to the surface, towards the camera, and high reflection and low transparency at skew angles towards the camera. The higher the **(Index of) Refraction** of the material, the stronger this effect. On the other hand: the less reflective a material, the more noticeable the effect as an object will become completely reflective at the edges, whatever its IOR.

Reflection makes lightrays bounce from an object surface, to show scene elements between the object and the camera, and behind the camera as well. Transparency makes lightrays pass through the object surface, showing scene elements behind the object (from the cameras point of view). Refraction then makes those latter lightrays bend when passing the surface, and color them too. Transparency is discussed in 08 WHAT'S THE TRANSPARENCY COMPONENT INTENDED FOR?, Reflection in 59 WHAT'S THE RAYTRACE > REFLECT NODE USED FOR? and Refraction in 60 WHAT'S THE RAYTRACE > REFRACT NODE USED FOR?. Do note however that refraction brings its own transparency, and like refraction and reflection, Fresnel is supposed to work with fully non-transparent surfaces. If not, numerous issues have to be dealt with to get any believable result within feasible render times. See 64 HOW DO I PROPERLY COMBINE TRANSPARENCY AND RAYTRACING? on details.

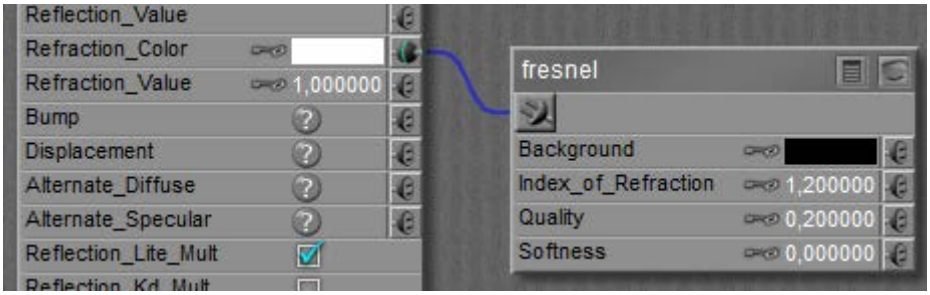
The Fresnel effect is supported by the Fresnel node (discussed here) and the Fresnel_Blend node (see 81 WHAT ARE THE MATH NODES USED FOR?). Like Refraction, Fresnel does require raytracing to be switched ON in Render Settings. The quality of the result depends on the 'Number of Bounces' set in Render Settings as well. This number is a maximum value, when Poser does not need them it won't use them, but if the number of bounces for a light ray exceeds this limit, this light ray is killed. This might speed up the rendering while it also might introduce artifacts (black spots) in the result. The tradeoff is mine, but as nature has an infinite number of bounces, the max value is the best when I can afford it.

Do note that Fresnel – like Refraction - only handles the light from objects in the surrounding scene. It does not cater for the rays from direct light sources (spot, point, etc. lights), these will not get bend at all. Fresnel does not let light pass through an object shining onto another object, the object will be opaque for direct light, and shadows will be dark as a result. But Transparency behaves as expected, as discussed in 08 WHAT'S THE TRANSPARENCY COMPONENT INTENDED FOR? (basic) and 64 HOW DO I PROPERLY COMBINE TRANSPARENCY AND RAYTRACING? (advanced).



The right pawn still shows refraction as before, the left pawn shows Fresnel. Especially the upper edge of the left pawn clearly shows that Fresnel is not only transmitting the wall at its back, but also is reflecting it. These reflections are missing on the right pawn.

About the Fresnel node parameters:



The **Index Of Refraction** is mentioned above. Values for various materials can be found in 32 WHAT'S A PROPER VALUE FOR REFLECTIONS?.

Quality offers a tradeoff between speed and result; high values require longer render times but present crispier results. **Softness** increases the blur of the refracted image, representing irregularities, impurities and even minor movements and vibrations in the refracting surface.

Background is meant to fill in the pixels where no scene elements are around to be reflected, but it should be used with care because this idea is pretty meaningless for transparency / refraction. Otherwise I'm looking through a transparent surface, seeing things which are not there at all.

Refraction Color (in the PoserSurface column) gives color to the material, like red to wine or sapphire. But do note that this is a surface effect only, like a colored plastic cover around the object. Poser cannot do volumetrics, the wine will be equally red whatever way I look at it, and glass will be equally red whatever the thickness.

The final amount of refraction will be made up from the combination of Refraction_Color * **Refraction_Value**. This holds for the surface color as well as for the refractive and reflective effects themselves. To represent a dark colored material, I can either use a dark color in the swatch, or a low value. The Color will be affected by Gamma Correction, the Value will not so 80% White and 100% Value will behave different from 100% White and 80% Value under GC render conditions.

Rendering

Although reflection and refraction themselves do have another parameter, RayBias, in common, that one is missing here. This RayBias is introduced to avoid undesired optical effects from tiny surface irregularities, induced by displacement maps. Think of scars etc. on skin. Using RayBias speeds up the rendering but might introduce artifacts when set too high.
So… Fresnel is more accurate at the tiny surface details, will not have the artifacts, but certainly will suffer from low render speeds when applied to a displacement-mapped surface.

Raytraced refractions and reflections are (sort of) realistic, detailed, and therefore time and resource consuming at render time. As a consequence, one should be careful not to put too many raytracing intensive challenges into one scene, otherwise the rendering will take forever. InDirect Lighting (IDL) is such a challenge, Fresnel – combining reflection and refraction is such a challenge, having a lot of reflective and/or refractive surfaces in one scene is a challenge, having reflections, refraction (and especially Fresnel) on a complex surface is a challenge, and having Max Bounces (and the IDL Quality options) set high in Render Settings make a challenge as well.

Take the Refraction vs Fresnel image shown above. It's IDL, and reflective wall and floor, and quite high values in Render Settings. Rendering took 3.5 hours on quite a fast PC. Poser does have its limitations.

On top of all those things, raytracing is designed to work with a completely non-transparent surface; refraction caters for (full) transparency on its own. While doing so, raytracing works for objects in the scene only, it cannot handle direct light, nor the shadows or specularity thereof. Mixing raytracing with transparency however might produce unexpected or even erroneous results, while also taking render time to infinity. See 64 How do I properly combine Transparency and Raytracing? on details.

64 How do I properly combine Transparency and Raytracing?

Raytracing nodes like reflection, refraction and Fresnel work well for fully non-transparent surfaces. Refraction and Fresnel bring in full transparency on their own. As a consequence, those materials behave opaque to direct lighting, and produce dark shadows and block specularity as well. I might not want that.

Throwing in transparency however does have serious pitfalls, especially when combined with raytracing. Pitfalls and some ways to climb out of them are discussed here.
Before doing so, it's worthwhile pointing out that in Poser, transparency might mean:

- Lace-like transparency. The surface has holes where the light shines through completely, and opaque areas with full diffuse, specular, reflective etc. properties. This kind of transparency is not discussed here, just set Poser transparency to 1 and use image maps to define the present and absent areas of the surface; then all will be fine.
- Surface transparency. Imagine the object made of clear glass covered with a layer that absorbs light to a limited extend. This is what Poser has in mind when the material transparency value is reduced. 90% means that 10% of the light is blocked by that surface layer, so 90% x 90% = 81% of the light will pass the object (first object in, then out).
- Volume transparency, the object is made from non-clear glass and light is dimmed and colored when it passes through, making the final result depend on object thickness and shape as well. This is **not** what Poser has in mind but one can mimic the dimming effect, for instance by translating an object transparency of 81% to a Poser surface transparency value of 90%. That assumes that the object has equal thickness all over. And when that's not the case, Poser offers Transparency_Edge and Transparency_Falloff to mimic the effects of that.

Advanced

Transparency and Reflection

As discussed in 34 DO RENDER SETTINGS AFFECT THE BEHAVIOR OF MATERIALS? Poser raytracing does a fine and efficient job, when the scene presents a limited amount of bounces between objects. Parallel mirrors, fields loaded with mutually reflecting Christmas balls and more like that might either take large amounts of render time, or might produce artefacts when the rays are killed too early.

Exactly that happens when an object becomes both transparent and reflective. Light passes the object at the front, bounces at the back, bounces at the front and so on resulting in an infinite amount of internal reflections. This can prolong render times considerably, while reducing those with low(er) values of Raytrace Bounces in Render Settings might cripple the results somewhat. Which means that I've got a challenge in finding the proper balance between time and quality.

Because light not only bounces from the surface at the first hit, but also gets light added from the bounces following on that, the final brightness of the reflection increases. This is according to real life, but unfortunately the dimming of the light during those internal bounces is not. Transparency in Poser is a surface effect, while it's something volumetric in nature. The thicker the object, or the more bounces have taken place, the more light is absorbed on the way and the more the light is dimmed.

In math:

- the Poser surface has a transparency T, which means that each time a ray hits the surface, that percentage of the light (say T=30%) will pass through. Then only 1-T (say 1-30% = 70%) remains for initial reflection, and reflectivity R can't get larger than that. So if the surface is covered with a thin metallic foil which lets 30% of the light pass through, and that metal has

a reflectivity by itself of 80%, then the surface will reflect 70% x 80% = 56% of the light initially.
Sounds simple, but I have to adjust the reflectivity from 80% down to 56% to cater for the transparency, as Poser is not doing that when I use the Reflection component I in PoserSurface. Plugging the Reflect nod into Alternate_Diffuse however handles this for me, as (only!) Diffuse and Alternate_Diffuse are affected by the Transparency setting. In that case, I can simply use the 80% reflectivity as well.

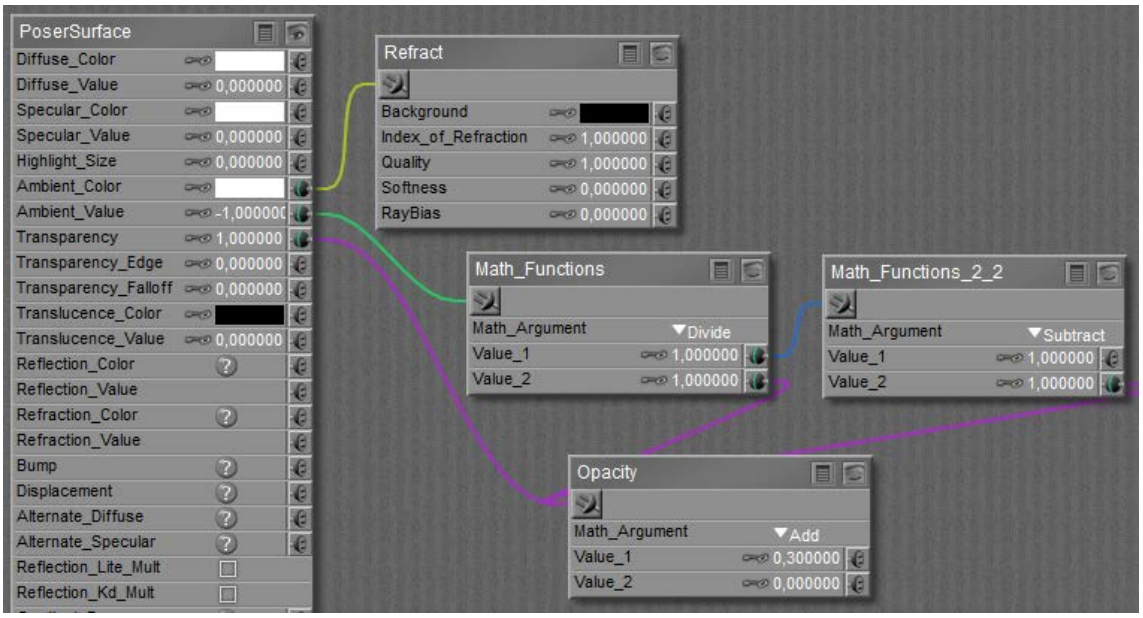
- Due to the internal reflections, the amount of light that will come out at the front, the final reflection, will be R $(1+ T^2/(1- R^2))$ like $0.56 * (1+ 0.3^2/(1-0.56^2)) = 0.633$ which is a serious increase over the initial 0.56. This implies that there are no simple relationships between lighting levels measured in real life, and the Transparency settings for Poser.
- In real life, the output would have been R $(1+ T^2/(1-(RT)^2))$ like $0.56*(1+0,3^2/(1-(0.56*0.3)^2)) = 0.611$ which is slightly less. So despite the large render time due to internal reflections, Poser is not doing so bad at all, the difference between the volume-model and the surface-model is something I can live with. Or I could just use a slightly less reflectivity instead (55% instead of 56% might do).

Note:
Poser, from version 10 up and in all Pro versions, offers a Gamma Correction mechanism to enhance the photorealism of render results. Much recommended, see 98 WHAT'S THE FUZZ ABOUT GAMMA CORRECTION? for details. As reflected light is considered an independent component in the surface definitions, the gamma mechanism should be applied to the Reflect settings. That is: all Values should remain at 1.0 (or 0.0, but no intermediate values), any reduction or coloring in reflectivity should be embedded in color swatches, and any image map involved should have set its Gamma set to Render Gamma or alike.
However, Transparency itself is considered a 'blender': more of one component (say foreground reflecting) implies less of the other (background shining through). For those elements, the Gamma mechanism should be avoided or bypassed whenever possible. That is: Color swatches should remain white (or black), all intermediate values should be in a Value slot, and any images involved should have their Gamma value set to 1.0 if applicable. See 99 GAMMA, TRANSPARENCY AND BLENDS for details.

Transparency and Refraction

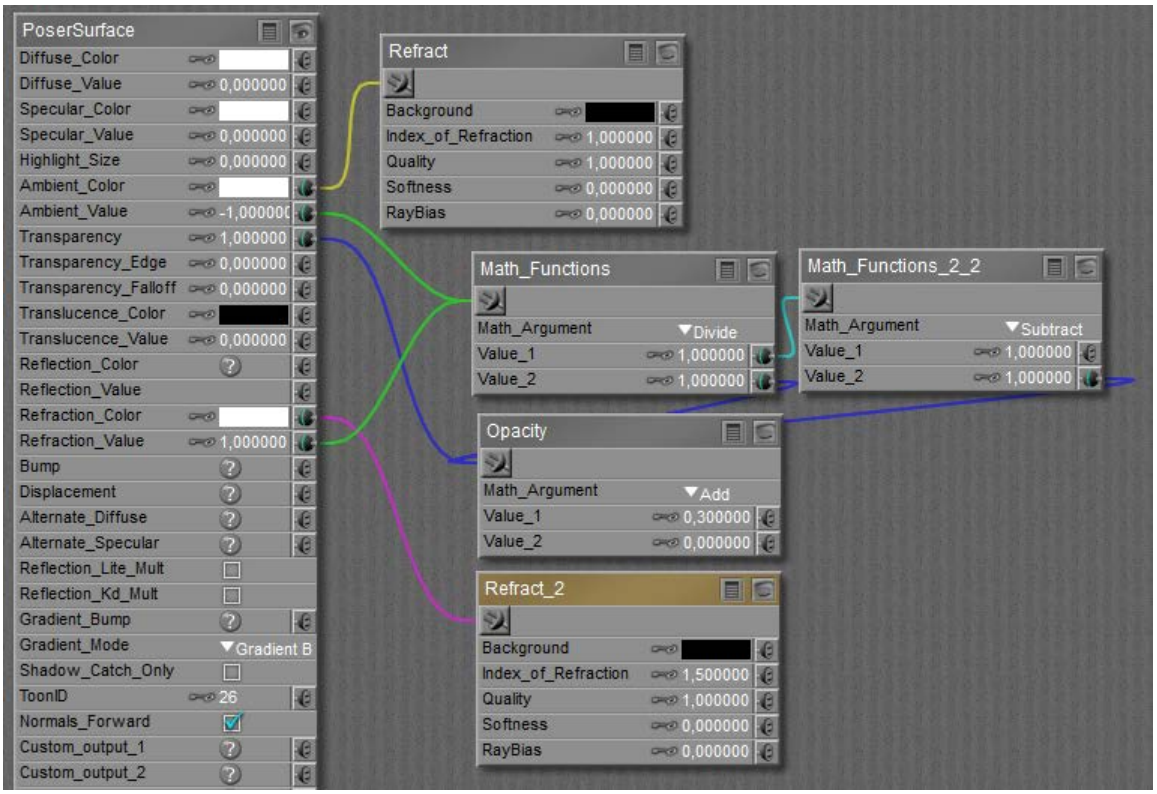
The all raytracing, including the Refraction node as discussed in 60 WHAT'S THE RAYTRACE > REFRACT NODE USED FOR? does work on indirect light from objects only. Direct light, whether diffuse or specular, is not reflected nor refracted. Such light, passing a refractive object, makes deep shadows and can't make any highlights behind the object anymore. When I want the proper bright shadows of transparent objects, I need to use Transparency instead.

But now the scene behind the object will shine through twice: once due to transparency and one due to refraction, while I want the latter only since glasseous objects and liquids to refract (bend and shift the scene behind the object), while transparency does not. So I have to take the Indirect Light portion out of the Transparency, and this is the way to do it:



By using an IoR of 1.0 refraction is transparent only, no bending involved, and applies to indirect light only. The transparency itself applies to direct and indirect light, and so the subtraction results in the direct light portion only. The amount of refraction to be used here is proportional to the amount of transparency, but unfortunately the transparency slot turns into Opacity the moment a node gets plugged in. Then, black or low values make transparent and the value 0.3 shown above indicates a 70% transparency.

Now I can add in the required refraction (additional Refract node), with an IoR related to the material at hand, say 1.5. Again, the refraction is proportional to the amount of transparency so I can re-use that function outcome:



Transparency and Fresnel

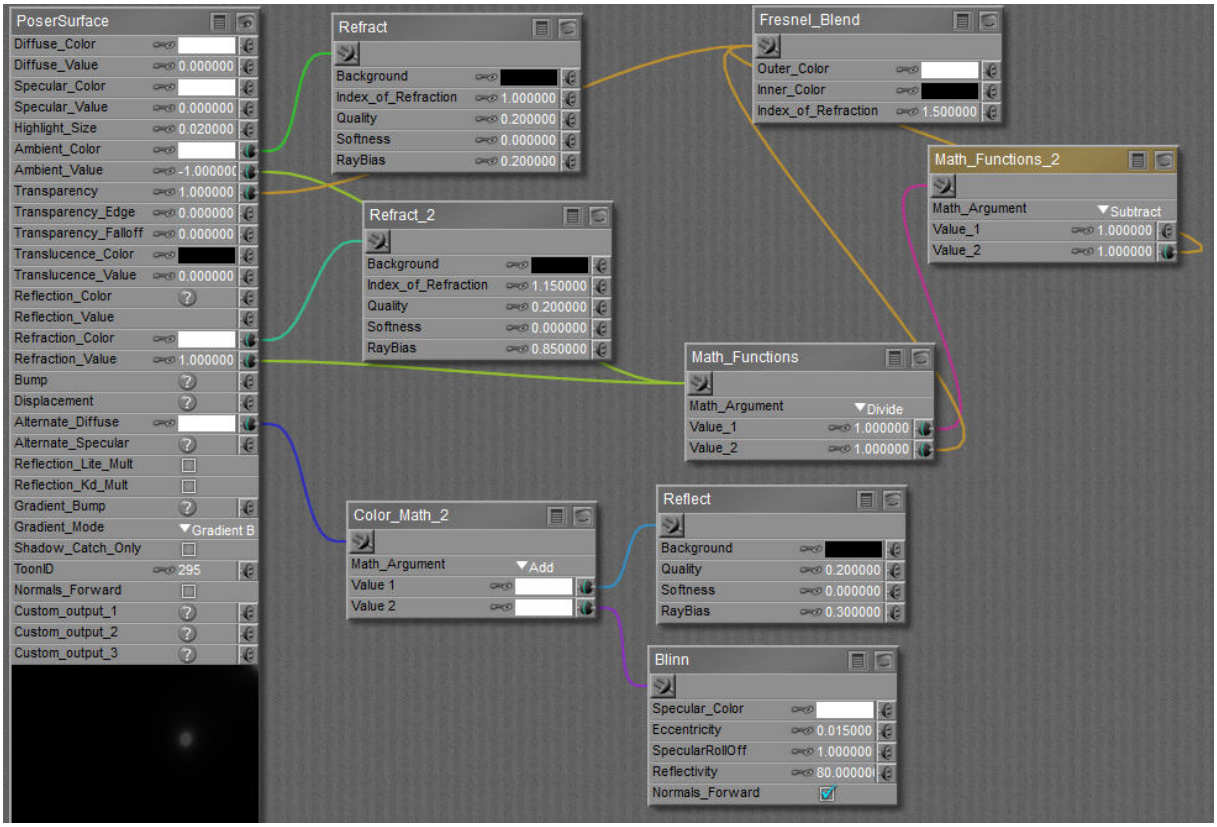
In real life, at skew angles it gets harder for light waves to enter a material. So transparency decreases, and since the energy has to go somewhere, reflectivity increases. This is understood as the Fresnel principle.

The Fresnel node itself, or any simple combination of refraction with Fresnel_Blend, has the same (dis)advantages as a straightforward refraction: it assumes a full transparent clear object, which then behaves as opaque for direct light and produces far too hard shadows. So I'll elaborate on the previous way of work instead.

First, the fixed value for opacity can be replaced by a Fresnel_Blend node. This node has to represent opacity too, and so it has to produce high (white) values at skew angles and low (black) ones at the inside of the object.

Second, I add Reflect, and I'll do so in the Alternate_Diffuse slot which gets adjusted for transparency automatically (like regular Diffuse).And since reflection caters for indirect light only I'll add its equivalent for direct light: specular. In which case the Blinn node provides the best way forward.

So now I've got the definition for homogeneous glass or a liquid, uncolored but with a limited transparency, as presented by Bagginsbill (Renderosity forum, March 20, 2014):



For colored glass or liquid, the Refraction_Color can be adjusted. No other swatches need adjustment, as reflection and specularity go uncolored (except for metals, which are non-transparent) and Transparency goes uncolored as well, so the first Refract (added into Ambient) for compensation should stay uncolored as well: Ambient_Color remains white.

65 What's the Environment Map > Spherical Map node used for?

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

Intermediate

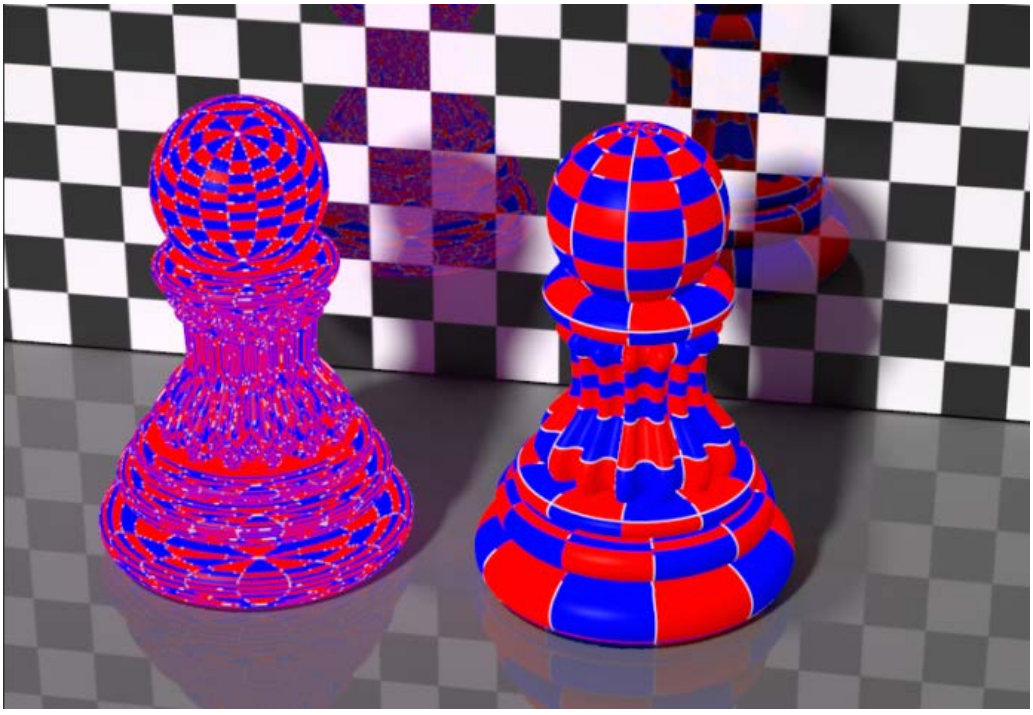
In order to show some kind of reflections in the render, the scene needs an environment to reflect. This might require the build of a complete scene behind the camera, which won't show except for its reflections. As this can be quite tedious and far from cost-effective, one can use image maps instead. For just mimicking blurred reflections of far-away objects, skies and landscapes any simple projection (mapping) of an appropriate image onto the reflection component will do. This however should be considered unsatisfactory when sharper reflections have to be considered.

Now, say, my scene is under a sky dome and on a ground floor, and an image is mapped to these environmental objects. In the scene, a single fully reflective object is present. Then: how would the dome and ground be reflected from that object (assuming there are no other things around to reflect)? Rendering full, crisp raytraced reflections from a complex shaped object can be time consuming.

This is when applying a "spherical environmental reflection map" becomes useful. Just plug the Sphere Map (the only node in the Lighting > Environment Map group) into the PoserSurface Reflection_Color (or Alternate_Diffuse) slot, and the image that might have been used for the sky dome into the Sphere-map Color slot. Now I'll see the same reflections, but I don't need the dome, and neither I need any raytracing for the reflections.

I might want to **Color** filter the image though, and I might want to **Rotate** it to match any sky dome actually used.

Now, is this any different from a regular mapping of the image around the object? Sure it is, have a look at this:



The right pawn has the tiling image mapped onto the object surface, in the Reflection or (alternate) diffuse component of the PoserSurface. The left pawn shows environmental spherical mapping, and looks like a mess at first sight. Unless I realize that the tiling image is mapped onto a (virtual) sky dome surrounding the scene, like the way the tiles are mapped onto the head ball of the right pawn. The 'converging point' where the tiles come together is not on the object itself, but somewhere straight above all objects, high in the sky.

Then such a colored sky dome is reflected by the left pawn, towards the camera. And that is what the spherical mapped texture is showing, in a correct way. Where image mapping usually either depends on the shape of the object (UV mapping) or the position of the object in the scene (XYZ mapping), this Environmental Spherical Reflection Mapping depends on the position of the object under the dome, relative to the camera. When either the object moves, or the camera moves (or both), the mapping will get adjusted.

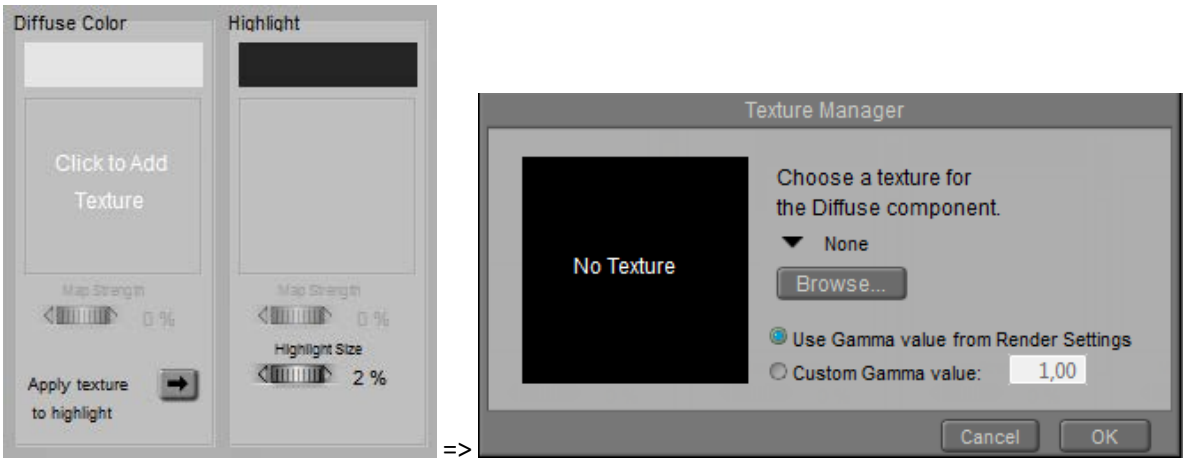
Do I need it?

The obvious advantage is render speed, the obvious disadvantage is: it does not reflect any objects in the scene, let alone portions of the same object, since these are not in the image used. So, for stills of filled scenes rendered on modern, fast PC's deploying IDL illumination and other raytracing demanding approaches, the use of the regular Reflect (or even Fresnel) node might be a better way.

But for those "shiny car on an empty road" advert-like animations, deploying this environmental spherical reflection mapping might be a game winner. And it might serve well in test runs too.

66 How can I assign an image to a material?

Assigning an (external) image to a material is quite a common action, whether it's for light gels, background, or object surfaces. In Material Room, the Simple interface requires one click to open the Texture Manager window



which lets me pick an image file (many formats supported) and (for Poser 10 / Pro 2010 and up) a Gamma value. As a rule of thumb:

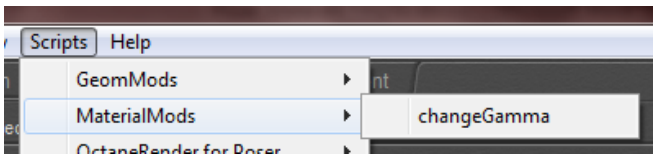
- When the component represents some kind of light towards the camera, which has to be added up to the light from other components, then the first (default) option is the right one. Think: (Alternate_)Diffuse, (Alternate_)Specular, Ambient and Translucence, Reflection and Refraction.
- When the image represents just an amount of something and color is far from relevant (think: Bump/Displacement, Transparency, Highlight Size, …), then the second option should be selected with 1.0 as the right value.

Just a brief explanation for this: Gamma Correction, as discussed in 20 A POSERSURFACE MATERIAL OFFERS DIFFUSE, SPECULAR, ETC. HOW DO THESE WORK TOGETHER?, darkens the image before it's applied and brightens the final render result for compensation. This brightens up the shadows to mimic the effects of radiosity from floors, walls , ceilings and the like, and from the scattering of light in a real, somewhat dusty atmosphere; effects which are hardly available in a Poser render. This Gamma mechanism also reduces the (risk of) overlighting when multiple sources of light (diffuse, specular, reflection, …) from a surface towards the camera are combined. Poser just adds them up and clips at 100%, while in real life our eyes adjust to the increased light levels.

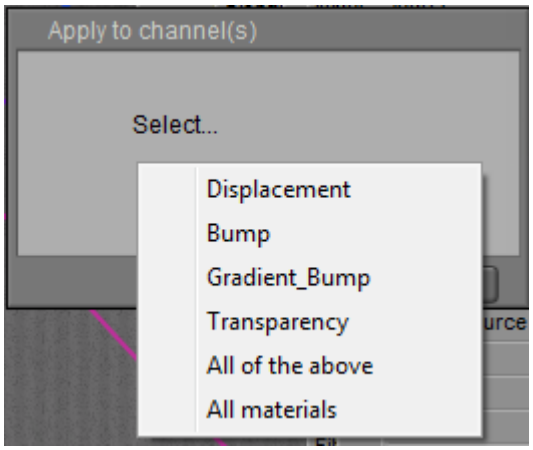
But for the greyscale image-driven amounts, the render consequences suffer from the distortion: reduced displacements for instance cannot be compensated for in the final brightening pass. So, for these effects, the Gamma Correction has to be disabled by overriding the default value with a neutral 1.0.

Image management

When I already have various images in my scene, and when I do not want to recheck and reset all of them, Poser can help me with various scripts:



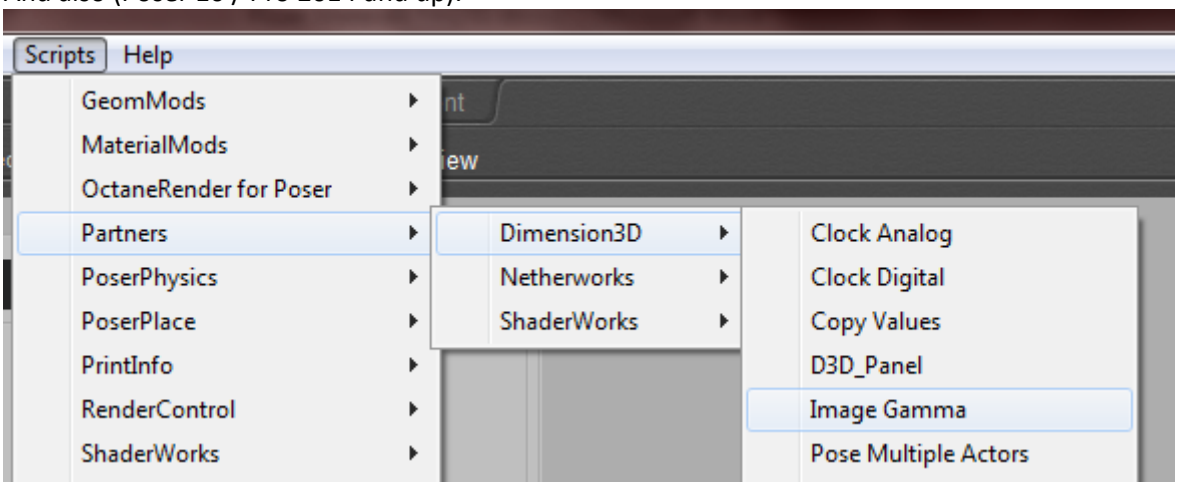
Which leads me to …



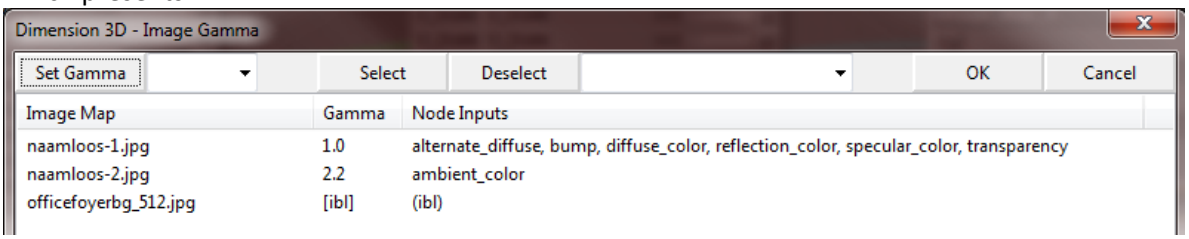
where “All of the above” is the recommended option.

Note that images driving Edge_Blend and nodes alike have to be dealt with manually; these also should have their Gamma value set to 1.0.

And also (Poser 10 / Pro 2014 and up):



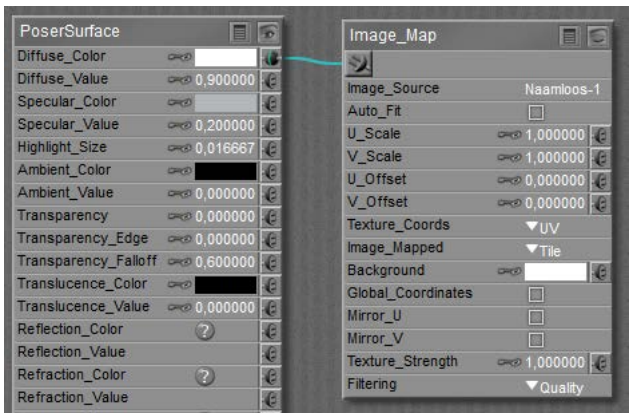
which presents:



Warning: when a single image file is referred to in multiple places, for multiple material definition components, and/or for various objects, then the image is imported and handled into Poser only **ONCE**. It’s this single instance which gets a Gamma value assigned, whatever way I choose. Changing the value at one place alters it in all other places, for that image.

Intermediate

In the Advanced interface to the Poser Material Room, nodes are the essential building blocks. They are the graphical representation of mathematical function calls, calculation procedures turning parameters (inputs) to a result (output).



For applying images, the Image_map node can be found in the 2D Textures group, and reads like …

Clicking **Image_Source** opens the Texture Manager as discussed above, and enables me to assign an image file and a Gamma value. As said, addressing the same file in multiple nodes will still associate

ONE gamma value to that value. When I use various values in the nodes, then the last one entered is the final value for all of them.

The next parameters (**U/V_Scale**, **U/V_Offset**, **Texture_Coords**, **Image_Mapped** plus the **Global_Coordinates** checkbox) all affect the mapping of the image onto the object surface. Just consult the manual for the meaning of the various options.
Checking Global_Coordinates is meaningful in combination with the XY, XZ, YZ options of Texture_Coords, while the **Mirror_U/V** options are meaningful in combination with the Tile option from Image_Mapped: the images is tiled and the tiles are flipped successively.

For all places where no image pixel is available (due to the mapping parameters), the Background will be used instead. When Image_Mapped is set to Alpha then any transparency information from the image itself is used for this too, and Background might fill in some spots within the textured area as well.

Texture Strength works as follows: say the image map is connected to a color-swatch in PoserSurface. Then first the image is merged with white: Strength 100% means full image, 0% means no image at all. Second the result is filtered by the color-swatch, as usual. In other words, for an 80% Texture Strength, the final result consists of 80% color-filtered image, and for the remaining 20% of the color from the swatch itself.

Filtering comes into play at render time, when (usually) multiple pixels from the images become associated with a single pixel in the render result, especially when the rendering uses anti-aliasing itself (that's Pixel Samples in the Render Settings). Or the other way around, and a single pixel in the image is used to fill multiple pixels in the result.
Just that image pixel can be used ('None' for filtering) or info from its direct neighbors can be used as well ('Fast'). The latter is fine for test renders, and for small sized animations for the web (say 640x480, regular YouTube). Looking one step further ('Quality') is the default, and recommended for larger animations (HD, DVD quality) and still images for the web. Looking another step further ('Crisp') takes even more pixels from the image into consideration, and might be of use for large scale, print-oriented render results. As usual, the higher the setting, the more memory and render time are required to finish the process.

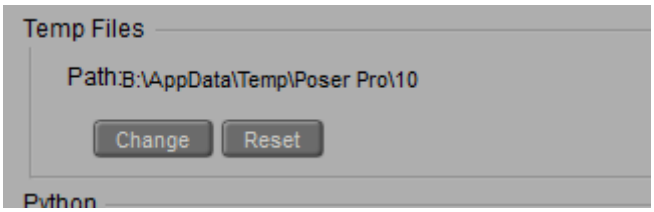
67 How does Poser handle my images for texturing?

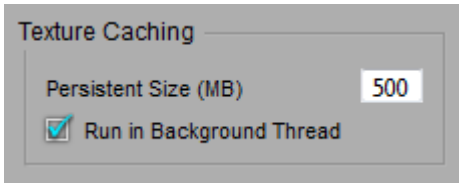
For rendering purposes, Poser internally works with 16-bits-per color (High Definition) inputs and results. For display and for most image exports or the render result, a translation is made to the (Low Definition, 8-bits-per-color) image formats like JPG, PNG, etc. In Poser Pro, export to a High Definition format (HDR, EXR) is possible.

For input, most images will be in Low Definition format as well, usually JPG from photographs. These are fine for preview, but for rendering they are translated first to EXR format, and saved at a temporary place

(Poser Temp Folder)\PoserTextureCache

where the Poser Temp Folder is set in the Edit > General Preferences menu, Misc Tab

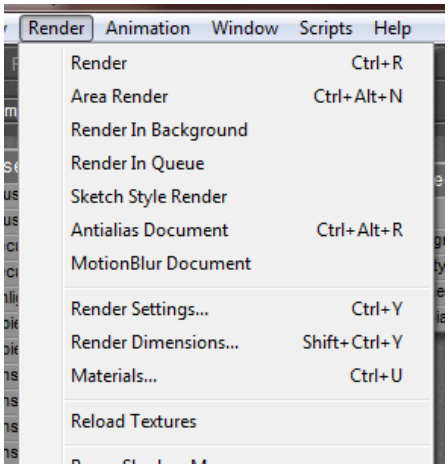




More handling details are managed through the Render tab in General Preferences

After an image is assigned, or when a material with image references is assigned to a surface or an object is loaded with such material definitions, the translation from Low to High Definition is made as a first pass of the render process. Unless I have the **Run in Background Thread** option checked (it's ON by default), then Poser saves me the waits and utilizes spare CPU capacity when available.

The Texture Cache is filled up while building the scene and rendering, and cleared when Poser exits (in a regular way). Except for the <so many> MB's of most recently added images. This way, some translation is avoided when I reopen Poser to continue my work on the same scene. This Persistent Size can be set as well.



Note that an EXR file requires about 7 bytes per pixel, which is 5 to 10 times as large as the JPG's they originate from. So a single hires (4000x4000) image as used for most character skins takes about 100MB disk space. This space requirement is something to keep in mind when setting Persistent Size, and of course all the required space for keeping all images from my entire scene needs to be available when the rendering process kicks in.

When an image is already translated into the Texture Cache, and then is modified in Photoshop or alike, such a modification will go undetected by Poser and the stack of available EXR files need to be refreshed. I have to do so manually, using the Render > Reload Textures menu.

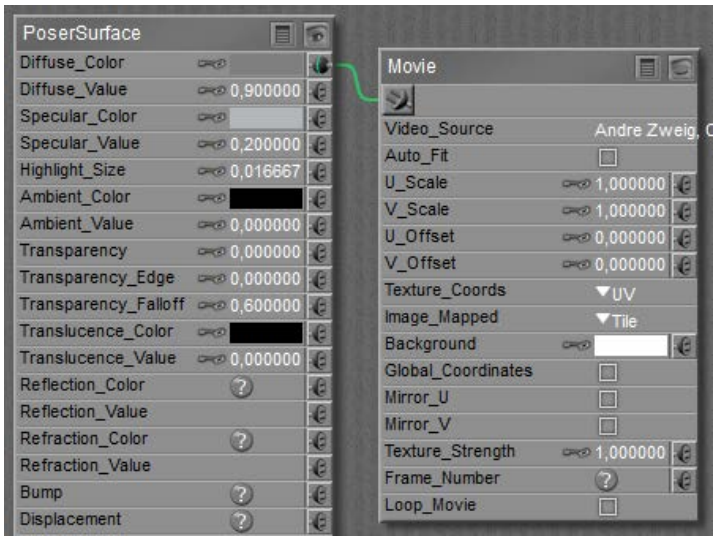
You might be interested in 69 How does Poser handle my movies for texturing?, on handling movie files.

68 How can I assign a movie to a material?

From the Simple interface to Material Room, there is no real difference between assigning a still image, or a movie. So, consult 66 HOW CAN I ASSIGN AN IMAGE TO A MATERIAL? on images first, and when selecting a file via Texture Manager, just select the appropriate movie file. Various formats are supported, somewhat depending on the Operating System and on the video codecs installed.

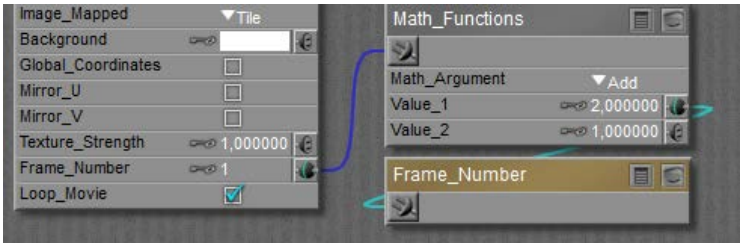
Intermediate

In the Advanced interface to the Poser Material Room, nodes are the essential building blocks. They are the graphical representation of mathematical function calls, calculation procedures turning parameters (inputs) to a result (output). For applying movies, the Movie node can be found in the 2D Textures group, and reads like …



Note: when a still image is assigned via the Simple interface, switching to Advanced will show an Image_map node attached. When via the Simple interface a movie file is assigned instead, switching to Advanced will show a Movie node.

When comparing the Movie and Image_Map nodes, I'll notice that most parameters are similar. But the Movie node lacks filtering, as 'None' out of these is applied. And movies do have frames, like my animation. Without any further steps, both are just matched so frame 1 from the movie will be applied in frame 1 of my render, and so on. But the node offers the possibility to add some math into it, so I can let the movie run faster, or start ahead, of the rendered animation. As in



Where (Movie Frame) = 1* (2 * Scene Frame + 1)

As can be expected, **Frame_Number** in the Movie node refers to the frame in the movie, while the Frame_Number node itself (from the Variables group) refers to the rendered frame in the scene. And when the Movie is not long enough to deliver the required frame, it can start all over again. But that requires that the **Loop_Movie** checkbox is ON. Which it is by default.

69 How does Poser handle my movies for texturing?

In principle, a movie is just a series of images. So – in principle again – a movie is dealt with the same way as an image, as discussed in 67 HOW DOES POSER HANDLE MY IMAGES FOR TEXTURING?. With a few differences.

As the movie file itself is not an image, and such an image file is required for preview, the image is extracted into the Texture Cache folder (in PNG format). And, when the **Run in Background Thread** option checked, the EXR file is created at about that moment too instead of at render time.

Now, which frame is extracted at what moment? All frames of the entire movie at the start? No.

The respective image files are extracted when they are requested for. This might be in preview, when I loop through the animation. Or it might be at render time, when each and every frame is dealt with. And in case only the even or odd frames are requested for (when the movie frame numbers follow some formula, or just a limited amount of frames is rendered anyway as set in the Movie Settings tab (Increment \ Every Nth Frame) in Render Settings) then only those are extracted.

Using movies for texturing certainly will load the Texture Cache disk space. The good news is: the files don't need (and don't have) the large resolutions which are required for high end stills. In most cases a 640x480 size might do, and full HD (1980x1024) can be considered rare for texture input.

The bad news is: one needs a PNG as well as an EXR. That's about 4.0MB for 640x480 or 16MB for full HD. But more relevant, I might need a lot of them. A 10 sec animation makes 300 frames, which are 4.8 GB on full HD frames alone in my Texture Cache. So it might take a while to generate all those files, and I'd better be sure I've got the space available when I push the Render button.

Note that when I use the Render > Reload Textures menu, the entire Texture Cache will be cleared and reloaded. This will regenerate the EXR's for all static images, but will only extract the PNG (and make the EXR) for the movie frame required for the Preview. All other frames will not be generated until requested for.

IV Advanced Surface Definitions

The articles in this section discuss all Material Room nodes required for either procedural textures, or explicitly aimed at node-tree building.

A procedural texture is not derived from an (eventually color filtered) external image or movie still, but is mathematically generated internally from surface or spatial coordinates. The nodes to accomplish such textures can be found in the 2D Textures (70 WHAT ARE THE 2D SURFACE TEXTURE BUILDING BLOCKS USED FOR? and 3D Textures groups (71 WHAT ARE THE 3D SURFACE TEXTURE BUILDING BLOCKS USED FOR? and on).

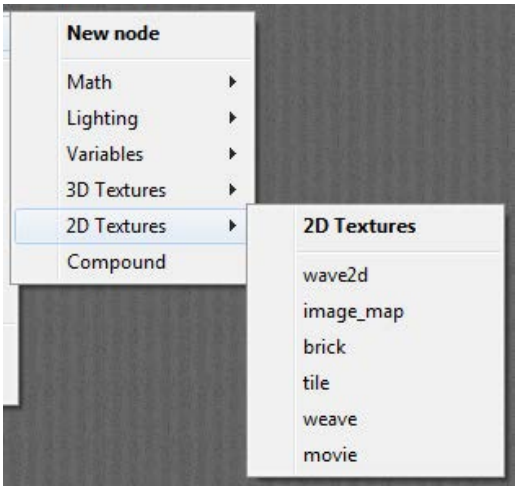
Materials are applied to objects, objects parts and more precise: to specific Material Zones within those objects and parts. 74 MATERIAL ZONES, OR: TO WHICH BODY PARTS DO THE MATERIALS APPLY TO? discusses the details.

Material Room supports the creation of quite elaborate node-trees, like a programming language into material definitions. This section will not address the art of such programming itself, but will present and discuss the building blocks alone. These can be found in the Variables (77 WHAT ARE THE VARIABLE NODES USED FOR?) and Math (81 WHAT ARE THE MATH NODES USED FOR?) groups.

The next subsection V MATERIALS FOR NON-OBJECTS presents articles on defining the properties of atmosphere, background and lights as far as these are handled through the Material Room interface. The Appendix: Poser nodes / settings and versions lists all Material Room nodes and relevant Render Settings, and their availability in the various Poser versions.

70 What are the 2D surface texture building blocks used for?

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).



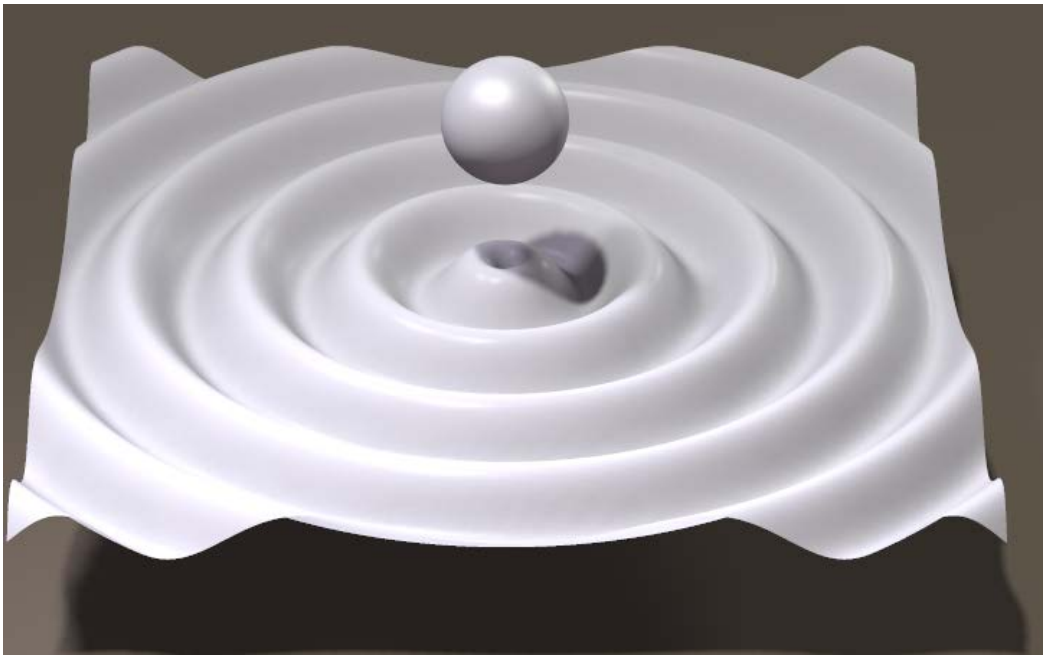
Advanced

The various nodes from the 2D Textures group all serve the same purpose: offering an image or tiling pattern which is to be (UV-) mapped onto the object surface. Image_Map and Movie do so with external static or dynamic sources, see 66 How can I assign an image to a material? and 68 How can I assign a movie to a material? respectively. The other options generate patterns on their own. Let's have a closer look at those.

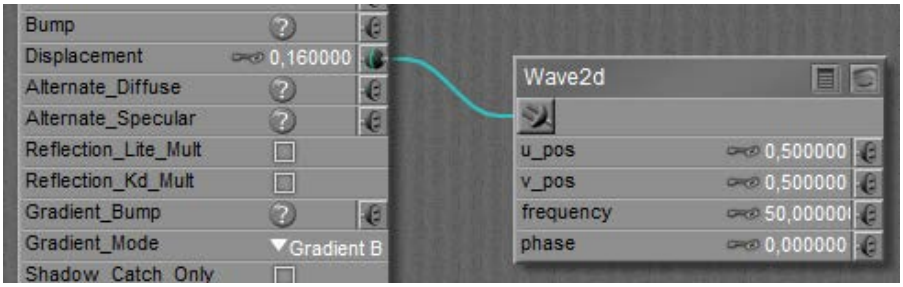
In all cases: for each portion of an object that has its own UV coordinates, both U and V vary from 0.0 to 1.0. U mainly in horizontal direction, V in vertical direction.

Wave2D

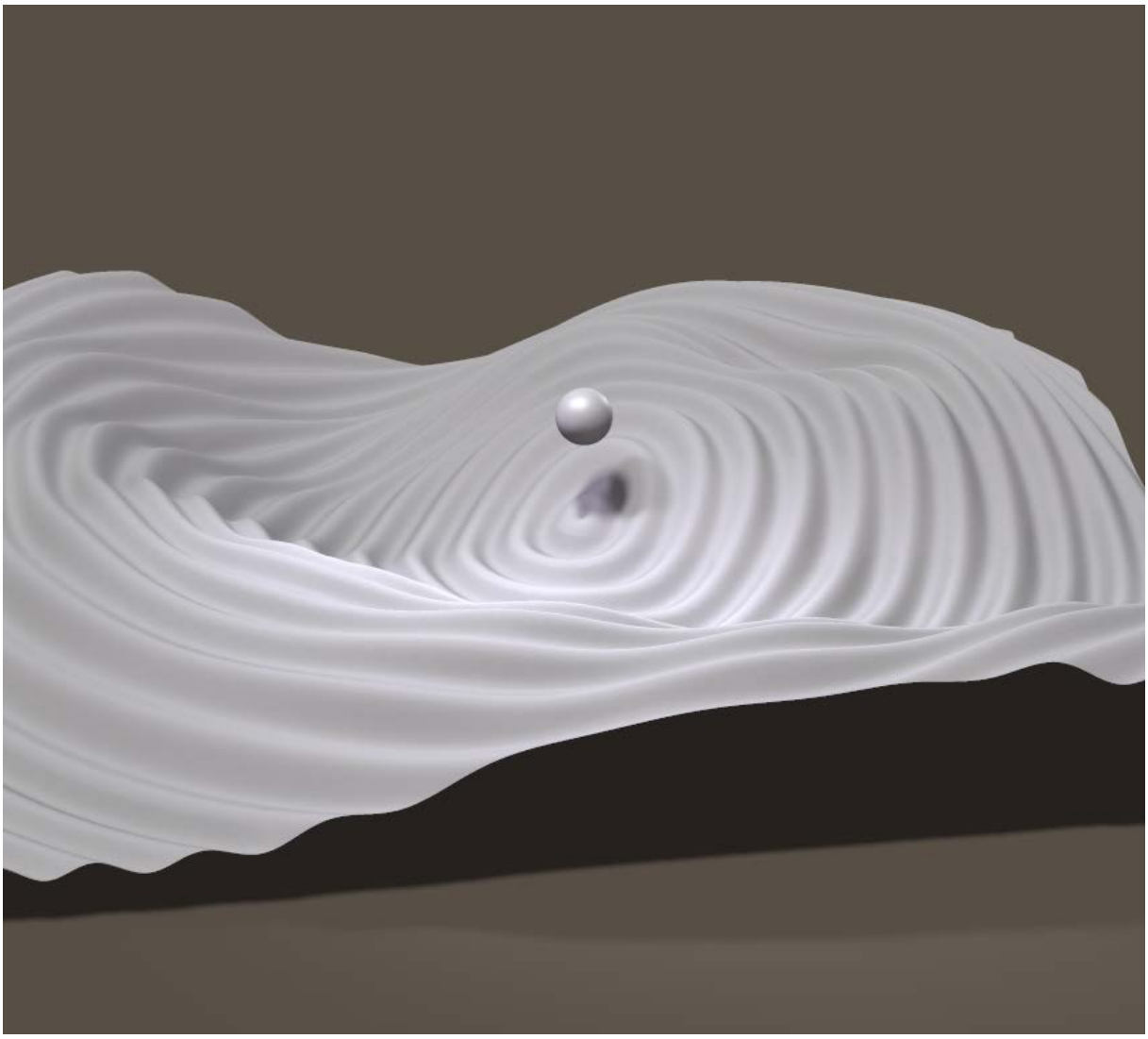
This node generates a circular sine wave pattern, which is ideal for wavy displacement maps.



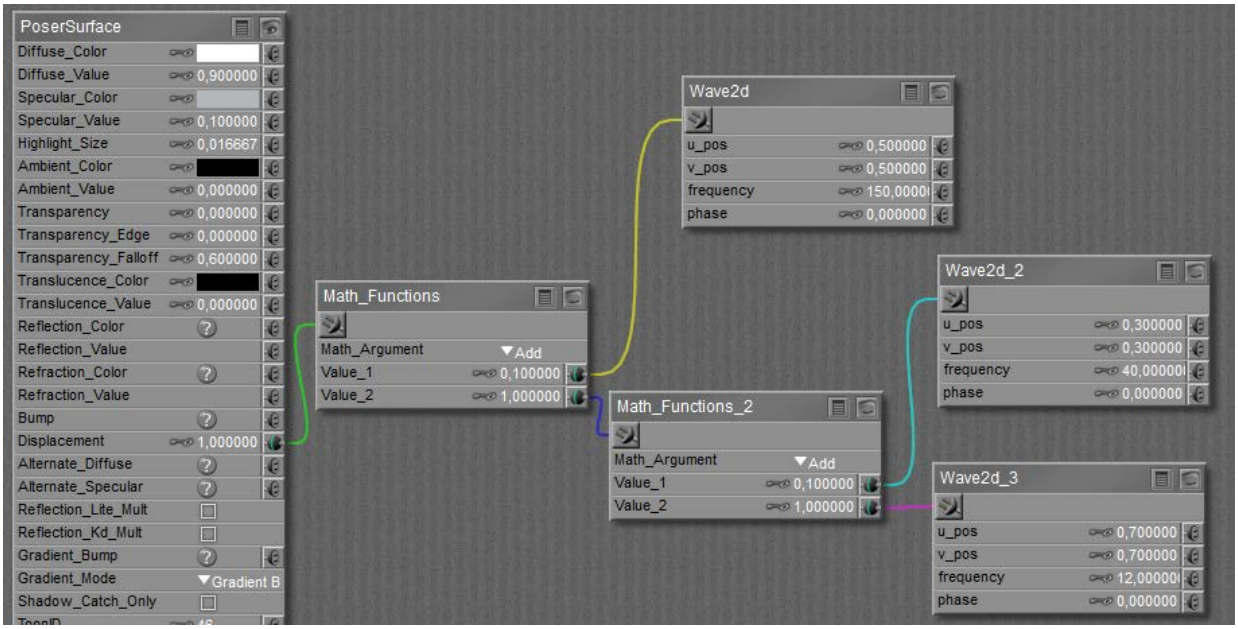
The **U_pos**, **V_pos** parameters determine the center of the pattern, **Frequency** determines the number of repetitions and by animating the **Phase**, the pattern starts rippling.



As a single pattern looks quite simplistic and cartoony, one can combine patterns with different frequencies (and origins, phases as well) to make more complex, realistic ones.

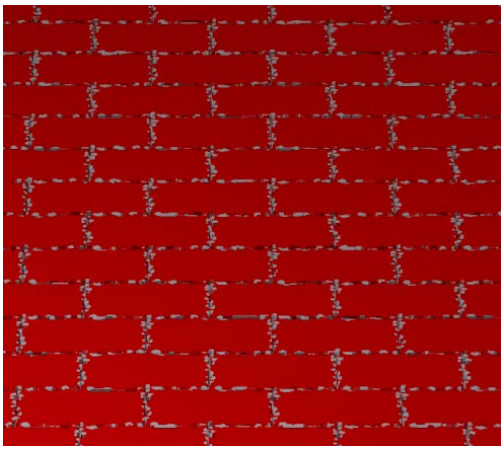


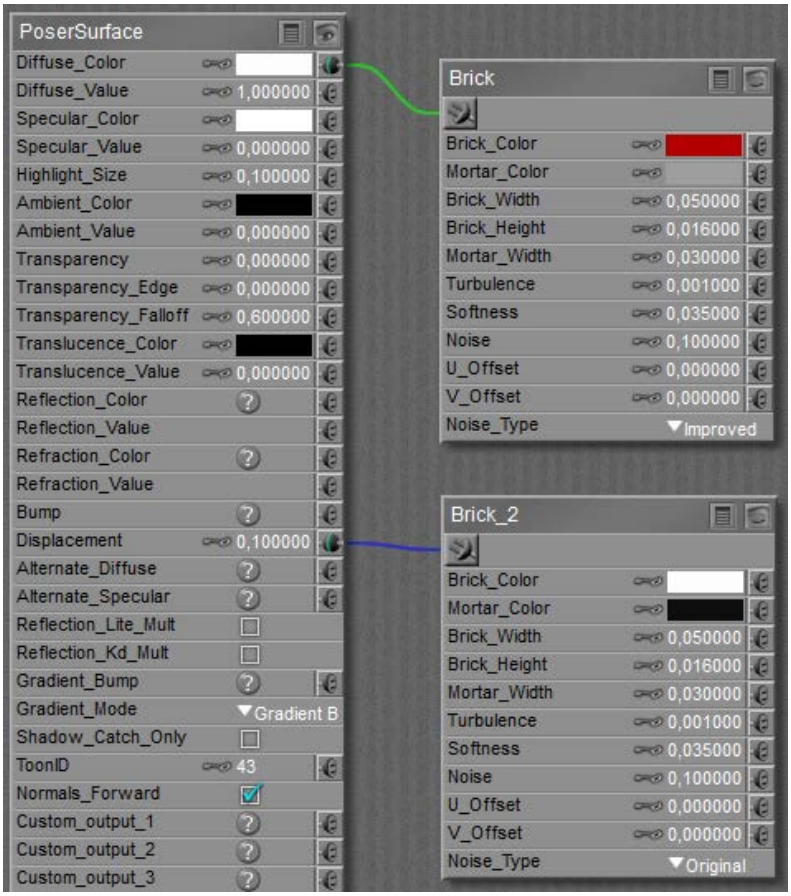
Made from



Brick

This node generates a brick & mortar texture, in the first place for use in the Diffuse slot, or in combination with its alternatives. But because brick and mortar behave different for specularity, and for bump/displacement, variants of the node are usually combined with some other components of the PoserSurface.





Brick_Width and **Height**, and **Mortar_Width** define the main aspects of the texture, while **U_offset**, **V_offset** define the placement of the texture on the object. Note that all of those are expected to vary within the 0..1 bounds.

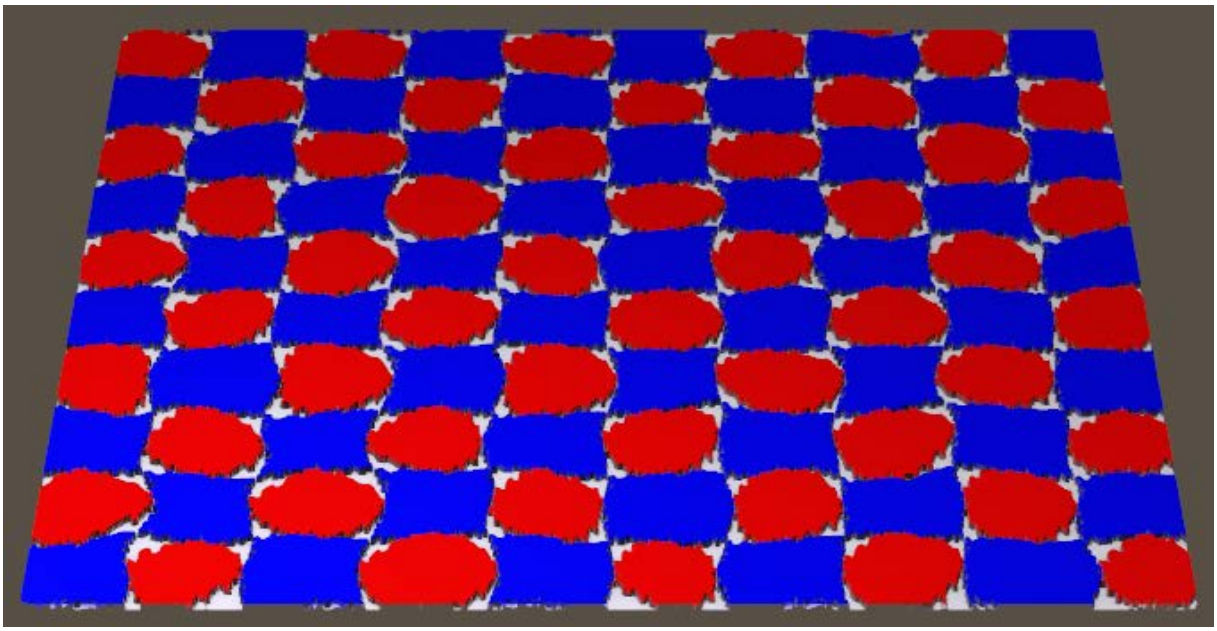
Turbulence makes the mortar-lines less straight and perfect, while **Noise** makes the mortar filling more irregular. As the manual states: Turbulence is low frequency, Noise is high frequency variation. In plain English: Turbulence affects the shape of the stones, Noise affects the structure of the mortar. I can choose between original and improved noise, the first one is meant for compatibility with older Poser versions.

Softness defines the brick-to-mortar distinction, the higher the value the more brick blends into the mortar.

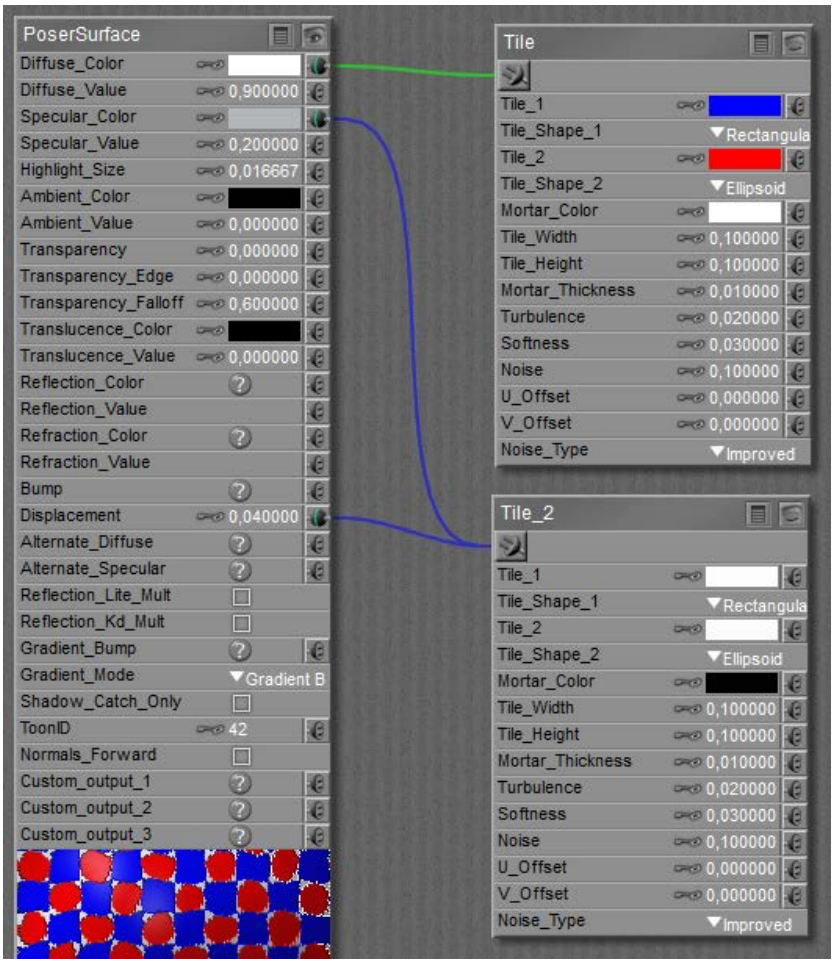
Tip: believable brick textures do need serious variation in brick color and brightness. Using additional (non-brick) textures for Diffuse_Value, or plugged into to Brick_Color slot, can help a lot in this. Especially the Clouds node (from the 3D Textures collection) might be a nice one.

Tile

Although the Brick node can be used to make shiny tiles as well, all of those will be in a similar color, and all of them will be rectangular too. The Tile node makes a variation on this, by offering alternating colors, and roundish (ellipsoid) shapes as well.



Made with

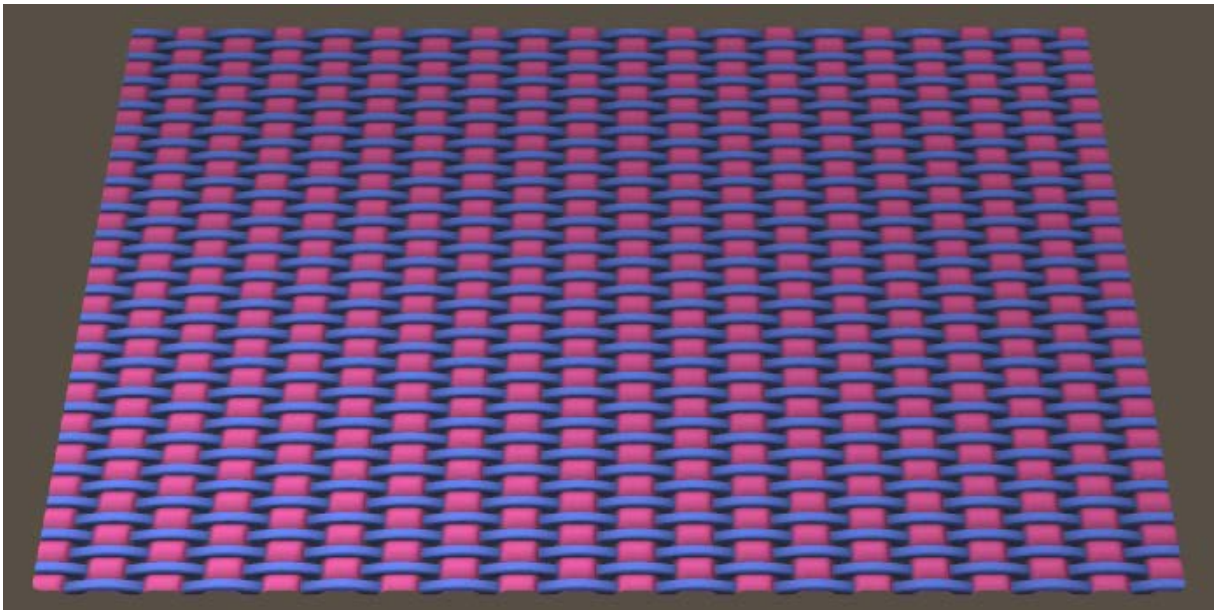


Parameters are similar to Brick: **Tile_Width** and **Height**, and **Mortar_Width** define the main aspects of the texture, while **U_offset**, **V_offset** define the placement of the texture on the object. Note that all of those are expected to vary within the 0..1 bounds.

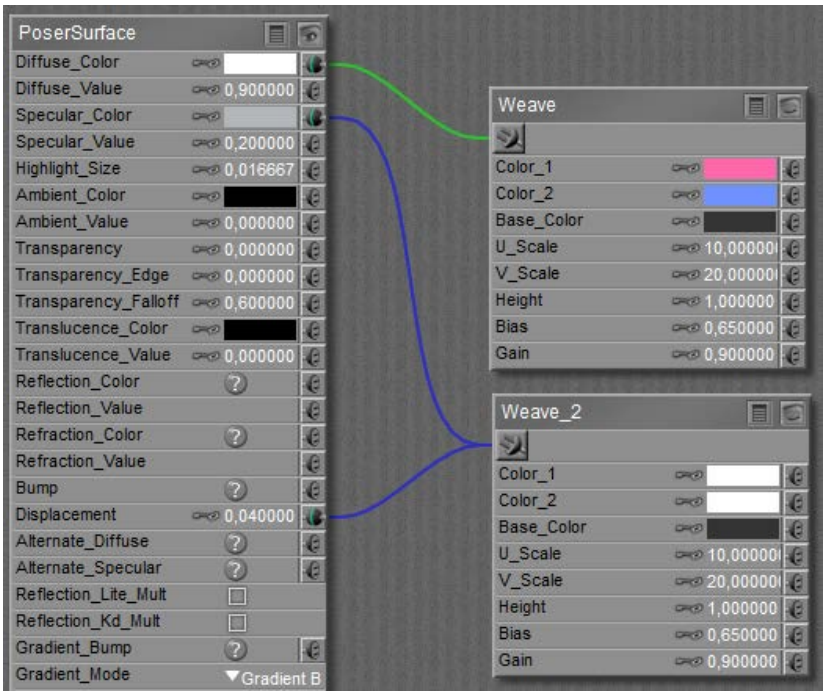
Turbulence affects the shape of the tiles, **Noise** affects the structure of the mortar. I can choose between original and improved noise, the first one is meant for compatibility with older Poser versions. **Softness** defines the brick-to-mortar distinction, the higher the value the more brick blends into the mortar.

Weave

Weaves are meant for cloth, but of course one can make nice garden fences with it too. It's just a matter of scale (and color, and material details). This:



Is made with



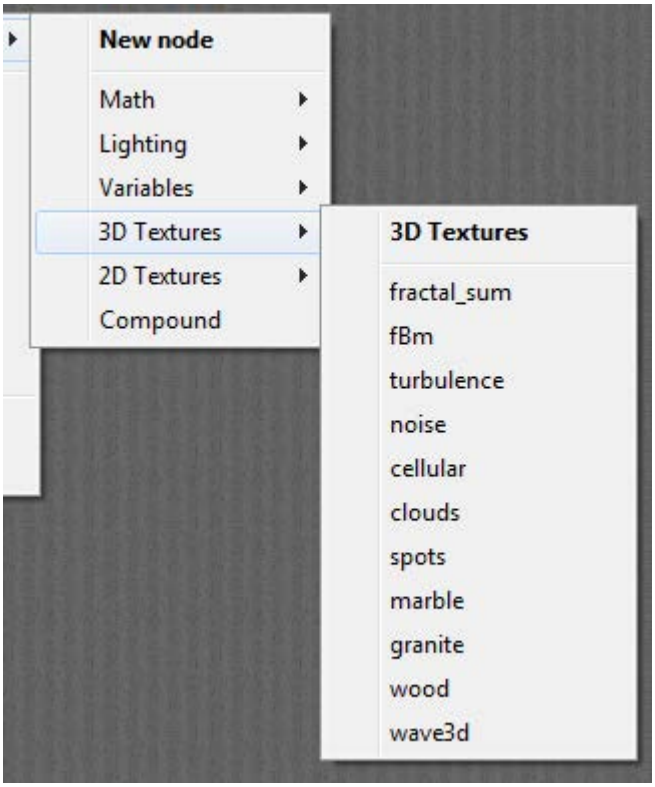
The colors speak for themselves, and the **U_Scale** and **V_Scale** determine the amount of threads on the surface (times two, as the object surface apparently has U and V from -1..1). So the plane shown above has 2x 10 (U-Scale) threads in Color-1, and 2x 20 (V-Scale) threads in Color-2 which then of course have half the thickness.

To my observation, the **Height** parameter has no effect at all. Larger **Bias** values create thicker and flatter threads, or: reduce the space between them. Larger **Gain** values sharpen the texture, higher values make more distinct threads, as for paper / wood strips while more fuzzy threads are fine for textile weaves. Bias and Gain vary from 0.0 to 1.0.
Note: at other places Bias is defined as brightness of the underlying greyscales, and Gain is defined as its contrast, both as if applied by Photoshop or alike in a pre-processing stage.

71 What are the 3D surface texture building blocks used for?

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

Advanced



3D Textures are meant to be 'mapped' against their position in space, such that multiple objects, eventually combined to one larger thing, show continuity in their looks. Think about a large city wall made from various block elements placed next to each other. The consequence of a texture driven by an objects position in space, is that changing position (e.g. in animation) make the texture change accordingly. This might either be utmost unwanted, or might be a really nice special effect. That's up to me.

Let's have a closer look. In 72 THE 3D CLOUDY TEXTURE NODES I offer a description of

- Fractal_sum
- fBm
- Clouds
- Turbulence
- Noise

The first ones are random number or fractal based "cloudy" textures which are quite suitable for dirt patterns on walls and for atmospheric clouds. The latter are more generic helper functions.

In 73 THE 3D SOLID TEXTURE NODES I offer a description of

- Cellular
- Spots
- Marble
- Granite
- Wood
- Wave4d

The first ones are meant to fill up solid objects with a texture that has a lot of randomness in it, the latter produce deterministic patterns, which some turbulence for a more natural effect.

72 The 3D Cloudy texture nodes

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

Advanced

This article presents a description of the following nodes from the 3D Textures group:

- Fractal_sum
- fBm
- Clouds
- Turbulence
- Noise

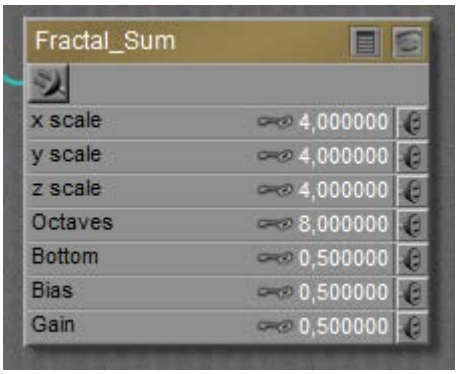
The first ones are random number or fractal based “cloudy” textures which are quite suitable for dirt patterns on walls and for atmospheric clouds. The latter or more generic helper functions. I offer descriptions of the remaining nodes from the 3D Textures group in 73 THE 3D SOLID TEXTURE NODES.

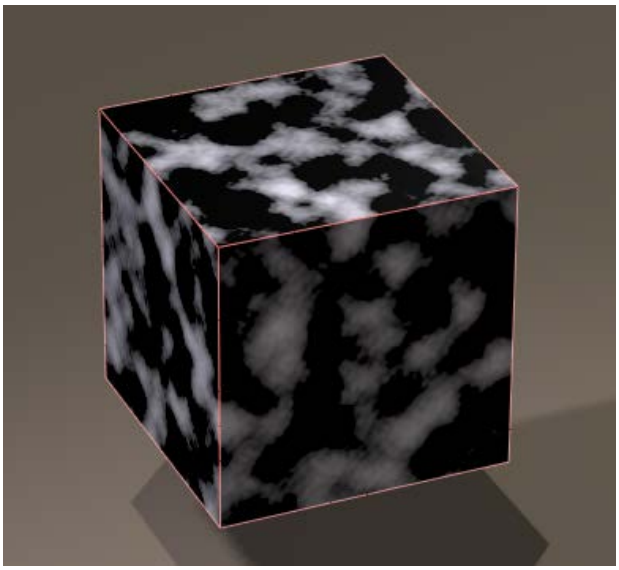
Fractal_Sum

This node returns a result between -1..+1, driven by a fractal function. Although fractal results make a strong irregular, even random impression, they differ from noise in various ways:

- The result is mathematically predetermined, repeatable
- The result in continuous in X,Y,Z

The Fractal Sum makes “mildly complex” patterns and can be considered very useful for surface patterns, like stains, dirt, rust etc.



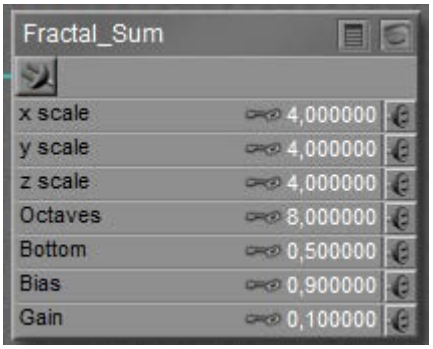
makes 

Larger **X/Y/Z Scale** values make smaller patterns, and as a fractal is an irregularity of an irregularity of … (etc.) and each step in this is “an Octave” (as in music), raising **Octaves** increases the complexity of the pattern.

Although the fractal function generates values from -1..+1 (with like a sine wave, more ‘hits’ in the extremes), these are folded into the 0..1 range. Without further measures, the 0.. -1 range is folded onto 0.. +1 and that will give a lot of high (bright) values and far less low (dark) values. The solution is to lift the result with the **Bottom** value, 0.5 by default. In that case, the result ranges from -0.5..+1.5

and the 0.. -0.5 is folded up onto 0.. +0.5 while 1.. 1.5 is folded down onto 1.. +0.5. That gives as much brights as darks in the pattern.
Simply stated: 0.0 or lower will result in a bright, bleached pattern. Raising it will darken the pattern till it's completely black, until 1.0 is entered exactly which turns the result completely white.

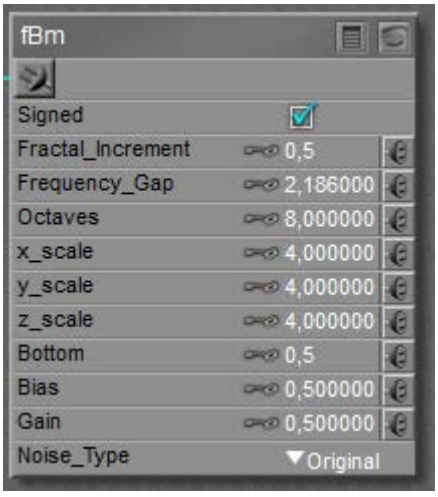
Bias and **Gain** get their (usual) incomprehensible explanation in the manual, sorry for that. Generally, when I consider the result of this node to be a greyscale image of some sort, Bias refers to its brightness while Gain refers to its contrasts, both as if applied by Photoshop or alike in a pre-processing stage.
Both default to a 0.5 value, and lowering Bias pushes the bright details into black until all brights are gone at 0.0 while increasing Bias adds more bright details into the already bright area. This does not reduce the black areas but just removes the details, and makes hard-edged splotches. Gain works about the other way around.

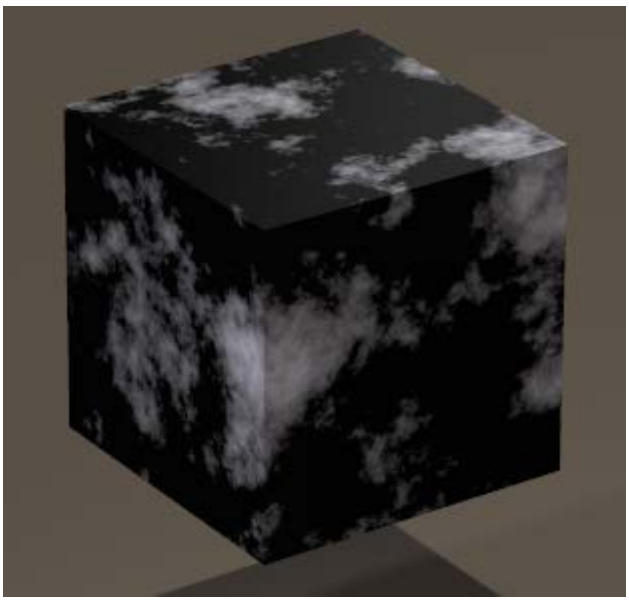


makes 

fBm

This is just another fractal function which offers parameters similar to Fractal_Sum, but results in more complex patterns as it's derived from various fractal formulas combined into one.



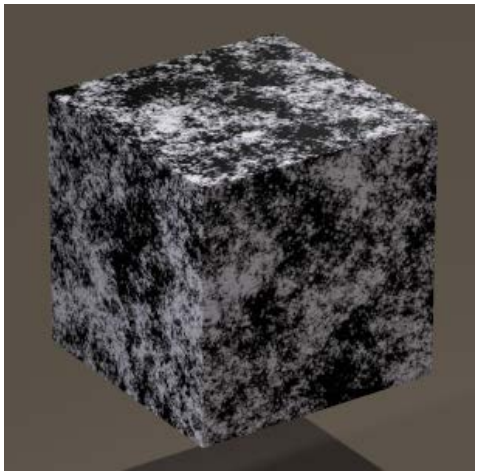
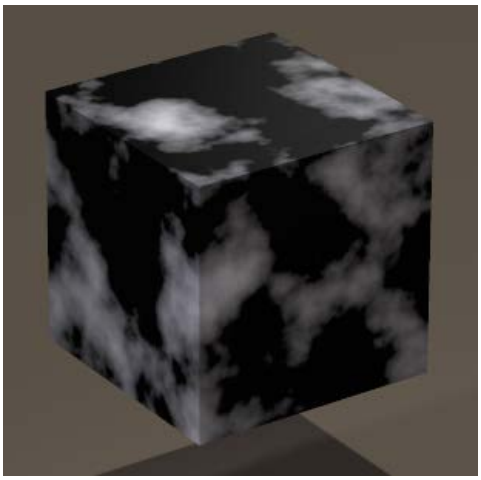
makes 

These patterns are considered especially useful for creating terrains, which requires that this node is to be plugged into the Displacement slot of PoserSurface. Though it shows too much detail for a simple stain, it might also be used for a worn, aged and rubbed one. When used for terrains, it's advised to reduce the Bottom parameter to its default 0.0 to get rid of the black (so non-displaced, flat) areas.

What are the other parameters about?

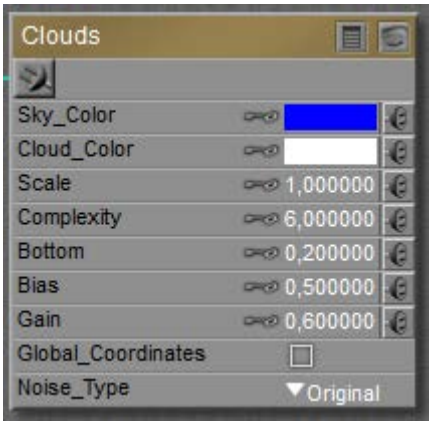
Noise_type offers the usual choice between Original (for compatibility) and Improved (recommended), and **Signed** folds the 0..-1 range of fractal results back onto the 0..+1 range. This folding somewhat disables the use of the Bottom parameter, what – as already said – can be kept at its default 0.0. **Frequency_Gap** effects the spatial distribution of the pattern, values between 2..3 are recommended and one can leave the default as is. Lower values take the details out and higher values have no effect.

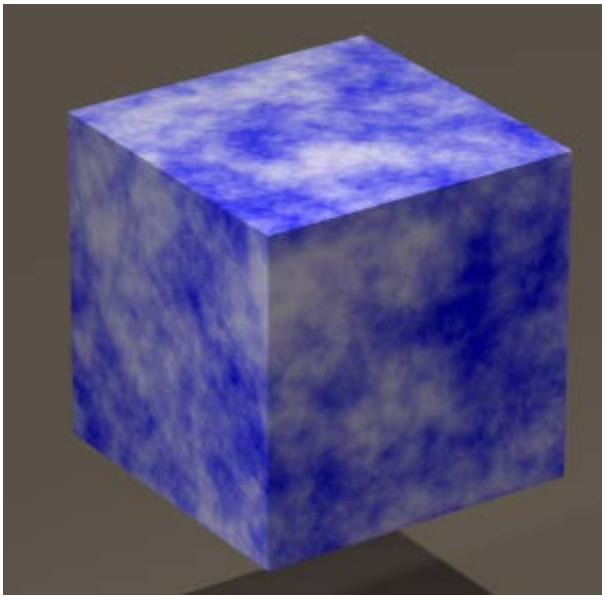
Fractal_Increment is the most interesting parameter. At 0.0 the result turns out very noisy, loose pickles everywhere. At 1.0 the result is quite consistent, and looks a lot like the simpler Fractal_Sum node. So here I can adjust the complexity and noisiness of the result, and the roughness of terrains made with it.

From 0.0  to 1.0 

Clouds

Clouds, a mixture of fractal and turbulence, serve well in backgrounds as well as in driving the fog density in Poser Atmosphere. They also serve well for fine patterns in stone.



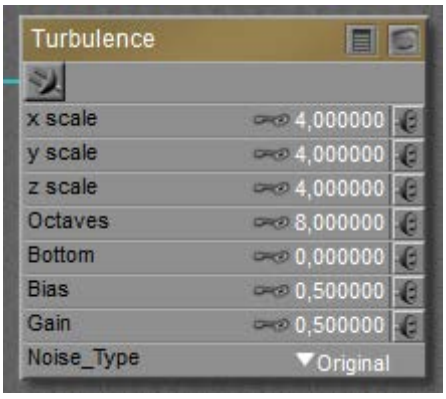
makes 

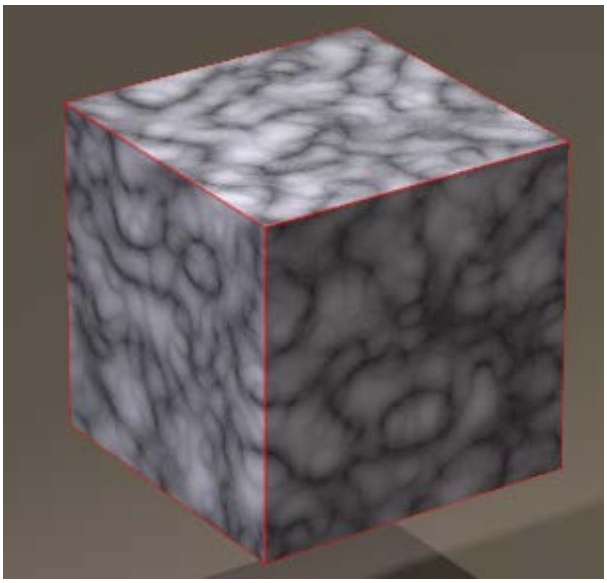
There is one **Scale** parameter for all directions, larger values make larger patterns. Higher **Complexities** make richer, more detailed and crispier patterns while **Bias** caters for the balance between dark (Sky) and bright (Cloud) areas. Higher values make less cloudy skies. **Gain** serves as a mixture: low values make a closed, less detailed cloud pattern while high values make an open sky with sharp-edges clouds.
Generally, when I consider the result of this node to be a greyscale image of some sort, Bias refers to its brightness while Gain refers to its contrasts, both as if applied by Photoshop or alike in a pre-processing stage.

As discussed at the other nodes above, **Noise_Type** lets me pick my favorite random number generator behind all the irregularities, and the **Global_Coordinates** option determines whether the pattern is relative to the object (OFF) or to the Poser space (ON).

Turbulence

Like Fractal_Sum is best for surface effects and fBm is best for terrain displacements, Turbulence is best for cloudy patterns as are required in a Poser Atmosphere (instead of a PoserSurface). The best way to look at it is to consider a space full of tiny droplets that make a fog, and some wind pushing some of them into other areas. So some areas get more filled, and adjacent areas get less.

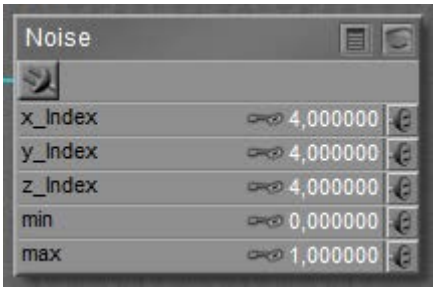


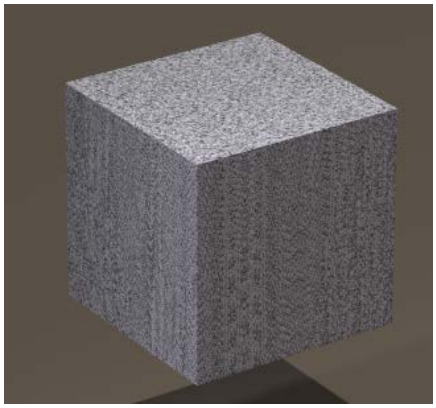
makes 

and the parameters can't be a surprise anymore, they all were discussed in Fractal_Sum as well. Turbulence does make nice marbles too, as I can see in the sample above.

Noise

Noise is not a fractal, but just generates random values (brightnesses) for each point in space. So adjacent positions do not have adjacent function results, the function is not continuous.



makes 

The **X/Y/Z_index** means the same as X/Y/Z_scale in the previously discussed nodes: larger values make smaller patterns. The result is a value between **Min** and **Max**, I can use values closer to each other to generate small color or brightness variations.

73 The 3D Solid texture nodes

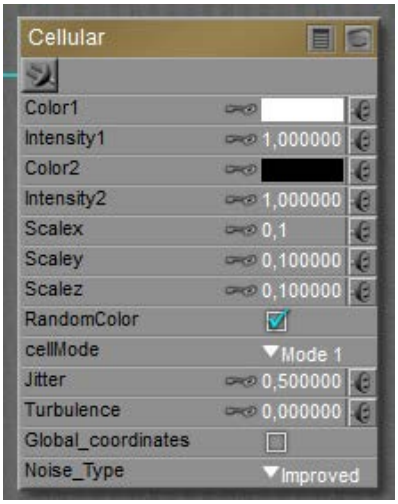
Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

Advanced

This article presents a description of the following nodes from the 3D Textures group:

- Cellular
- Spots
- Marble
- Granite
- Wood
- Wave4d

The first ones are meant to fill up solid objects with a texture that has a lot of randomness in it, the latter produce deterministic patterns, with some turbulence for a more natural effect. I offer descriptions of the remaining nodes from the 3D Textures group in 72 THE 3D CLOUDY TEXTURE NODES.

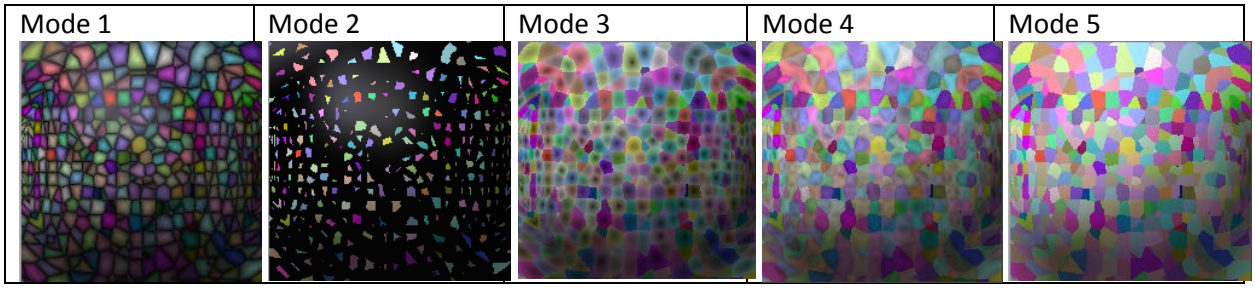


Cellular

This node generates a mosaic / stained glass-like pattern. The type of pattern is defined by the **CellMode** parameter, the **Color2 / Intensity2** defines the mortar or lead (walls of the cells) and the cells are filled with either **Color1 / Intensity1**, or by various colors when the **RandomColor** option is checked.

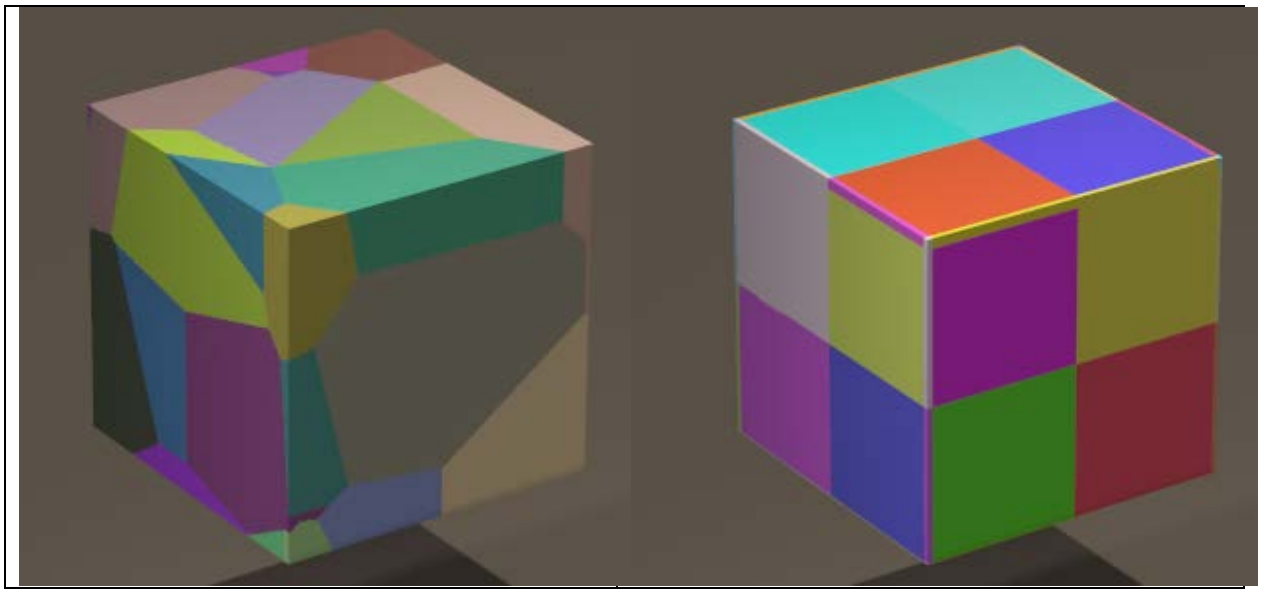
Larger **ScaleX/Y/Z** values make larger patterns (so these work opposite to the scale parameters of the previous nodes).

So what do the various (Cell)Modes do?

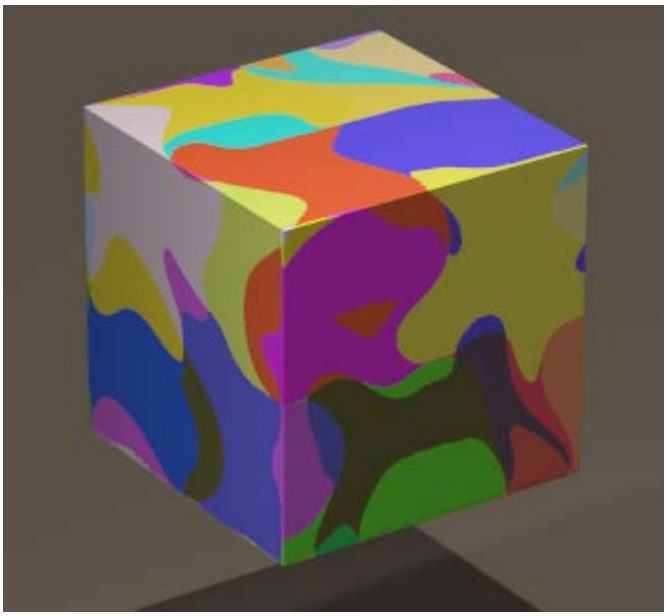


In all modes the cell patterns are similar, but the use of the “mortar” is different. Mode 1 makes nice stained-glass while Mode 2 might be nice for granites. Mode 3 and 4 use the mortar to make the structure visible in an indirect way, while Mode 5 does not apply any mortar at all. That means that this mode gives me a homogeneous fill with Color1 when the RandomColor is OFF.

Do note that the texture is a 3D one, it’s continuous in all spatial directions, mesh edges are irrelevant.



This however depends somewhat on **Jitter**, which controls cell-irregularity. The left render above was made with Jitter=1, the right one with Jitter=0.



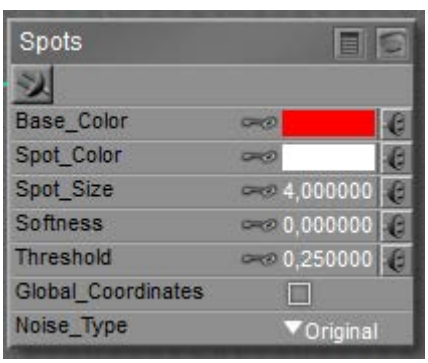
Low **Turbulence** makes straight cell edges, high values make curves ones. The right image above with Jitter=0, Turbulence=0 changes to …

when only Turbulence is raised to 1.

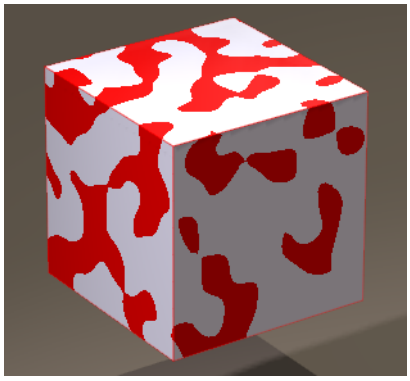
Noise_Type lets me pick my favorite random number generator behind all the irregularities, and the **Global_Coordinates** option determines whether the pattern is relative to the object (OFF) or to the Poser space (ON). The latter is relevant when multiple objects make up one 'thing' and the material has to be continuous over all of them.
Note that this also implies that the pattern changes when the object is moved around in an animation, so it's meant for inanimate objects in the scene.

Spots

Spots create the kind of pattern I get when pouring red paint (**Base_color**) into a bucked with white (**Spot_color**), though I can alter the colors of course. To me the naming of the colors seems a bit in reverse given the result below, but anyway.



makes



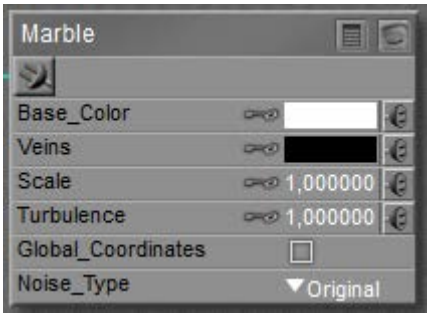
Threshold defines the Base to Spot ratio: a low value (down to 0.0) gives more white (Spots) and a high value (up to 1.0) gives more red (Base).

Softness defines the blur at the edges, at 0 the edges are well defined (image above) and higher values makes the impression that I have been stirring the mix. With 0.4 I get…

Larger **Spot_Sizes** make larger spots (no surprise there) and as discussed at the other nodes above, **Noise_Type** lets me pick my favorite random number generator behind all the irregularities, and the **Global_Coordinates** option determines whether the pattern is relative to the object (OFF) or to the Poser space (ON).

Marble

No doubt I can guess the use of this one, but… what's a proper marble pattern? As

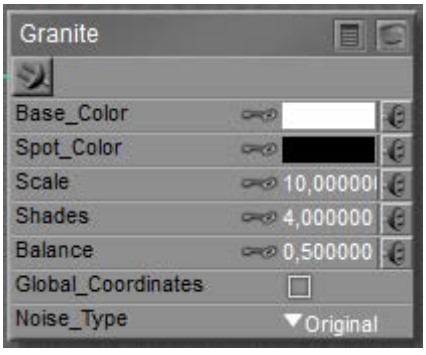


makes

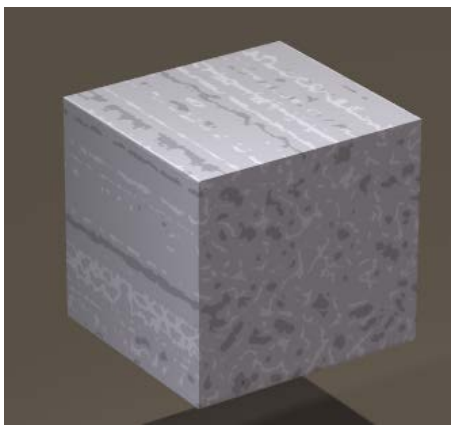
I see thick veins in one (X-) direction, and thin veins in the two other (Y, Z) directions. Larger **Scales** make larger patterns, **Turbulence** makes them wilder, and as discussed at the other nodes above, **Noise_Type** lets me pick my favorite random number generator behind all the irregularities, and the **Global_Coordinates** option determines whether the pattern is relative to the object (OFF) or to the Poser space (ON).

Granite

A no-brainer again, but what makes a granite pattern? It's sort of various kinds of gravel mixed with clay and squeezed together into one material, so



makes

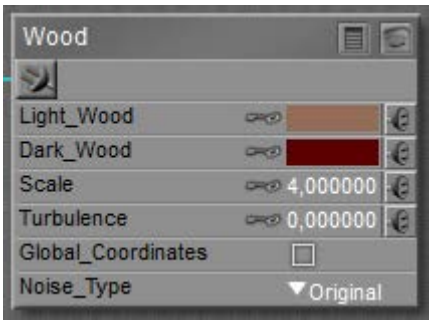


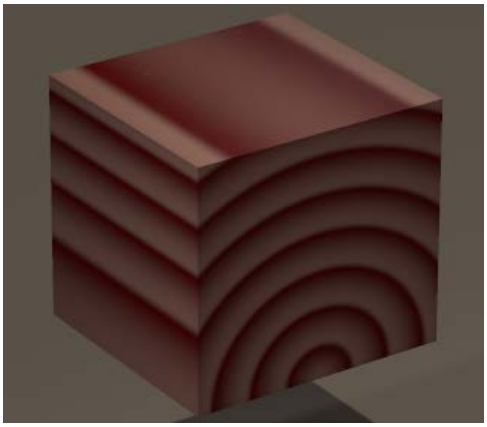
Larger **Scales** make larger patterns, note that the value is increased (from default 1) to 10 to get the granite effect visible. It's a bit like the Cellular node in Mode 2, with variants of the same (Spot) color. **Shades** determines the amount of colors in the granite, while a low **Balance** pushes the colors towards the **Spot_color**, and a high value pushes the colors towards the **Base**.

As discussed at the other nodes above, **Noise_Type** lets me pick my favorite random number generator behind all the irregularities, and the **Global_Coordinates** option determines whether the pattern is relative to the object (OFF) or to the Poser space (ON).

Wood

Another no-brainer, which indeed makes the ring-patterns I expect.

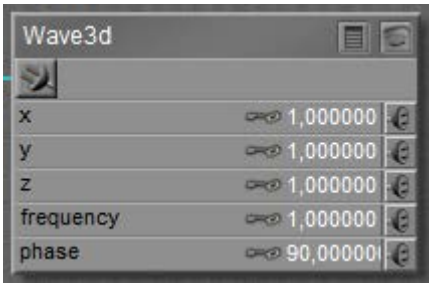


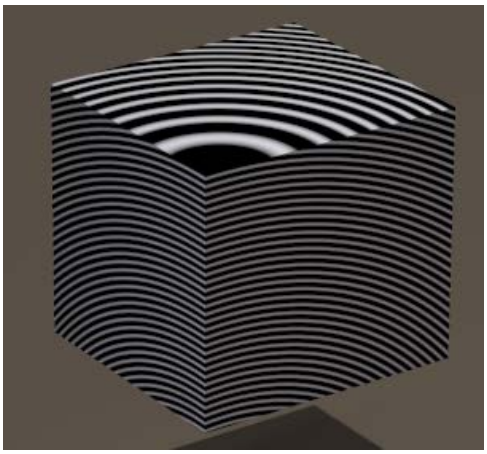
makes 

Larger **Scales** make larger patterns, and some **Turbulence** make them less regular and hence more natural. As discussed at the other nodes above, **Noise_Type** lets me pick my favorite random number generator behind all the irregularities, and the **Global_Coordinates** option determines whether the pattern is relative to the object (OFF) or to the Poser space (ON).

Wave3D

This is the 3D spatial equivalent of the 2D surface wave as discussed in 70 WHAT ARE THE 2D SURFACE TEXTURE BUILDING BLOCKS USED FOR?. In 3D, it's like a sound blast, with alternating high (white) and low (black) pressure areas.



makes 

The **X,Y,Z** parameters do NOT scale the pattern, but move the center around instead. Higher **Frequencies** make smaller patterns, and more frequent repetitions in the same area. **Phase** caters for the alternating state, and is quite valuable in animation.

The pattern is in Global Coordinates and ripples through when the object is moved around.

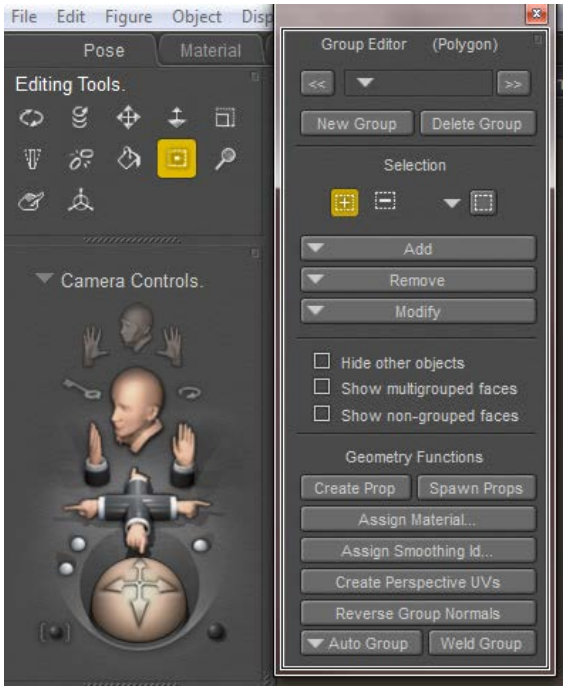
I'm sure one can find an artistic purpose for texturing with this node. Using it as a density driver in the Poser Atmosphere can make the sound-blast patterns visible in a room (vibrating smoke).

74 Material Zones, or: to which Body Parts do the materials apply to?

Although various configurations can be setup via Material Room menus, and can be managed through the Simple interface, this topic is considered Intermediate level.

Intermediate

Each object in the Poser scene is a mesh, consisting of vertices, and polygons defined as ordered groups of vertices; usually three (tri geometry), or four (quads). On one hand, an object mesh even can be formed by multiple submeshes declared to be one, even when they are not connected. On the other hand, a single continuous (sub)mesh can have its vertices and/or polygons organized into groups.



In Poser, groups are managed via the Group Editor, accessible via the Grouping Tool icon in the Editing Tools palette. The Group Editor takes a serious portion of the Figures and Props Editors chapter in the Poser Reference Manual, and reading this carefully before use is much recommended as the tool is quite powerful and can have drastic effects on the behavior of objects it's applied to.

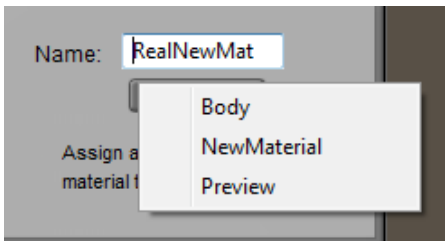
The tool can be used in various ways:

- To select polygons, and duplicate those into a prop. For instance: select portions of the head into a new Mask prop.
- To group polygons into Body Parts, this in turn can be affected by the bones of a figures skeleton. Those bones turn a prop into an animatable figure; this grouping is done from the Setup Room.
- To group polygons into Material Groups or Zones. See below.
- To group polygons into Hair Groups, this is done from the Hair Room. Those are the areas I can grow hair on, and apply the dynamics of wind and gravity to, according to my requirements. In http://www.book.artbeeweb.nl/?book=hair-room you can find some Hair Room tutorials to start with.
- To group vertices into Cloth Groups, this is done from the Cloth Room. Those are the areas I can apply the dynamics of body moves, wind and gravity to, according to my requirements. In http://www.book.artbeeweb.nl/?book=cloth-room you can find some Cloth Room tutorials to start with.

Material Zones

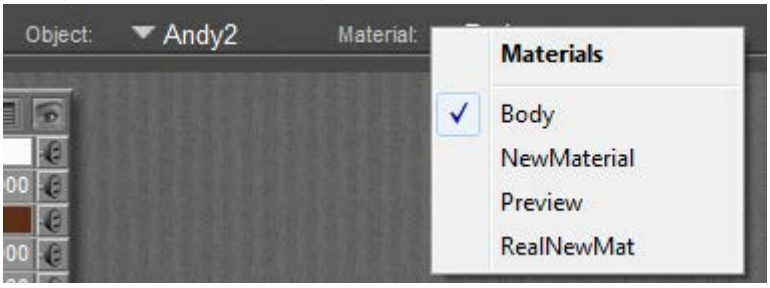
When I'm in the Group Editor, and I have selected / created a (Body Part) group of polygons, and I click the [Assign Material..] button, I can either enter a new material name, or select an existing one from the list.

or

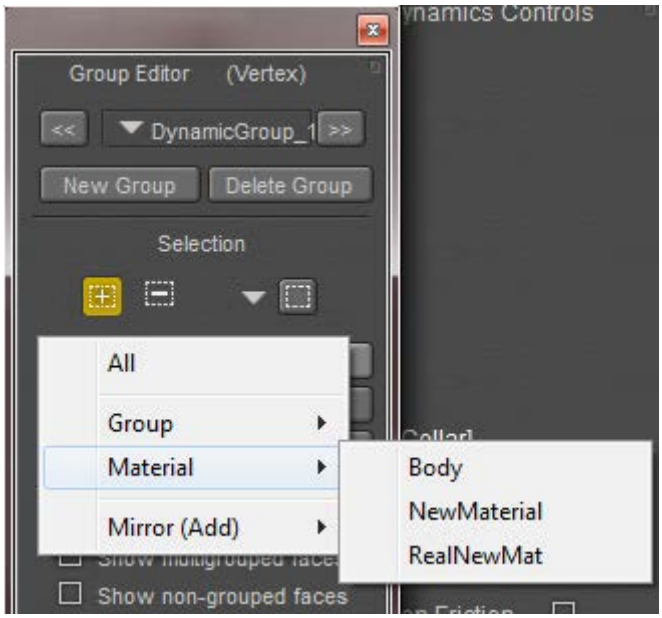


Note: in Cloth Room, the Group Editor works on vertices instead of polygons. Therefore, new material groups cannot be defined from there. The button just will not respond.

Then, from Material Room, I can define a material definition for that (Materials) zone:



But also, those Material Zones are available for defining and refining selections in groupings elsewhere (via the [Add] or [Remove] buttons). So, all vertices or polys from a Material Zone can be assigned specific dynamic properties in Cloth Room. This way, I can make portions of a dress not only look like leather or lace, but also behave as such.



And via Hair Room, I can make long and short fur or hair, both having different dynamic behavior and different colors (via Material Room) as well, by using the same groupings.

75 Shader tree construction: Math, Variable and Compound nodes

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

Advanced

Like I can define my own surface definitions by plugging node driven effects into PoserSurface (and other root node) inputs, so I can define my own effects by combining nodes into constructions. Functions calling functions calling … and so on.

Of course the effect nodes themselves, as discussed in the various articles, are relevant bricks in such constructions. But to glue them together and to help me with the construction, the nodes from the Math group (see 81 WHAT ARE THE MATH NODES USED FOR?) make the mortar between the bricks. And the nodes from the Variables group (see 77 WHAT ARE THE VARIABLE NODES USED FOR?) get the information from the scene, like the time and place of the spot on the surface for which PoserSurface is defining the material for.
As node constructions can be huge, the Compound node (see 76 WHAT IS THE COMPOUND NODE USED FOR?) was introduced (Poser 10/Pro 2014) to keep clear boards in the Material Room.

76 What is the Compound node used for?

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

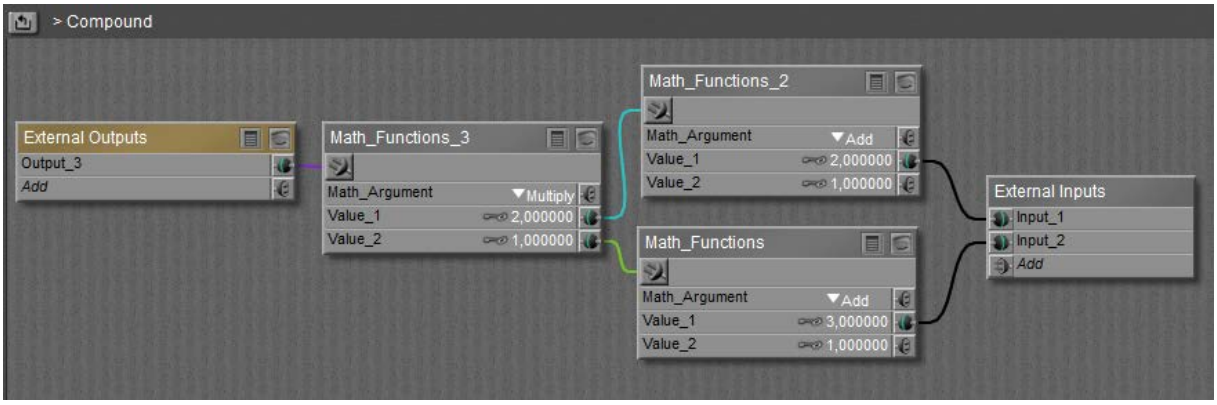
Advanced

Connecting node outputs (function call results) to node inputs (function call variables) make a very flexible way to build material definitions, like a programming language. The disadvantage is that the resulting node-trees might become quite large, and incomprehensible.

The Compound node is offered to resolve this, as a way of packaging.

Let's take an example. In the definition area of Material Room, add a Compound node. It's empty, and looks like:

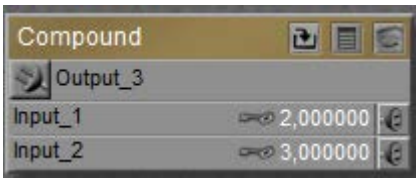
Open the node (the  button), and I can build a complete structure between the final node containing all External Outputs (left in the scheme below) and External Inputs (right in the scheme).



This example structure calculates

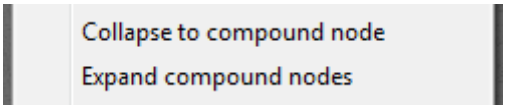
Output_3 = [2 * (2 * Input_1 +1)] * [1 * (3 * Input_2 +1)]

When done building the node can be collapsed (the  button) and will just show the outputs and inputs defined in the Material Room definition area:



Yes, you read it correctly; the compound node can offer multiple output slots.

On top of all this, the context (right-mouse-button) menu offers two additional options:

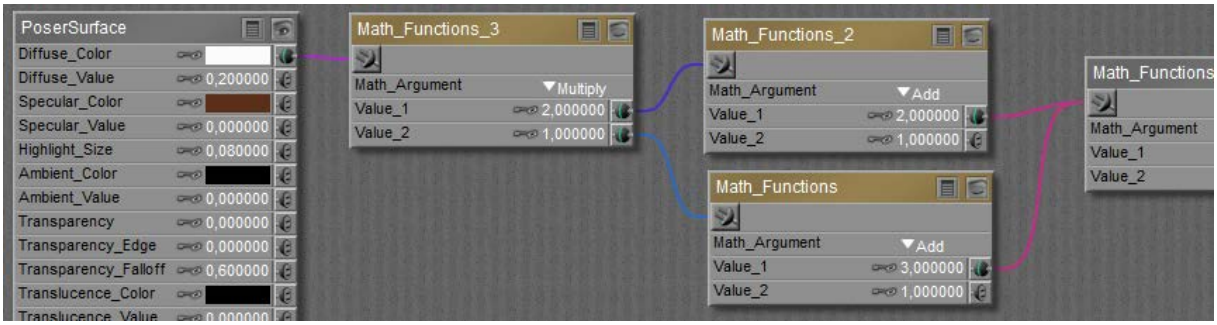


The first, **Collapse**, is a fast way to create a new Compound node. Just select all the relevant nodes I want to be moved into a new Compound one, and click this option. This will give me a Compound node with those nodes, plus an External Outputs and an External Inputs one. The only thing I have to

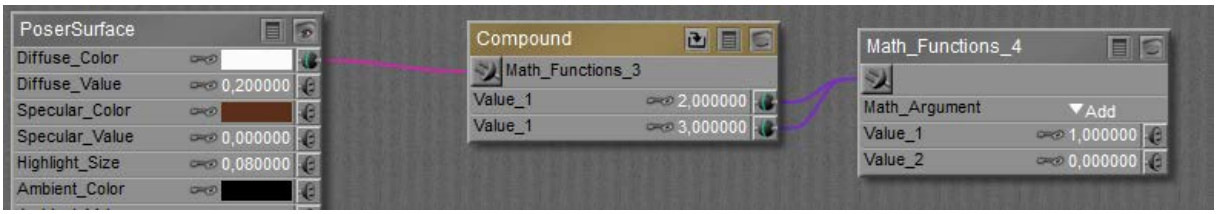
do now is to create – or check – the connections between these External nodes, and the structure I've moved in.

The second, **Expand**, does the opposite. It takes the selected Compound node, moves the embedded node structure to the Material Room area, and eliminated both External nodes.
In both cases, existing connections between nodes outside and inside the Compound nodes are maintained, and (de-)assigned to the External nodes as appropriate.

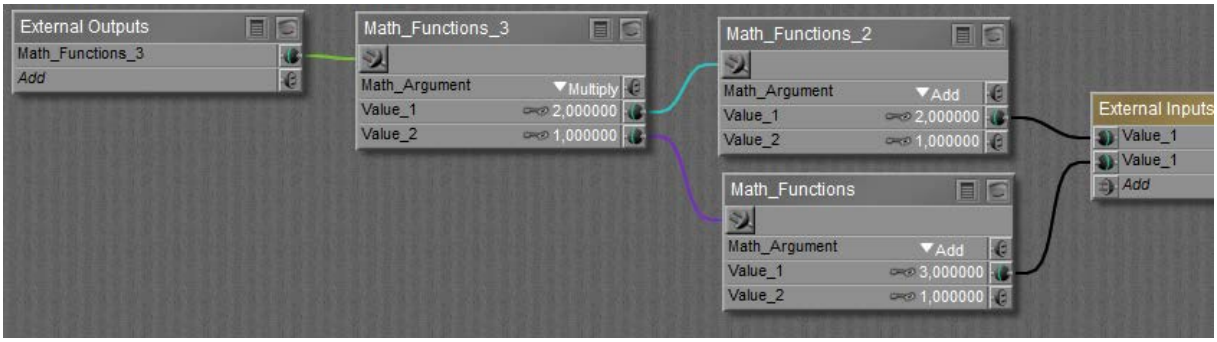
So



Becomes



And



And reverse.

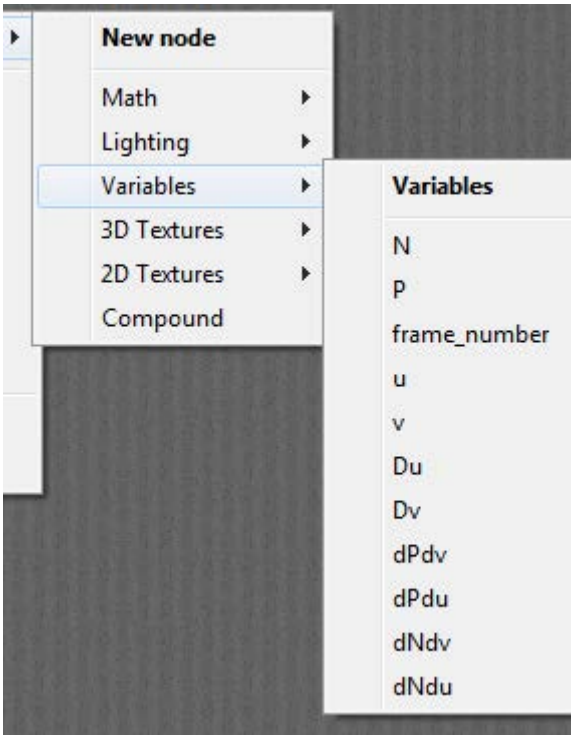
77 What are the Variable nodes used for?

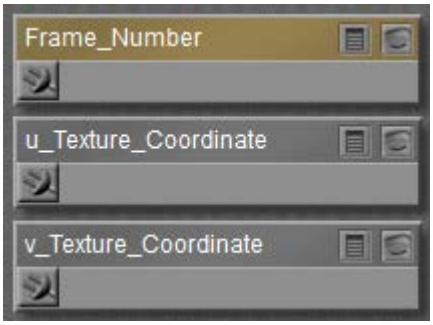
Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

Advanced

The nodes from the Variables group tell me about the position, in space and time, of the spot that the PoserSurface definition is constructed for. There are two kinds of results from these nodes:

- Single values. These are discussed in this article.
- Three value 'vectors'. Like a color is defined by (Red, Green, Blue), so is a point in space defined by (X,Y,Z) and so on. These are discussed in 78 WHAT ARE THE POINT P AND NORMAL N NODES USED FOR?, in 79 WHAT ARE THE STRETCHING (DPDU, DPDV) NODES USED FOR? and in 80 WHAT ARE THE CURVATURE (DNDU, DNDV) NODES USED FOR?

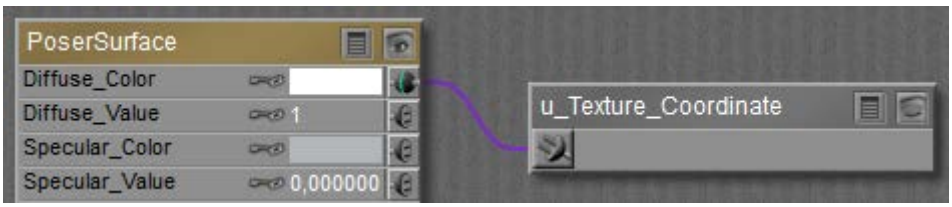




The easiest to comprehend are U, V and Frame_Number. These nodes don't take any inputs but just report the situation in the scene: at what place on the object surface(U,V, both 0..1) at what time frame the spot is located for which the PoserSurface definition has to report on to the renderer.

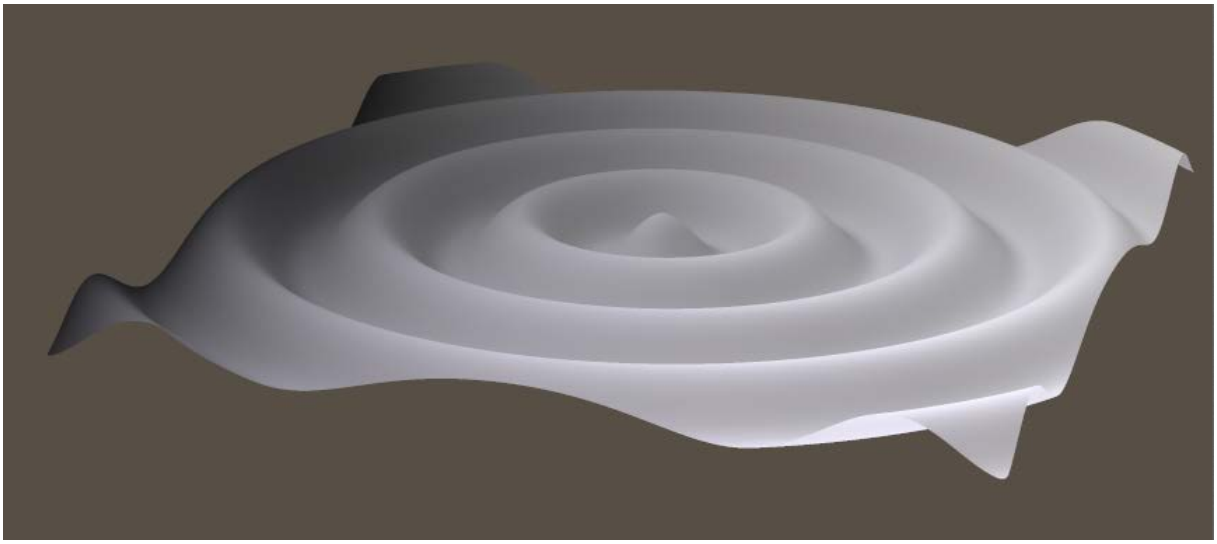
Du and Dv are sold in the manual as "defining a rate of change" but actually both produce a solid 0.0 whatever the shape and mapping of the object. Sorry for that.

To visualize U and V, just plug the nodes tight into the Diffuse channel, like

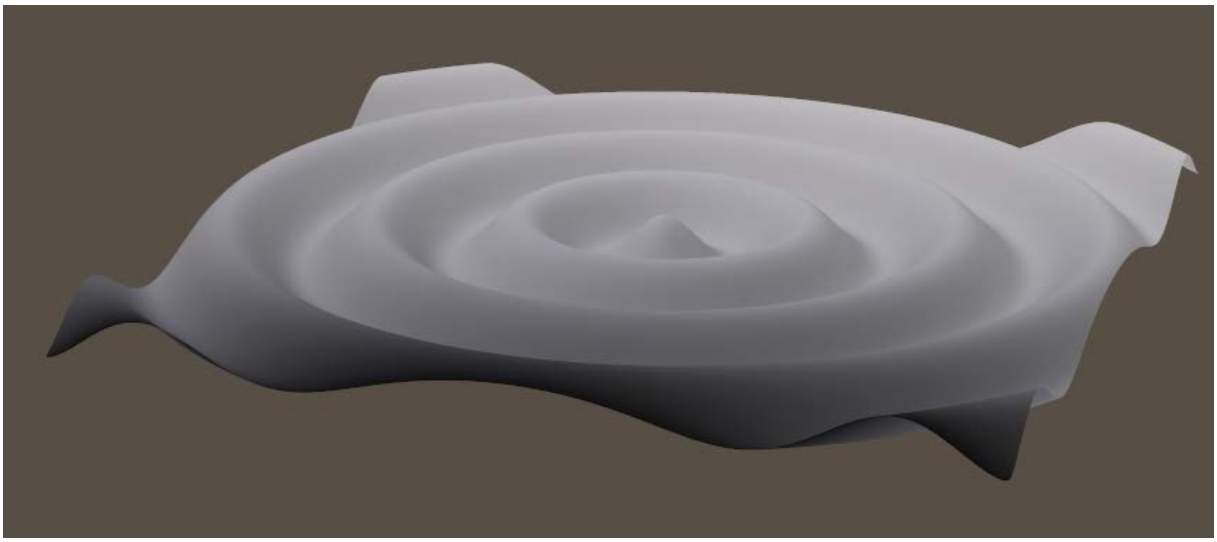


And the value from the node will drive the brightness of the color.

U from 0 (black) to 1 (white), from left to right:



V from 0 (black) to 1 (white), from front to back:



78 What are the point P and normal N nodes used for?

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

Advanced

The nodes from the Variables group tell me about the position, in space and time, of the spot that the PoserSurface definition is constructed for. There are two kinds of results from these nodes:

- Single values. These are discussed 77 WHAT ARE THE VARIABLE NODES USED FOR?.
- Three value 'vectors'. Like a color is defined by (Red, Green, Blue), so is a point in space defined by (X,Y,Z) and so on. These are discussed in this article, in 79 WHAT ARE THE STRETCHING (DPDU, DPDV) NODES USED FOR?) and in 80 WHAT ARE THE CURVATURE (DNDU, DNDV) NODES USED FOR?

Slightly more intriguing than just single values U and V are normal N and point P:

- P is a vector representing the position of that spot itself in space, and
- N is a vector, representing the surface normal at the spot at hand.

So P= (0,1,0) for the spot on the object surface one unit above the scene center, and N= (0,1,0) for pointing upward from that spot.

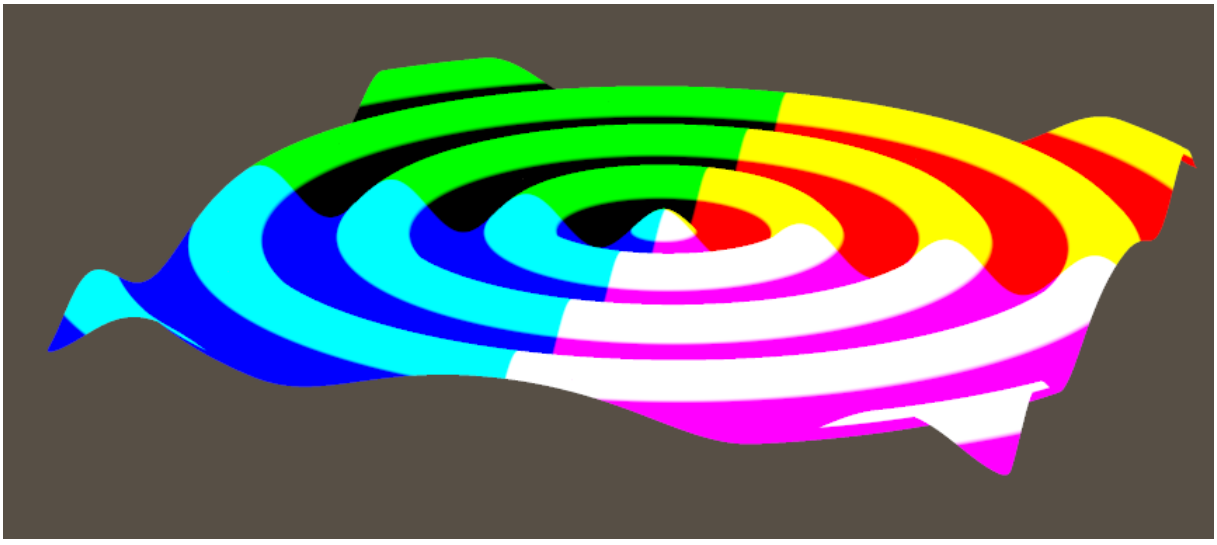
Actually, the first component of the vector is the first component of place P or normal N respectively, multiplied by the value presented in the node, and multiplied by the inputted value (or image brightness) plugged into that slot as well:

$$Result_X (xyz) = N_X (xyz) * NodeValue_X * ImageBrightness (xyz)$$

Visualization

To visualize these for a probably better understanding, the respective node is plugged into Diffuse Color. The (X,Y,Z) vector will then be interpreted as (Red, Green, Blue) as far as the values are in the 0..1 range. Values outside that range will be clipped, like negatives are interpreted as 0.

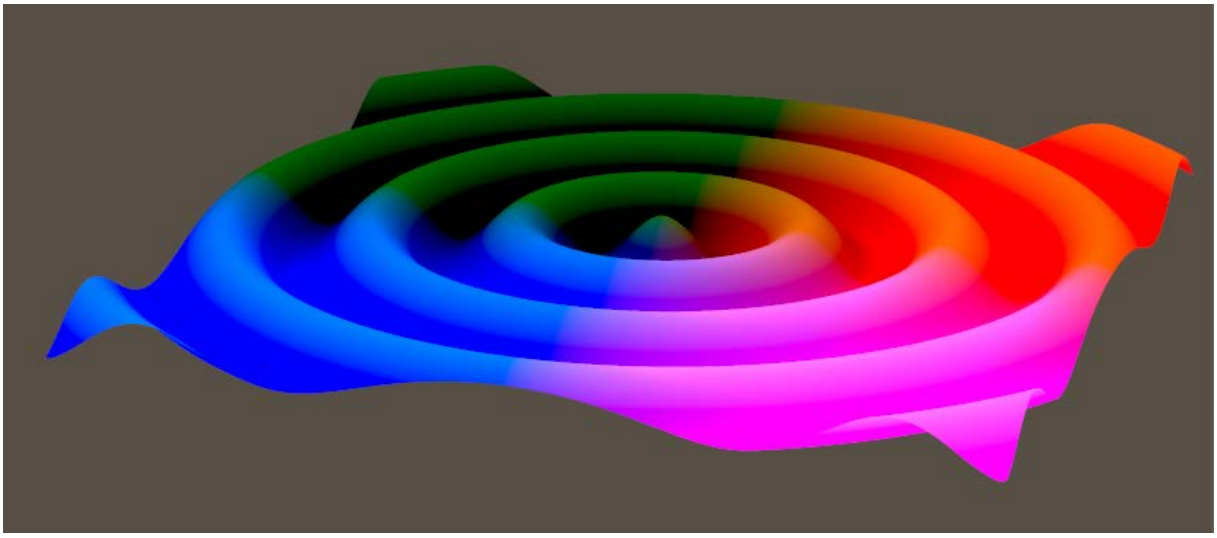
The P node, translated to colors, gives:



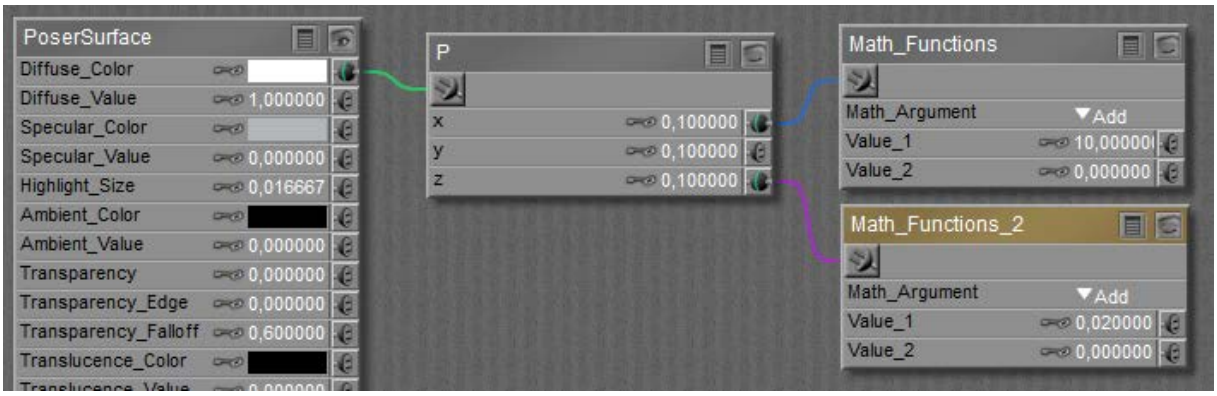
In the right half of the object, X is positive so Red is “in” but in the left half, Red is missing.
In the upper half of the object, Y is positive so Green is “in” but in the lower half, Green is missing.
In the front half of the object, Z is positive so Blue is “in” but in the back half, Blue is missing.

So the right upper back part will show full Red and Green but no Blue, that is: full Yellow. And so on.
The colors are hard, as the vector component exceed 1.0 quickly.

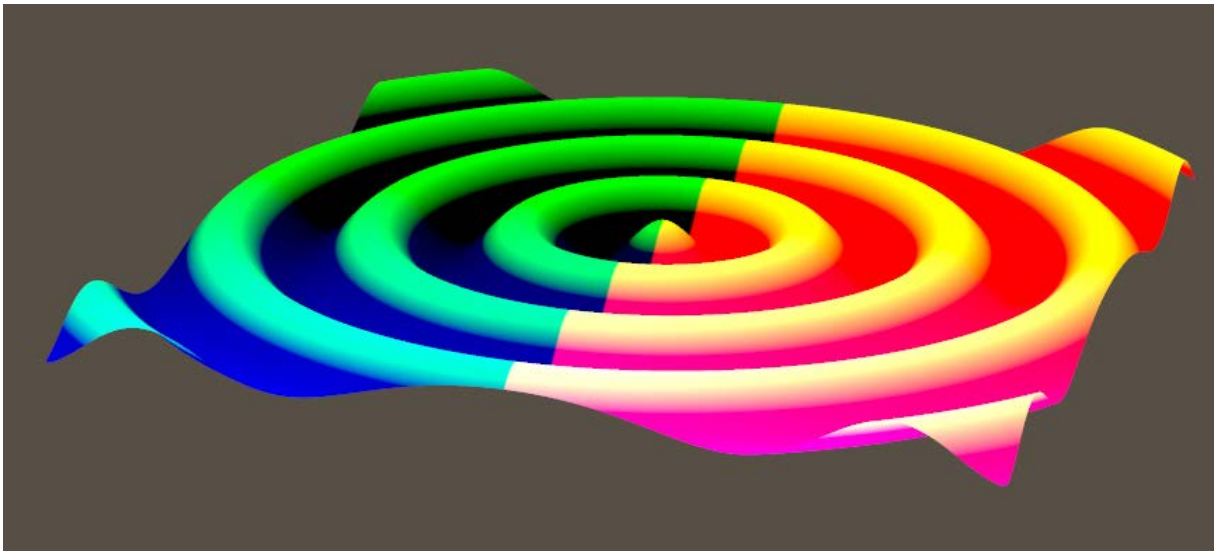
To visualize things even more, this is what I get when the P vector is shrunk: softer colors because of smaller RGB values, because the values in the node were set to 0.01 (instead of 1.0):



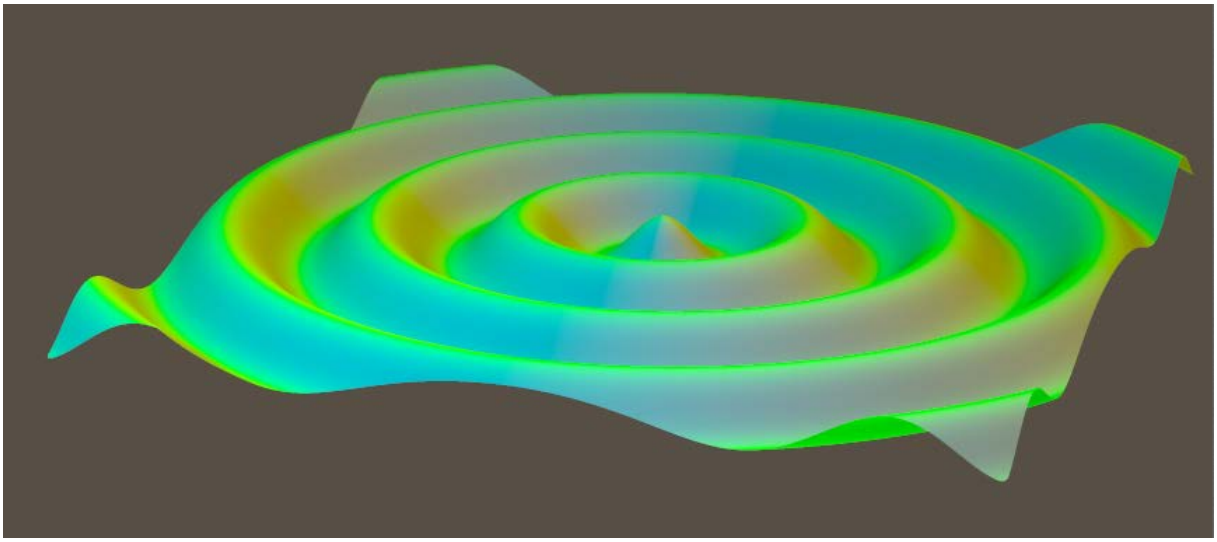
And when the first component is made large and the third one small:



Then I’ll get hard color edges in X (left right) direction, and soft gradients in Z (back front) direction:



The N node can be visualized as:



Normals point – perpendicularly – away from the surface. Pointing towards the upper (pos Y) left (neg X) front (pos Z) disables the Red (negative X is clipped to 0) and hence leaves Green and Blue making Cyan.
And so on.

79 What are the stretching (dPdu, dPdv) nodes used for?

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

Advanced

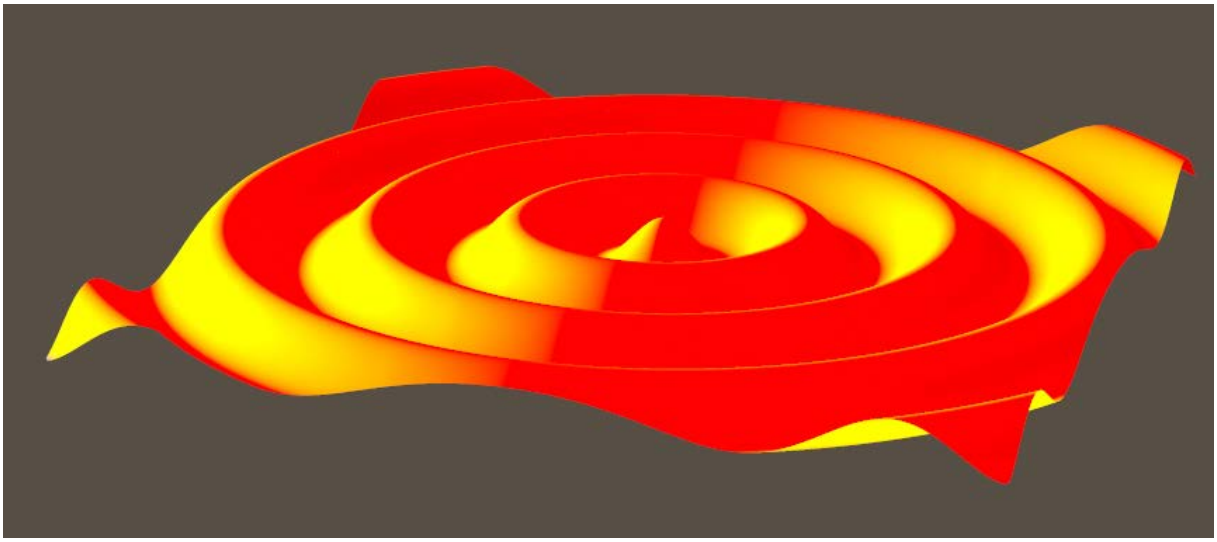
The nodes from the Variables group tell me about the position, in space and time, of the spot that the PoserSurface definition is constructed for. There are two kinds of results from these nodes:

- Single values. These are discussed in 77 WHAT ARE THE VARIABLE NODES USED FOR?.
- Three value 'vectors'. Like a color is defined by (Red, Green, Blue), so is a point in space defined by (X,Y,Z) and so on. These are discussed in 78 WHAT ARE THE POINT P AND NORMAL N NODES USED FOR?, in this article (stretching dPdu, dPdv) and in 80 WHAT ARE THE CURVATURE (DNDU, DNDV) NODES USED FOR?

The four Variables mentioned last tell us about the shape of the object, and the mapping that comes with it. When a large step in Poser space from one point on the surface to another comes with just a tiny step in U or V, then the mapping stretches the texture over a large area. In the inverse case, the texture is squeezed into a tiny spot. That is what dPdu and dPdv are about. A large value for dPdu means: stretched. And by the way: it's a vector, so we can have stretching in X and squeezing in Z direction, for instance.

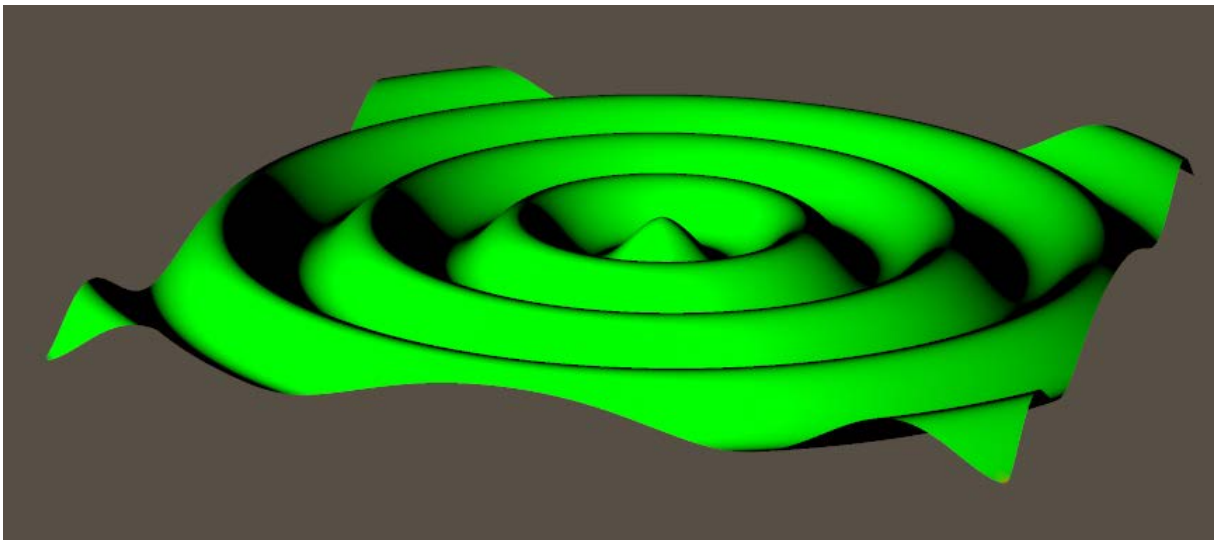
Visualization

To visualize them for a probably better understanding, the following renders show them translated to colors. First dPdu:



Let's start in the middle and move gracefully to the right, the positive U direction. So, adjacent point will not differ in Z direction, and hence there will be no Blue (3rd vector component) in the result. And they will positively differ in X, which introduces Red (1st component).
At the upgoing side of the wave, points will differ in Y in a positive way (the path is up). This makes Green (2nd component), and with the already available Red this will make Yellow, as we see. At the downgoing side of the wave, points will differ in Y as well, but in a negative way. Negative values are clipped to 0, and only the Red will remain.
Note that when I walk to the left, I make negative steps in U, so the upgoing path will be red, and the downgoing part – with a negative Y as well – will turn yellow as negative * negative makes positive.

Now let's look at dPdv:



Let's start in the middle and gracefully move to the back (that's increasing ! in V). As the X value won't change, there will be no Red (1st component). The Z value decreases gradually as Z is positive in forward direction, and we're walking backward. This negative value is clipped out when turned into a color, which is why we won't see any Blue (3rd component) either. While walking to the back (positive V steps) the upgoing portions of the path have a positive Y difference, and so the result will turn Green (2nd component) there. The downgoing portions have a negative Y difference, which will be clipped out. Then in that case no color-values are left, and the result is black.

Brief conclusion: dPdu and dPdv are vectors. Large negative or large positive values indicate surface stretching in either direction, small values indicate squeezing of the image onto the object. Large and small in this are relative to the size of the object. From left to right, U changes from -1 or 0 to 1, while P changes with the object-size in X direction. So, dP/du equals (half of) objectsize_X for a non-stretched non-squeezed mapping.

80 What are the curvature (dNdu, dNdv) nodes used for?

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

Advanced

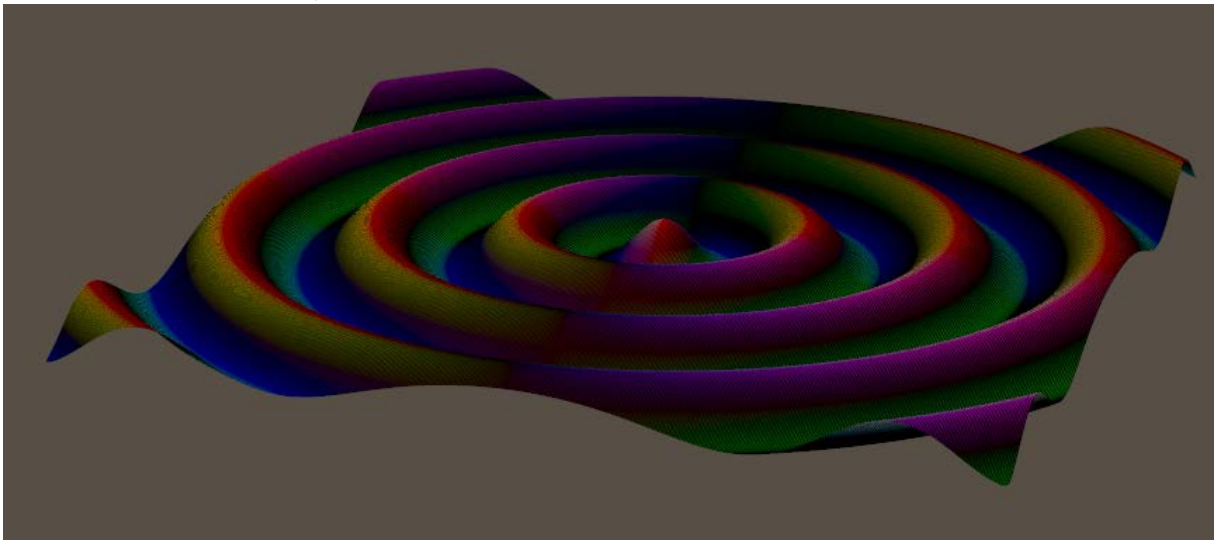
The nodes from the Variables group tell me about the position, in space and time, of the spot that the PoserSurface definition is constructed for. There are two kinds of results from these nodes:

- Single values. These are discussed 77 WHAT ARE THE VARIABLE NODES USED FOR?.
- Three value 'vectors'. Like a color is defined by (Red, Green, Blue), so is a point in space defined by (X,Y,Z) and so on. These are discussed in 78 WHAT ARE THE POINT P AND NORMAL N NODES USED FOR?, in 79 WHAT ARE THE STRETCHING (DPDU, DPDV) NODES USED FOR? and in this article (curvature dNdu, dNdv)

The four Variables mentioned last tell us about the shape of the object, and the mapping that comes with it. The last two of these, dNdu and dNdv, look at variations in the surface Normal. When that varies a lot in a small step in U or V direction, it means that the surface is much curved. When it hardly varies, and all components of such a vector are close to 0, the surface is pretty smooth and flat.
Yes, curvature is a vector too. I might have a lot in one direction, and hardly some at another. Do note that curvature is not slope. Slope is indicated by the normal at the point of interest, curvature is the rate of change in that slope.

Visualization

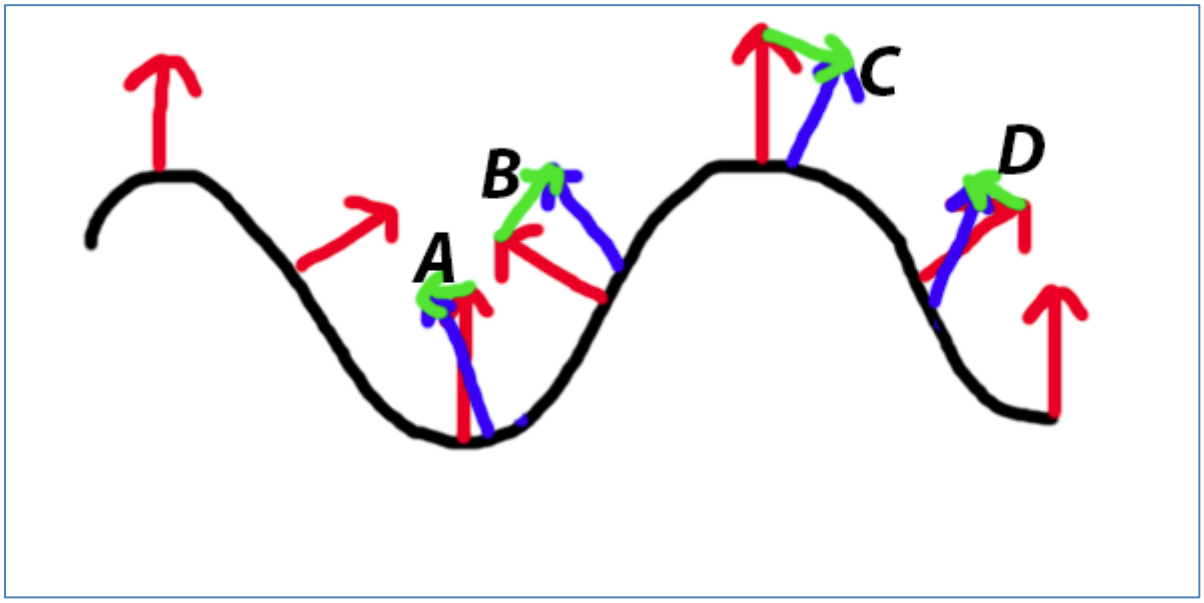
These renders show dNdu, translated to colors:



When it's much curved, the colors are intense. Just before and after the tops, and just before and after the valleys. When it's straight, either skewed or flat, the colors are mixed and dimmed. At the top and in the valley, but also halfway at the flanks.

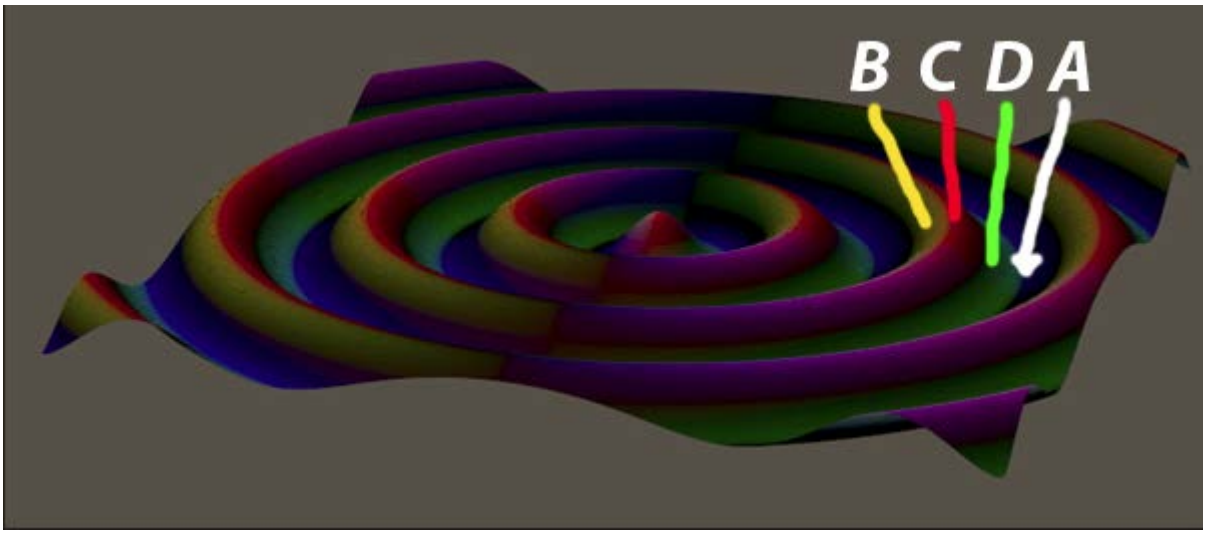
Why do we get those colors? Let's take the first case, start in the middle and gently walk to the right, the positive U direction. All normals will point up then, with no inclination forward or backward. This implies that two adjacent normals will have the same Z-value (being 0.0 in this case), so any difference between them will be 0 in Z-direction. When the resulting XYZ vector then is interpreted as a RGB color, it will show no Blue. For walking in that positive U direction, that is.

Meanwhile, the normal vector is UP at the lowest position, then leaning backwards (X negative, Y positive) when going uphill, then going straight UP again at the highest position, then leaning forward (X positive, Y positive) when going downhill, and then straight UP again. See the red and blue vectors in the illustration below:

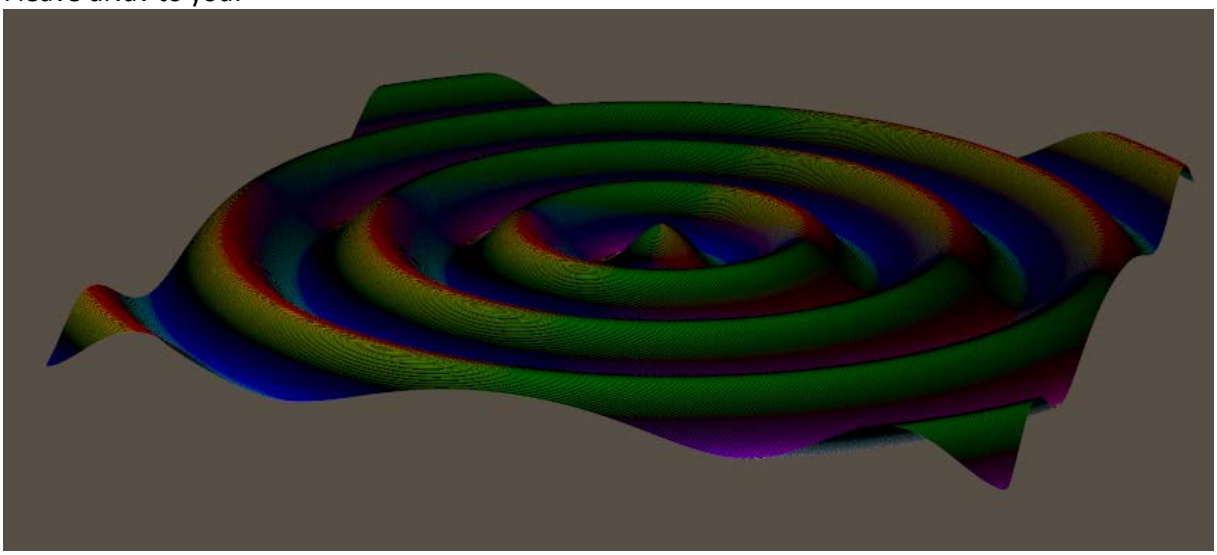


The blue normals are exaggerated a bit, to illustrate the next point: while walking over the top half of the hill (B and C), the normal will lean less backward (B), and then more forward (C). So the difference (green arrows) between those adjacent normals is upwards to the right (B) and then downwards to the right (C). But while walking over or into the valleys (A and D) the normals lean backward more and more. So the differences between adjacent normals is to the left, downwards (A) or upwards (D).

Translating this to colors, what can we expect?
A. downwards left: negative X (left), negative Y (down) and zero Z. The negatives are clipped to 0 when turned into colors, so the result is RGB = (0,0,0) aka Black.
B. upwards right: positive X, positive Y, and zero Z. That makes Red + Green + no Blue => Yellow.
C. downwards right: positive X, negative Y, zero Z makes Red + no Green + no Blue => Red
D. upwards left: negative X, positive Y, zero Z makes no Red, Green, no Blue => Green.



I leave dNdv to you:



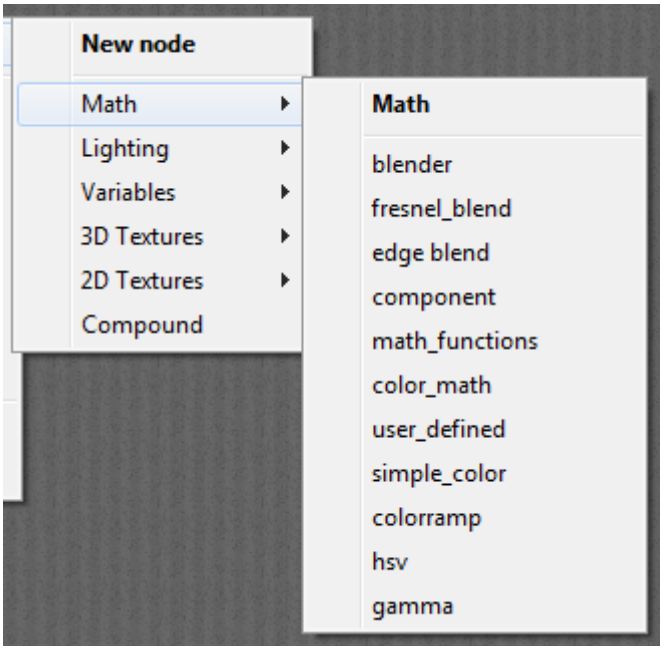
Note that you've got to walk to the back, the positive V direction, which makes Z increasingly negative while X remains 0.

81 What are the Math nodes used for?

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

Advanced

The Math group offers various nodes that can adjust and combine colors, images and other texture elements.



In this article, and some following ones, these nodes are discussed in subgroups. In this article:

- Blender
- Fresnel_blend
- Edge blend

which mix two color, image or texture inputs into one result. In 82 What are the Component and Math_functions / Color_math nodes used for?:

- Component
- Math_functions
- Color_math

which split and combine value or color / image inputs in a mathematical way. In 83 What are the User_defined, Simple_color and Colorramp nodes used for?:

- User_defined
- Simple_color
- Colorramp

which generate colors from value inputs, and in 84 What are the HSV and Gamma (Math) nodes used for?:

- Hsv
- Gamma

which modify image characteristics like brightness and saturation.

Blender

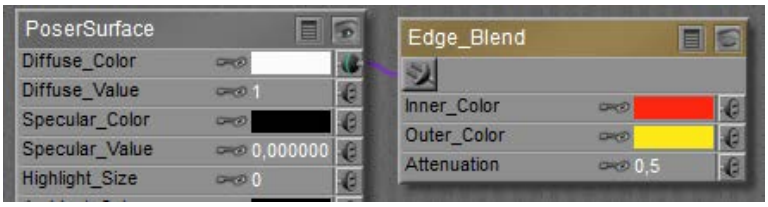
This node takes two inputs, color or image or whatever, and blends them together. When the Blending value equals 0 the result is entirely **Input_1**, when the value equals 1 the result is entirely **Input_2**, and the real blends are made by intermediate values.

Of course, for static **Blending** values there are other ways of combining colors or images. But the power of this node is that the blending can be made dependent from space and time, by feeding any node from the 2D Textures (a pattern or blending image map), 3D Textures (a fractal) of Variables (U, V, frame number) into it. Note than when an image-map is attached, this one should have its Gamma

value set to 1.0 explicitly (Poser 1 or any PoserPro), to prevent the Gamma mechanism from interfering with the blending ratios.

Edge_Blend

This node blends two inputs depending on the angle between the surface and the camera. When the camera looks straight (perpendicular) upon the surface the **Inner Color** will show, and at the edges, at skew angles towards the camera, the **Outer Color** shows.



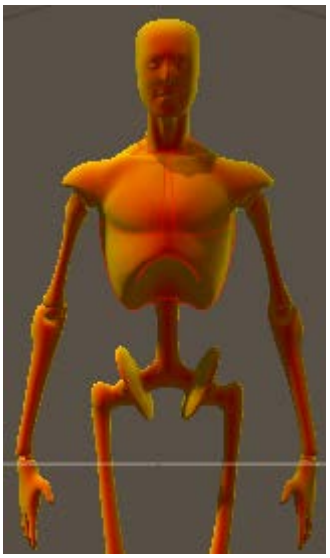
makes

When I take a 2D perspective on the result, the edges are “out” and the area between the edges is “in”. I can also take a more volumetric approach; the more I look upon the middle portion the more transparent an object will be and the more I’ll see from the Inside, while at the edges I’ll see more of the surface itself, the Outside. Both give me a handle on the meaning of Inner and Outer Colors.

The **Attenuation** drives the (exponential) transition function. At minimum 0.0 no outer color will show, but at increasing values the edge effect will become apparent:

At 0.1:

and at 0.9:



Edge blends make velvet effects, and make more realistic transparencies and reflections (the Fresnel effect).

Fresnel_Blend

Since Edge_Blend was used frequently to approximate a Fresnel effect, additional nodes were added to recent Poser versions to help users out. One of those is Fresnel_Blend, which also takes **Inner** and **Outer Color** and an appropriate **Index of Refraction** (IoR). Instead of the (exponential) transition offered by Edge_Blend, the actual Fresnel function is used.

See 32 WHAT'S A PROPER VALUE FOR REFLECTIONS? for real-life IoR values. This Fresnel effect in real life concerns the balance between Reflection on one hand (high at the edges) and Refraction (with embedded Transparency, low at the edges) on the other hand. So combining Fresnel_Blend with those respective nodes is an obvious thing to do.

For that reason, an explicit Fresnel node is offered which offers exactly that. See 63 WHAT'S THE RAYTRACE > FRESNEL NODE USED FOR? for details. Unfortunately, all those reflections and refractions are very computation intensive, and require vast amounts of render time. When I want to compromise, to cheat, to take shortcuts, then this Fresnel_Blend node can offer a nice way out.
And when I like to make things even more complex, like combining reflections with specularity and assign Fresnel to this combination, or when I want to get Fresnel effects into the PoserSurface Transparency itself. Then I need this Fresnel_Blend node as well.

82 What are the Component and Math_functions / Color_math nodes used for?

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

Advanced

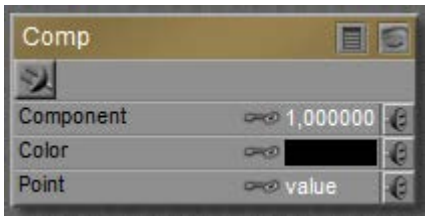
The Math group offers various nodes that can adjust and combine colors, images and other texture elements. This article discusses:

- Component
- Math_functions
- Color_math

which split and combine value or color / image inputs in a mathematical way. See 81 WHAT ARE THE MATH NODES USED FOR? for an overview of the Math group.

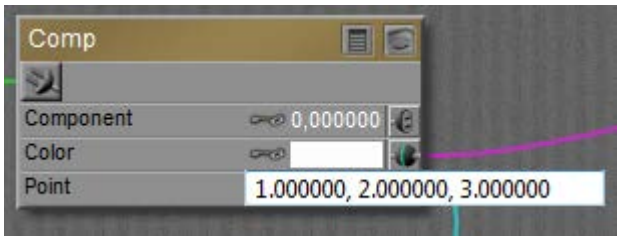
Component

This node accepts a vector, either representing a color (Red, Green, Blue) or a pointer in space (Variable nodes, like P,N and dPdu etc., see 77 WHAT ARE THE VARIABLE NODES USED FOR? and following articles on them). From that vector it picks the first, second or third **Component** (using values 0, 1 or 2 for that) and presents that value as a result.



How do the inputs work together to make an output?

- When something is plugged into the **Color** slot, that takes precedence over anything else. The input is filtered (multiplied) by the color swatch, and the appropriate component of that RGB value triplet is passed through. All **Point** information is ignored.
- When nothing is plugged into the **Color** slot, but something is plugged into the Point slot instead, the appropriate component of that input is passed through. The **Point** value and the Color swatch are both ignored.
- When nothing is plugged into any slot, the appropriate component of the **Color** swatch is passed through, the **Point** value is ignored, unless…
- Nothing is plugged in and the **Color** swatch is complete black (0,0,0),then the appropriate component of the **Point** value is passed through. Note that this value is a value triplet (point vector) of some sort.



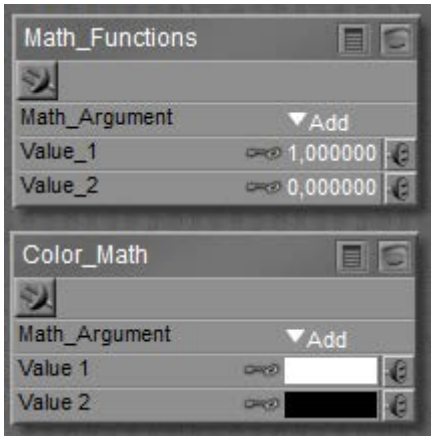
Math_Functions and Color_Math

Both nodes produce a result from a math operation on two inputs. They both offer the same list of choices for such operations. The difference between them is:

- Math_Functions produces a single value, and turns every input into a single value as well before the operation is performed. So, when I offer it a color or a pointing vector, that is turned into a value first (for colors: the brightness it calculated and used). The math is performed as one might be familiar with, just using function calls or performing algebra and the like.
- Color_Math produces a three-component vector, based on such color- or pointing vectors for input as well. Single values are turned into a vector with three equal components, like a

value is turned into a greyscale tone.
The math is performed on a per-component basis, that's the Math_Functions operation three times:

- Result 0 = math operation on inputA comp 0 and inputB comp 0
- Result 1 = math operation on inputA comp 1 …
- Result 2 = …



Most functions do what I can expect from them, as defined by any basic math textbook. See the Poser Reference Manual on them in case of doubt. Some operations need both inputs (like Add and Multiply), other operations work on **Input_1** only (like sine, square root).

The **Smoothstep**, **Bias** and **Gain** operations expect an image for Input-1. Smoothstep then will perform the bicubic enhancement which is about standard when an image is resized and resaved in my imaging software. Bias and Gain expect a value (or three-value vector) for **Input_2** as well, and will adjust image brightness (Bias) and contrast (Gain) accordingly. The value 0.5 is considered neutral (no net effect), lower values (down to 0) reduce brightness/contrast, higher values (up to 1) increase them. Negative contrasts will invert the image input.

Note that all images and all color swatches can be affected by the Gamma mechanism (Poser 10, or all Pro versions) at render time, and hence the outcome will be affected as well, unless explicit measures are taken to prevent that. Images should have their Gamma value set to 1.0 and swatches should have brightness 1.0 (that is: at least one color component should equal 100%).

83 What are the User_defined, Simple_color and Colorramp nodes used for?

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

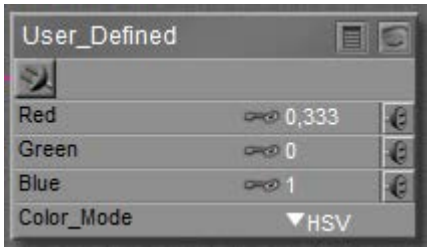
Advanced

The Math group offers various nodes that can adjust and combine colors, images and other texture elements. This article discusses:

- User_defined
- Simple_color
- Colorramp

which generate colors from value inputs. See 81 WHAT ARE THE MATH NODES USED FOR? for an overview of the Math group.

User_Defined



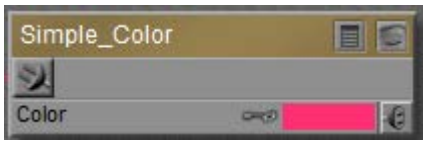
This node produces a color, that is: a three value RGB vector with each component between 0.0 and 1.0, from three independent input values.

When **Color_Mode** is set to RGB, the input values are just clipped to the 0..1 range and passed through. In the HSL or HSV modes, the inputs are clipped into the 0..1 range as well, and then translated to RGB colors. For instance, Hue=0 (or 1 or 2 or …) implies Red, Hue = 0.5 implies Cyan, Hue = 0,333 is Green. The input called **Red** contains Hue, **Green** contains Saturation and **Blue** contains Lightness or Value respectively.

In the HSL model, the Saturation starts at 0 with grey and mixes more and more pigment into it, until it's saturated at S=1. The taint of grey is determined by the Lightness: 0 for full black to 1 for full white. So full red requires (0,1,any) for input.
In the HSV model, Saturation determines the mixture with White (0 = white, 1 = full pigment) while Value determines the mixture with Black (0= black, 1 = full pigment). Full red requires (0,1,1).

Note: a color generated by this node will not be image-driven or color swatch-driven, but value-driven. As a consequence, it will not be pre-adjusted when renders are done with Gamma Correction ON (Poser 10, or Pro all versions, see 98 WHAT'S THE FUZZ ABOUT GAMMA CORRECTION? for details), and results will look brighter than for color swatches representing the same color.

Simple_Color



This node produces… a simple color, as set by a color-swatch. Or a simply color-filtering of anything inputted to it.

This is useful when a single color is used multiple times in a materials node tree. Now it can be defined once and managed at one place. It's simpler than using the User_Defined node, which requires three independent input values for R,G and B separately.

But there is another serious difference: when rendering under Gamma Correction, as available from Poser 10 and Pro 2010 and up, all color swatch defined colors are affected, but numerical settings as from User_Defined are not. This might give different results for a same color, when set in different ways. See 98 WHAT'S THE FUZZ ABOUT GAMMA CORRECTION? for details

Colorramp

This node produces a color from a four-point gradient, having the color-points set at 0%, 33%, 66% and 100%. The interpolated color is derived from the **Input** value, which should be between 0.0 and 1.0. I can plug in a greyscale image in here as well, to get it colored up.

Note: when rendering with Gamma Correction, the swatches will be pre-adjusted (darkened) but each at a different rate, and the Input-value will not. As a consequence, the resulting hues produced by this node will differ, depending on whether GC is used or not. See 99 GAMMA, TRANSPARENCY AND BLENDS for details, and for eventual solutions.

84 What are the HSV and Gamma (Math) nodes used for?

Nodes are the essential building blocks in the Advanced interface to the Poser Material Room. They are the graphical representation of mathematical function calls, that is: calculation procedures which turn parameters (inputs) to a result (output).

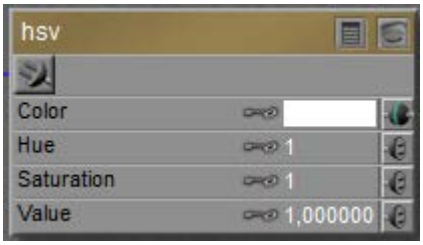
Advanced

The Math group offers various nodes that can adjust and combine colors, images and other texture elements. This article discusses:

- HSV
- Gamma

which modify image characteristics like brightness and saturation. See 81 WHAT ARE THE MATH NODES USED FOR? for an overview of the Math group.

HSV



This node takes a **Color** from its input, usually an image, translates is into HSV mode, adjusts it according to the node settings and delivers the result back as a regular RGB color.

In (its internal) HSV mode, hue determines the color from a 0.0 to 1.0 value. 0.0 represents Red, 0.5 represents Cyan, and at 1.0 we've made full circle and are back to Red again. This node takes the hue from the input color, and adds (**Hue** -1) to its already existing value. So when the Hue value is set to 1.33 it will turn Red (hue= 0.0) into hue= 0.0 + (1.33 – 1) = 0.33 is Green. A value 0.67 will turn Red into 1.0 + (0.67 – 1) = 0.67 is Blue. So an input value 1.0 is neutral, has no effect.

In (its internal) HSV mode, saturation and value determine the mixture of White and Black respectively, with full color pigment. A 0.0 value means no color, a 1.0 value on both means full color. As with Hue, the node values of 1.0 for **Saturation** and **Value** have no effect, the difference from 1 is added to (or subtracted from) the original input value.

Gamma

The Gamma node helps me to apply or undo Gamma Correction on specific images and colors, next to an eventual Gamma mechanism applied to all of them (Render Setting). See 98 WHAT'S THE FUZZ ABOUT GAMMA CORRECTION? and 99 GAMMA, TRANSPARENCY AND BLENDS for details, but for short: the images, movie frames and colors handled by Poser are 'in order', for input as well as output, and need no adjustment as such.

Poser is using the Gamma 'thing' for correcting on its own flaws:

- Direct lights and especially point- and spotlights produce far too deep shadows, while in nature indirect lighting and atmospheric scattering soften shadows a lot. In a similar way, shading effects like Lambert as standard applied in Poser Diffuse (see 30 CAN I GET A (SIMPLIFIED) EXPLANATION ON LAMBERT AND DIFFUSE SHADING?) are too strong and too dark.
  This is a generic issue in all CG rendering, not only Poser. Unfortunately, the darkening makes images loose contrast and detail in de darks, which is disliked a lot by most people. That's a deep down natural thing: the darks hides hunters as well as prey, and we like to be aware.
- Material channels like diffuse and specular are simply added up, multiple lights are simply added up, and all this contributes to overlighting and far too strong highlights (reflections, etc.).

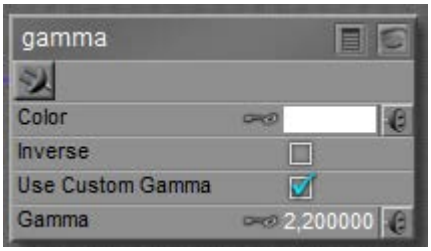
So, as discussed in 20 A POSERSURFACE MATERIAL OFFERS DIFFUSE, SPECULAR, ETC. HOW DO THESE WORK TOGETHER? as well, Poser (*) first applies (**) Inverse Gamma Correction to darken (the midtones from) images, frames and color swatches (***), then it renders, and then it applies Gamma Correction to brighten up the result. This leaves the main coloring intact, but softens all effects from shadows, shading and highlighting, and produces a far more pleasant and more natural result which is favored especially in photorealistic render styles.

(*) from Poser 10 and Poser Pro 2010 up, by checking the appropriate Render Setting.

(**) it should not be applied to images which define Bump, Displacement, Transparency or any blending. For those, the neutral / no effect value of 1.0 should be used. For all others, any other value is appropriate, the larger, the stronger the softening effect. The industry standard value of 2.2 is recommended but actually, as images handled by Poser are 'in order' and Poser is using the mechanism for its own purposes, any value will do. Values over 1.0 will soften shadows and highlights, values around 2 make a natural impression, and really high values (4.0 and up) might cause artifacts. Render styles which require hard shadows, like comics, do not need the adjustment at all.

(***) Value slot are not affected. As a consequence, a Diffuse (Color, Value) set to (white, 50%) will produce a result different from (50% white, 100%). Also, the output of the Simple Color node will be affected, but the output from the User_Defined node will not, and the image plugged into the HSV node will be adjusted before (!) the HSV adjustment is made itself.

The image plugged into **Color** can be color-filtered, and **Gamma** values over 1.0 will brighten the midtones as the node is applying regular Gamma Correction (in math: output=input $^{1/gamma}$). This is to take any Poser adjustments on the input out again before further handling. With **Inverse** checked, midtones will be darkened as the Inverse Correction is applied (in math: output=input ^{gamma}). This is to put the regular Poser adjustment into colors etc. which are missed by Poser itself, like a User_Defined color. Or to undo the adjustments of a Gamma node applied earlier in the materials node tree.

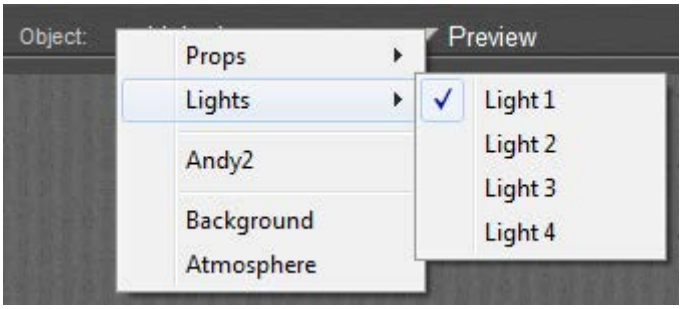


Use Custom Gamma means: use the **Gamma** value as explicitly defined in the node. Unchecking this option will use the value as defined in Render Settings. As the latter is recommended for consistency within the scene, checking is meant for introducing Gamma effects in case Poser does not.

Note: although Gamma Correction is available from Poser 10 and Poser Pro 2010 up, the Gamma node itself is around in earlier Poser versions as well. The node can be used to adjust the brightness of image maps used in the scene (like the HSV node), and to build elaborate node trees that address the first portion of the Gamma Correction procedure as supplied in later versions. The node cannot perform the final Inverse correction, as is applied by the renderer to the result.

V Materials for Non-Objects

This section discusses the materials definitions for Atmosphere, Background and Lights. These are not objects with a surface, but do have properties which are handled in Material Room, via the Object selector:



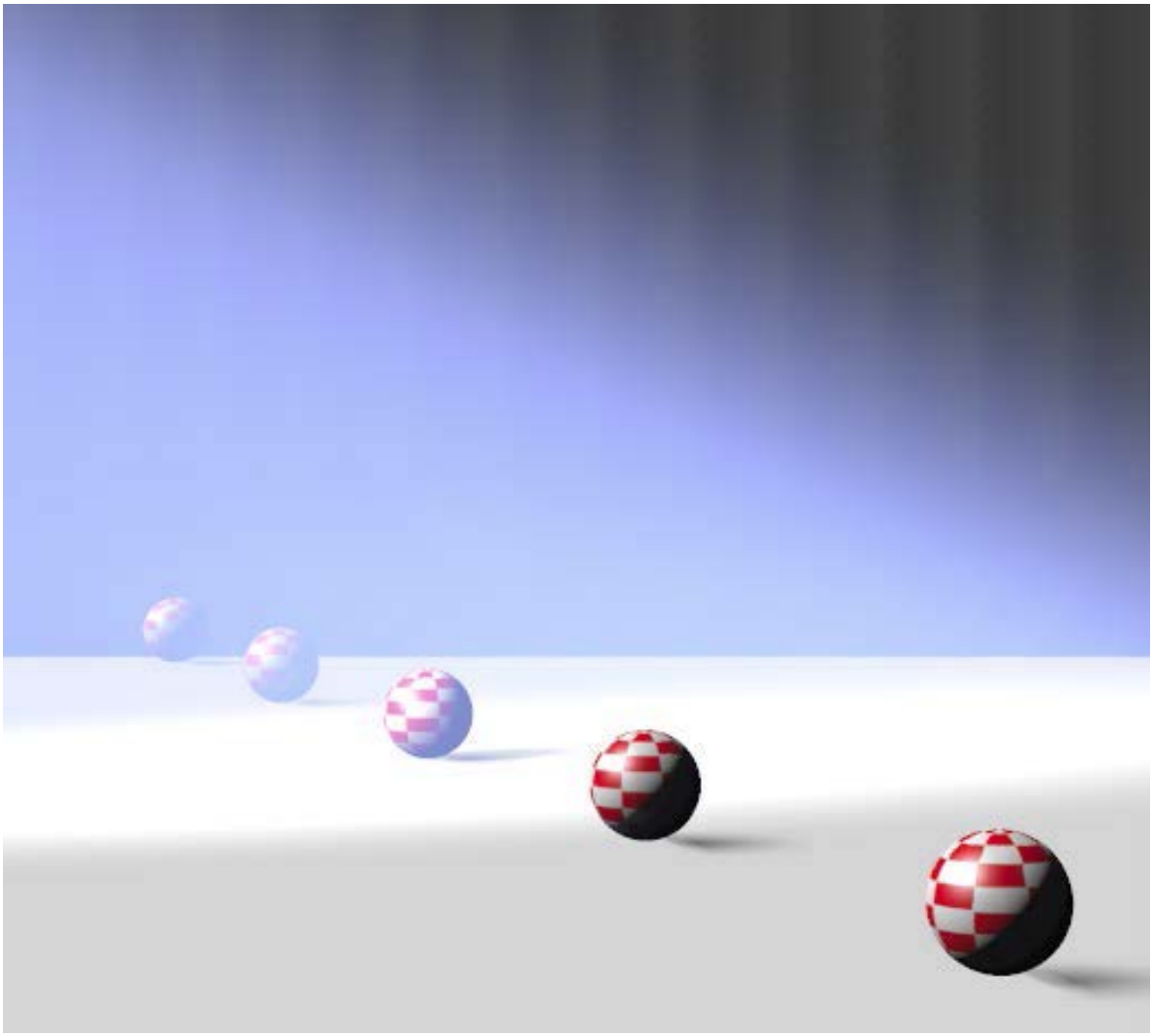
- The scene atmosphere is discussed in
  - 85 (How) can I create a smoky or foggy atmosphere in my scene?,
  - 86 (How) can I create volumetric spotlights in my scene? and
  - 87 My atmosphere or volumetric lights don't show up, how come?.
- The scene / render background (color, picture or movie) is discussed in
  - 88 (How) can I make a "green screen" shot, to add backgrounds in post?,
  - 89 (How) can I use a reference picture in the preview only?,
  - 90 (How) can I add a background picture in my render?,
  - 91 (How) can I rotoscope against a movie in the background? and
  - 92 An Overview on Backgrounds.
- The coloring properties of Lights are discussed in
  - 93 How do I set the color and intensity of my light(s)?,
  - 94 Can I use a light for diffuse or specular lighting only?,
  - 95 Can I use a light for preview only? and
  - 96 Can I use a light for a specific object only?.

Most of those topics are considered Intermediate level, although various configurations can be setup via Material Room menus, and can be managed through the Simple interface. On the other hand, managing the details of a scene Atmosphere requires the use of 'cloudy' nodes from the 3D Texture group 72 The 3D Cloudy texture nodes, which by themselves are considered Advanced.

This section concludes with some varied, advanced topics like mapping for IBL (in 97 How are Image Based Lighting, probeLight objects and IDL Sky Domes related?) and Gamma Correction (in 98 What's the fuzz about Gamma Correction? and 99 Gamma, Transparency and Blends).

V1 Poser Atmosphere

The first group of articles is on Atmosphere. I can create haze and fog, and volumetric effects for (spot)lights, at the cost of increased render time, of course.



- 85 (How) can I create a smoky or foggy atmosphere in my scene?,
- 86 (How) can I create volumetric spotlights in my scene? and
- 87 My atmosphere or volumetric lights don't show up, how come?.

85 (How) can I create a smoky or foggy atmosphere in my scene?

Although various configurations can be setup via Material Room menus, and can be managed through the Simple interface, this topic is considered Intermediate level. Next to that, managing the details of a scene Atmosphere requires the use of 'cloudy' nodes from the 3D Texture group (72 THE 3D CLOUDY TEXTURE NODES), which by themselves are considered Advanced.

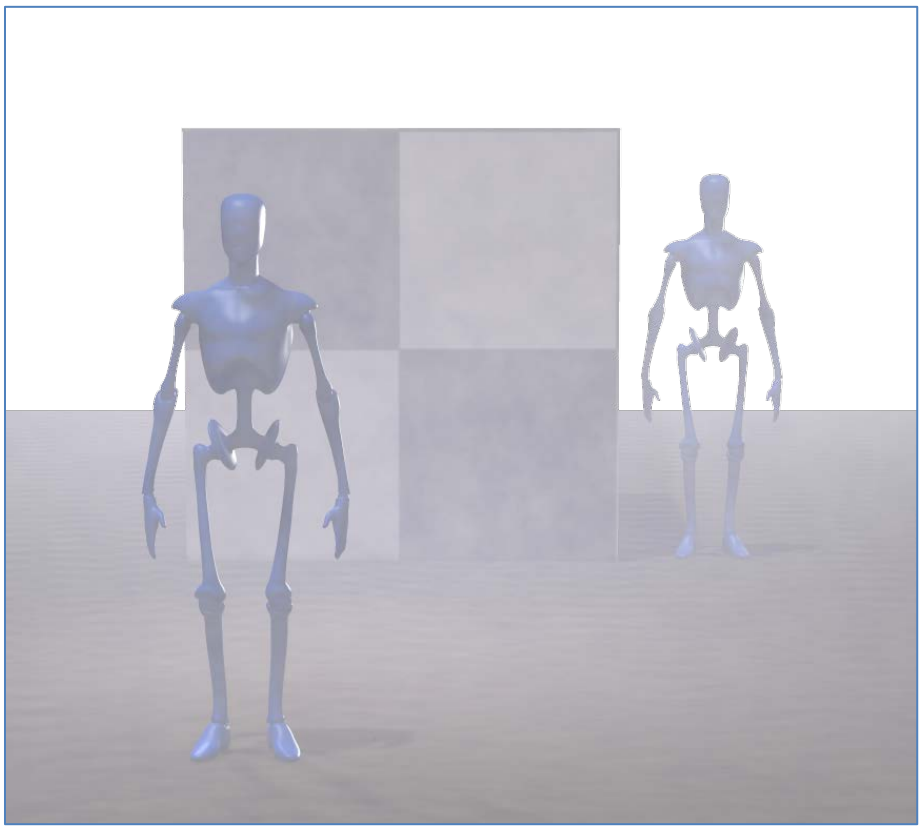
Intermediate

First, be aware that atmospheric effects can be created in post as well, while when done in Poser it might imply a serious extension of my render time.

A tutorial on Poser Render Passes, including atmospherics post work, can be found at http://www.book.artbeeweb.nl/?p=388
A tutorial on Poser atmospherics can be found at http://www.book.artbeeweb.nl/?p=3093

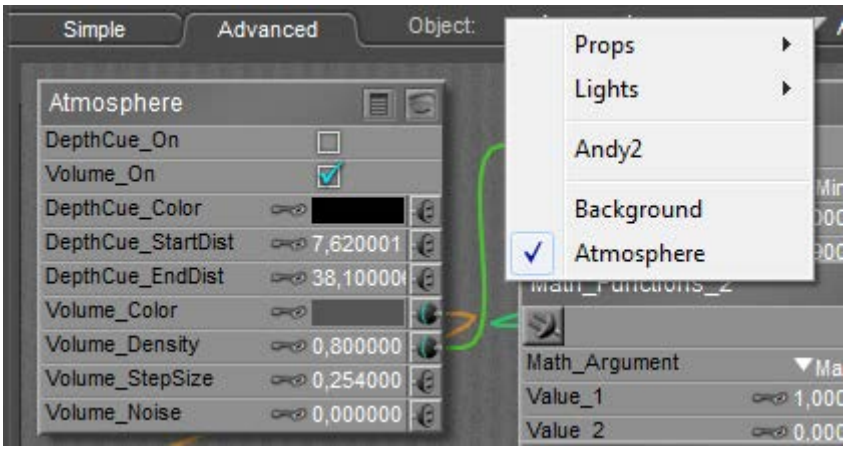
Second, Poser atmospherics reveal themselves only in front of an object in the scene. Any object bouncing light towards the camera will do. But in front of just a background color, a background picture or a background movie frame, the atmospherics remain invisible. Poser atmospherics do need a visible object facing the camera to get uncloaked. The color of the object is irrelevant, or better: the darker the object, the better a whitish fog will be visible.

For example: a nearby and a faraway character, a B/W colored box, the default ground floor and an "empty" background make:

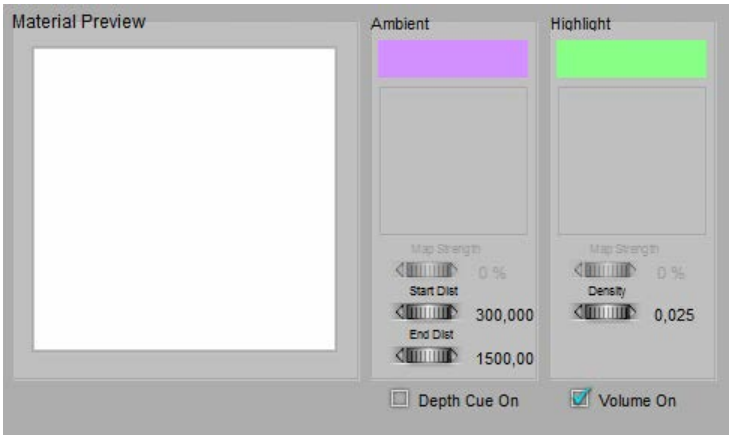


So, how to do it?
I enter the Material Room, click the [Create Atmosphere] button and select one.

How to alter the settings?
In the Material Room, I use the list next to Object and select Atmosphere.



When using the Simple interface, the material looks like:



Note that the Simple Interface contains some bug (all Poser versions up to Poser 10 / Pro 2014), see below.

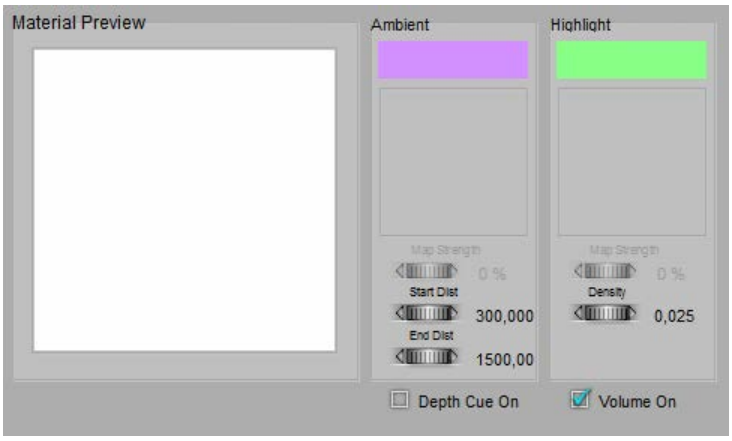
How to switch it off?
The Atmosphere main node supports DepthCue as well as Volumetrics. Both can be switched on and off independently.

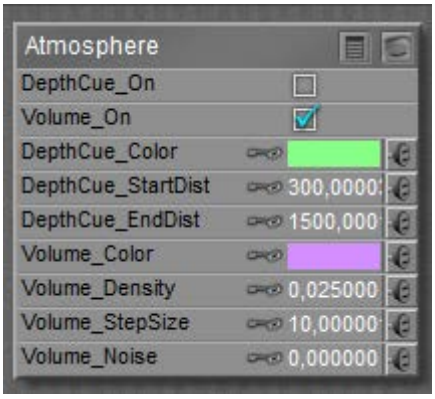
What about DepthCue and Volume?
DepthCue is a quite uniform haze effect which colorizes the scene, depending on the distance of the surface to the camera.
Volume is a smoky, cloudy fog effect which is located within the scene, not related to the camera position. Tip: in order to make an object or character disappear into the darkness, a dark fog can be used instead of a whitish one.

Advanced interface

When using the Advanced interface, I learned that

- Volume Step-Size and –Noise are not supported in the Simple Interface. These are quality-settings for better render results (at the cost of longer render times).
- The [Create Atmosphere] button can produce quite some node-tree to define such a material. 'Cloudy' nodes from the 3D Textures group will be present to drive the look and feel of fogs and alike. See 72 THE 3D CLOUDY TEXTURE NODES about them.
- From the Simple interface, Ambient matches Depth-Cue and Highlight matches Volumetrics. But I also find a bug (Poser 10 / PoserPro 2014 and before): the color and image-map assignments are reversed.



=>


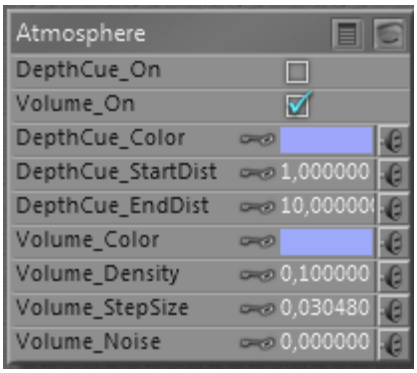
(The bug is reported to Smith Micro and will be addressed, some day).

86 (How) can I create volumetric spotlights in my scene?

Although various configurations can be setup via Material Room menus, and can be managed through the Simple interface, this topic is considered Intermediate level.

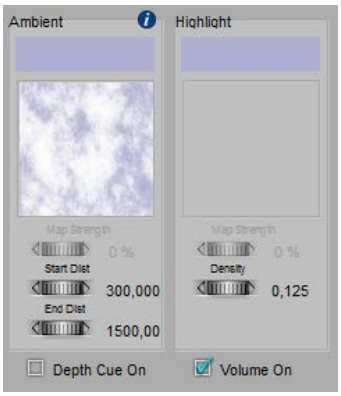
Intermediate

Spotlights with a volumetric effect need a foggy atmosphere to come into existence. See 85 (How) can I create a smoky or foggy atmosphere in my scene? on this.



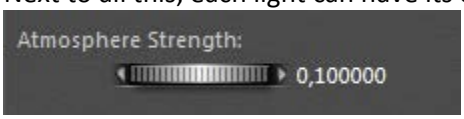
As Depth Cue relates to the camera, so does Volume relate to the lights. Volume effects can be switched on/off themselves too, so they can be set independent of the Depth Cue effects.

The main parameters are **Volume Color**, and **Density** (*). When a direct light illuminates a volume in the scene, that volume acts like a transparent fuzzy object with that specific internal color. The lower the Density the more transparent it seems.



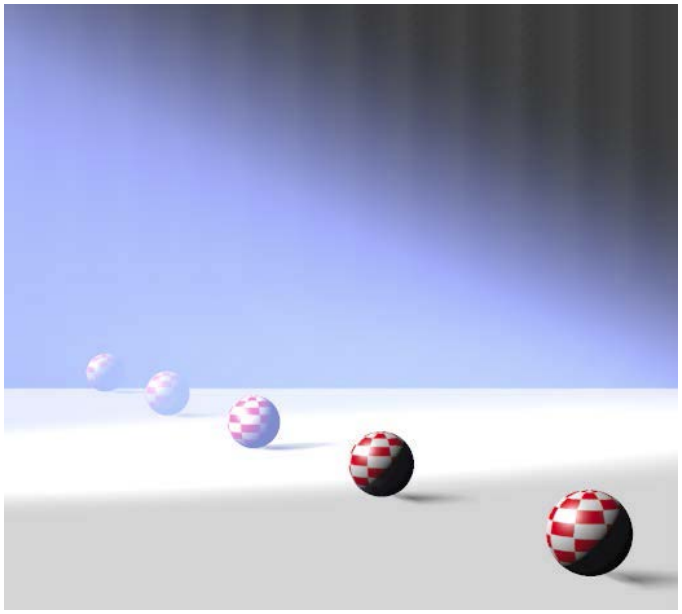
(*) Volume Color is referred to as Ambient Color in the Simple interface, while Density is shown in the Highlight column. This confusing presentation is known as an issue, and might be addressed in later versions of Poser.

Next to all this, each light can have its own Atmosphere Strength parameter:



So some lights can interact more than others. For example:

One infinite white light, Atmosphere Strength as low as 0.000010 plus one white spotlight, angular falloff from 10 to 20 (narrow light bundle), Atmosphere Strengths as high as 0.100.
From the different Atmosphere Strength settings of the lights one can discriminate the spotlight from the overall scene lighting. The bluish color is from the Volume settings.



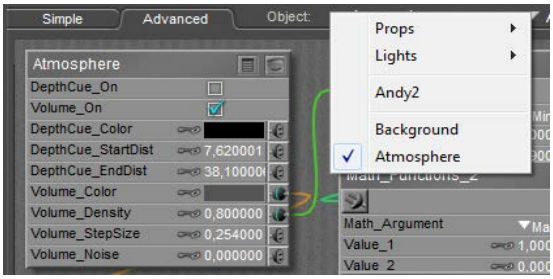
Do note that especially Volume effects take some time to render. A larger stepsize speeds up the calculations at the cost of quality and detail. Increasing the Noise parameter helps to improve on the quality especially at larger stepsizes. Noise and Stepsize are available in the Advanced interface only.

87 My atmosphere or volumetric lights don't show up, how come?

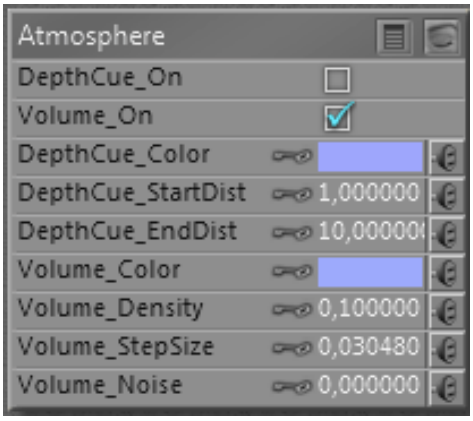
Although various configurations can be setup via Material Room menus, and can be managed through the Simple interface, this topic is considered Intermediate level.

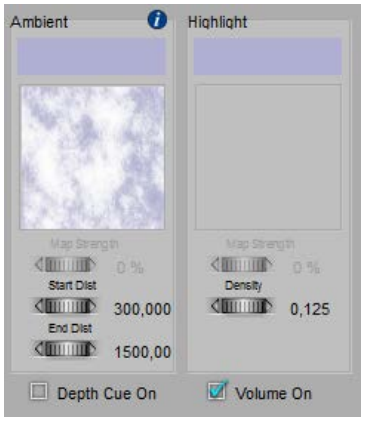
Intermediate

Various causes can cloak my atmospherics:



- In the Atmosphere main node, the DepthCue or Volume checkbox is not ticked.
  action:
  - go Material Room
  - open the Atmosphere main node
  - tick the appropriate box



or


- In the scene, there is no object behind the atmosphere
  action: put a backdrop object (e.g. a plane) or a surrounding object (e.g. a dome) in the scene

- The light has a too low value for Atmosphere Strength
  action: open the Properties for the Light, and increase its Atmosphere Strength

V2 Poser Background

The second group of articles is on Background. I can preview as well as render against an image (photo's or mattes), against footage (rotoscoping) or against a color (green screen shots).

- 88 (How) can I make a "green screen" shot, to add backgrounds in post?,
- 89 (How) can I use a reference picture in the preview only?,
- 90 (How) can I add a background picture in my render?,
- 91 (How) can I rotoscope against a movie in the background? and
- 92 An Overview on Backgrounds.

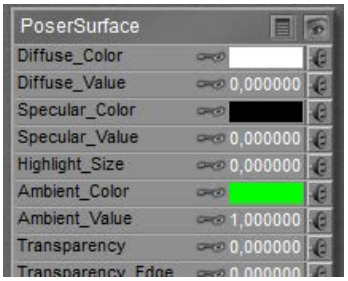
88 (How) can I make a "green screen" shot, to add backgrounds in post?

Although various configurations can be setup via Material Room menus, and can be managed through the Simple interface, this topic is considered Intermediate level.

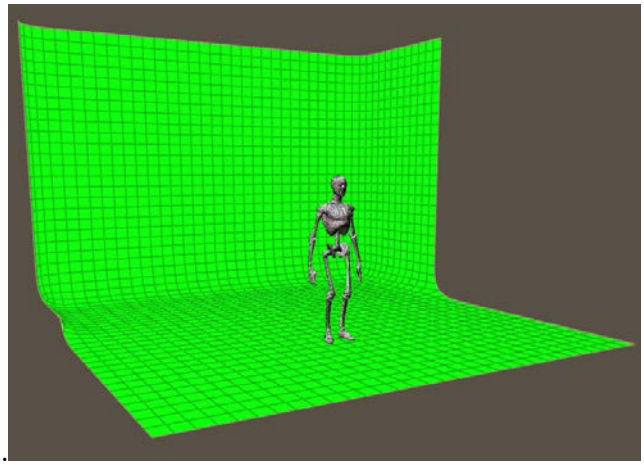
Intermediate

1) Either I put an object at the relevant place in the scene, and give it a uniform (green) texture,

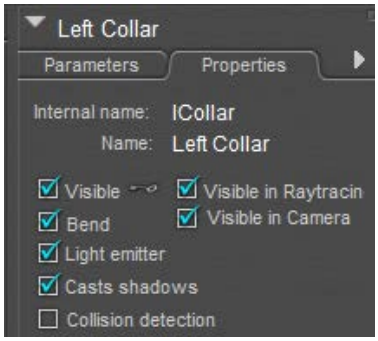
- The object material should be Ambient only.
  It should not support Diffuse, Specular, Transparency, Reflection, Bump/Displacement, etc.
  (unless I want to have the effects of the scene lighting on my backdrop replacement as well)



makes ..



- The object should not cast shadows (unless I want them on my backdrop replacement as well), nor should it act as a light emitter in IDL lighting conditions, nor should it be visible in raytracing (reflections!).
  Those properties should be switched OFF.

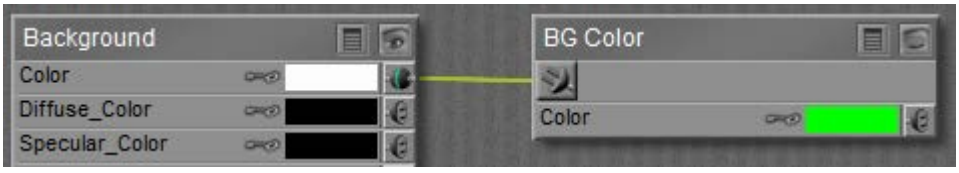


2) Or without any object obstructing the background, I export my image in a format which supports transparency (like PNG),

3) Or without any object obstructing the background, I activate Background Color and assign it an appropriate color.

Either 

Or 

Or 

will do the job.

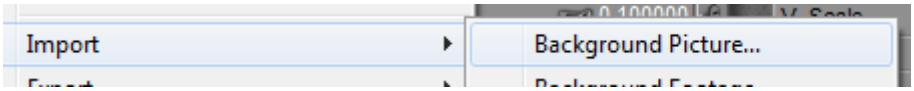
See 92 AN OVERVIEW ON BACKGROUNDS on the background nodes in general.

89 (How) can I use a reference picture in the preview only?

Although various configurations can be setup via Material Room menus, and can be managed through the Simple interface, this topic is considered Intermediate level.

Intermediate

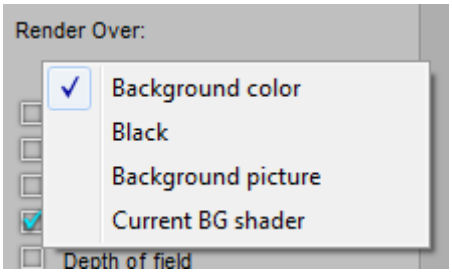
First I set a Background Picture, either via the menu File > Import > Background Picture:



or via the Material Room directly, connecting the BG picture node with the background node:

The BG Picture node is completely compatible to the regular image_map node as discussed in 66 HOW CAN I ASSIGN AN IMAGE TO A MATERIAL?.

Then I switch off the use of the picture in the render, via Render Settings:



That is: do NOT use Background Picture nor Current BG shader.

In Material Room the associated BG Color and Black nodes can be dressed up with any node-tree. So I even can connect a different image to them, and have a picture in my render different from the preview. See 92 AN OVERVIEW ON BACKGROUNDS on the background nodes in general.

When I'm still having problems with the Background, I just check:

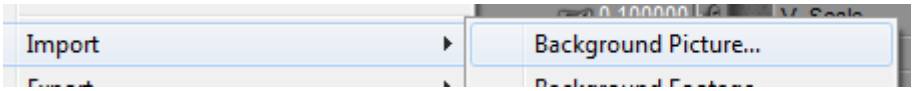
- Whether the GROUND object (or any other object) is obstructing the view
- Whether I'm trying to load an 16-bit-per-color image (don't)

90 (How) can I add a background picture in my render?

Although various configurations can be setup via Material Room menus, and can be managed through the Simple interface, this topic is considered Intermediate level.

Intermediate

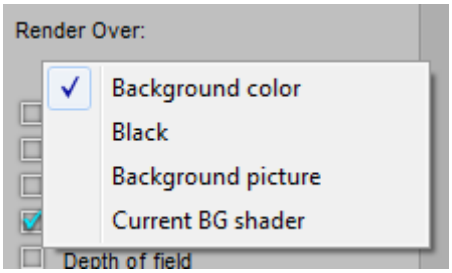
First I set a Background Picture, either via the menu File > Import > Background Picture:



or via the Material Room directly, connecting the BG Picture node with the background node:

The BG Picture node is completely compatible to the regular image_map node as discussed in 66 How can I assign an image to a material?.

Then I switch ON the use of the picture in the render, via Render Settings:



That is: do use Background Picture or Current BG shader. See 92 An Overview on Backgrounds on the background nodes in general.

When I'm still having problems with the Background, I just check:

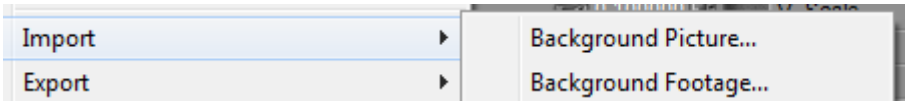
- Whether the GROUND object (or any other object) is obstructing the view
- Whether I'm trying to load an 16-bit-per-color image (don't)

91 (How) can I rotoscope against a movie in the background?

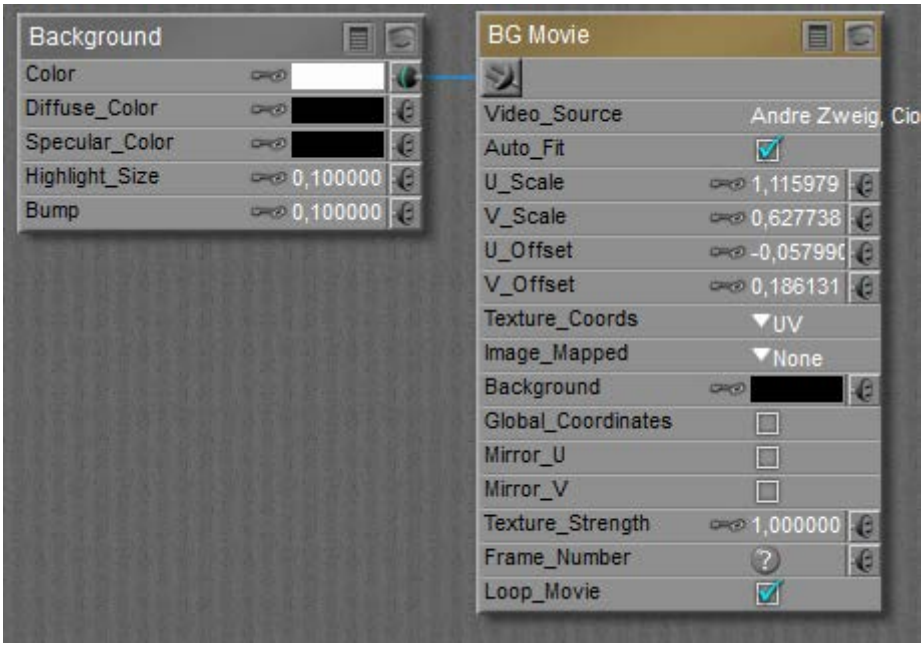
Although various configurations can be setup via Material Room menus, and can be managed through the Simple interface, this topic is considered Intermediate level.

Intermediate

First I set a Background Movie, either via the menu File > Import > Background Footage:

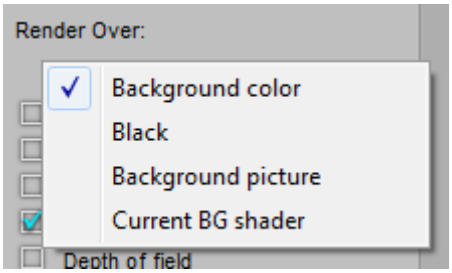


or via the Material Room directly, connecting the BG Movie node with the background node:



The BG Movie node is completely compatible to the regular movie node as discussed in 68 How can I assign a movie to a material?.
Note that I first have to connect BG Movie to the Background node, and then I select the video. When any video is already selected before the connection is made, then that selection will get lost and has to be re-established again. This also is the case when BG Movie is setup properly, gets disconnected for any reason, and is re-connected again.

Then I switch ON the use of the movie in the render, via Render Settings:



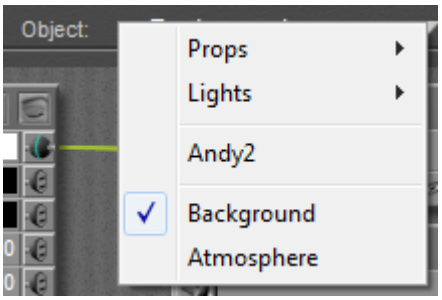
That is: do use Background Picture or Current BG shader. See 92 An Overview on Backgrounds on the background nodes in general.

92 An Overview on Backgrounds

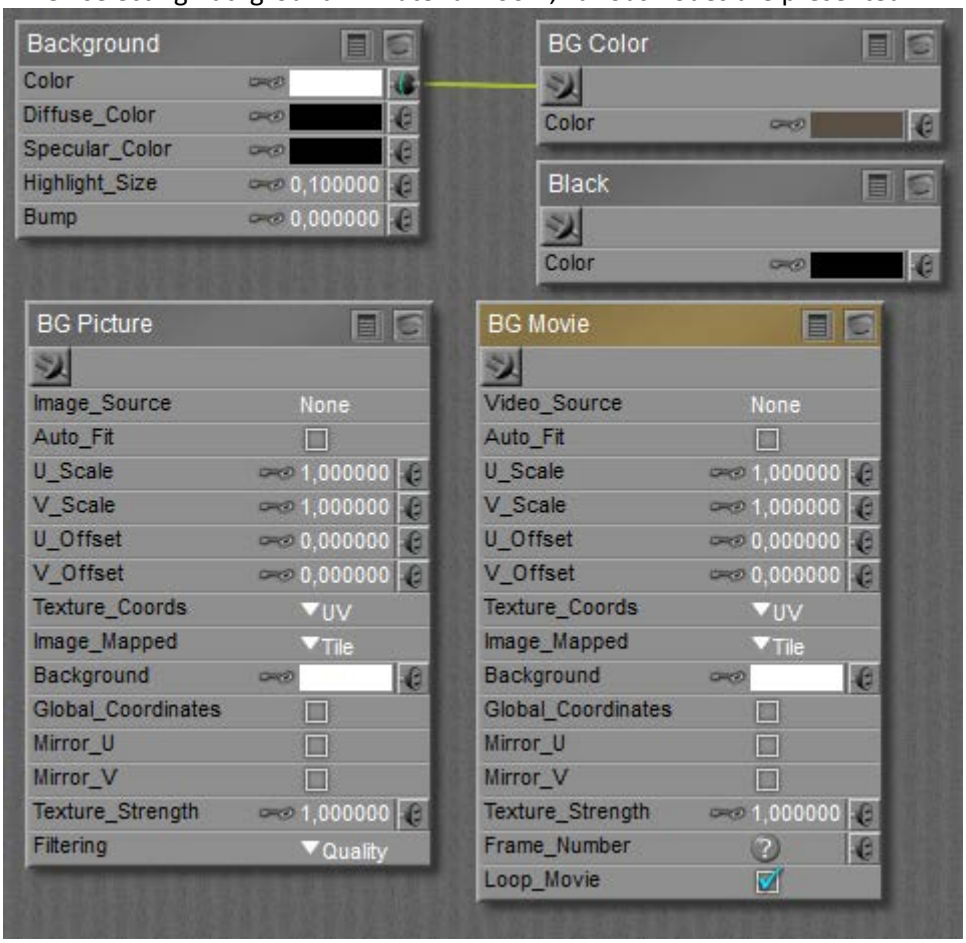
Although various configurations can be setup via Material Room menus, and can be managed through the Simple interface, this topic is considered Intermediate level.

Intermediate

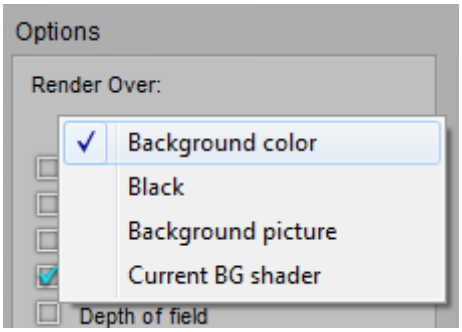
When the renderer cannot fill a pixel of the result due to the lack of light from any objects into the camera, as if the camera is looking into a void, then that pixel is filled with “background” (color, picture, movie frame).
When exporting to a format which supports transparency, such pixels are left transparent. When using a sky dome in IDL lighting conditions, or when using a plane (billboard) -like object with an image assigned to it, then there is always an object in view of the camera, and no background or transparency will be produced.



When selecting Background in Material Room, various nodes are presented:

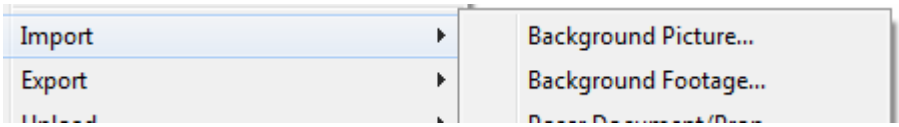


There are various ways to use those nodes in the Render Settings:



And there are various ways to fill those nodes

- Document Picker  ,
- File > Import menu



So, how do all these relate?

The Document Picker has entries for Foreground, Background, Shadow and Ground. The latter fills the Diffuse Color for the Ground object in the scene, Foreground and Shadow are relevant for Preview only. The 2nd from left, Background Color, fills the BG Color node:

 makes 

Importing a Background Picture loads the selected image into the BG Picture node, AND connects that node with the Background node, AND sets Render Settings to (Render Over:) Background Picture. This displays the picture in preview, and in the render.

Importing a Background Footage loads the selected image into the BG Movie node, AND connects that node with the Background node, AND sets Render Settings to (Render Over:) Background Picture. This displays the correct frame in the render, but it's not that good in showing the right frame in Preview, and in adjusting it when stepping through the frames with the animation controls.

Render Settings:

- Background Color and Black picks whatever is produced by the BG Color node or Black node respectively. They do not need to be connected to the Background node. I can set the nodes to any color, and/or attach a complete texture defining node-tree to them.
- Background Picture takes whatever is in BG Picture or BG Movie, whichever is connected to the Background node. If none of them is connected the Color from the Background node is used. If any of them is connected, any further content of the Background node is ignored. E.g. color filtering will not take place.
- Current BG Shader picks whatever is connected to the Background node, and includes the color filtering through the Color swatch of the node.

Handling the nodes directly

I'm free to fill the BG Color and/or the Black node with anything, and attach a complete node-tree to them as well. Any of them can be picked in Render Settings directly. The result of the BG Color node will show in Preview, unless… either BG Picture or BG Movie is attached to the Background node.

I can assign an image to the BG Picture node, it's completely compatible to the regular image_map node as discussed in 66 HOW CAN I ASSIGN AN IMAGE TO A MATERIAL?. It will show in Preview when connected to the Color slot of the Background node. It will show unmodified when selected as Background picture in Render Settings, and it will show color-filtered through the Background nodes Color swatch when the Current BG shader option is used.

I can assign an animation to the BG Movie node, it's completely compatible to the regular movie node as discussed in 68 HOW CAN I ASSIGN A MOVIE TO A MATERIAL?. It will show in Preview when connected to the Color slot of the Background node. It will show unmodified when selected as Background picture in Render Settings, and it will show color-filtered through the Background nodes Color swatch when the Current BG shader option is used.

Notes:

- When playing the animation in preview, Poser tries hard to show to proper frames for background. When stepping through the animation manually however, the background frames are not (always) updated. It's improved in the most recent versions, though.
- I first have to connect BG Movie to the Background node, and then I select the video. When any video is already selected before the connection is made, then that selection will get lost and has to be re-established again. This also is the case when BG Movie is setup properly, gets disconnected for any reason, and is re-connected again.
- Both cases are not bugs but features, aimed at saving resources (computing power, responsiveness, disk space) at caching the individual movie frames during scene development. See 69 HOW DOES POSER HANDLE MY MOVIES FOR TEXTURING? on details.

The Background node at last offers various slots, but after a lot of test renders the conclusion must be that only Color offers a well understood and well determined contribution to the result. The other slots do affect the result somewhat in some way, but are recommended to be left at their default (0 or black) value.
The Color slot, eventually filtering anything which is plugged into it, is shown in the render when Current BG shader is selected in Render Settings. The node does not show in preview, nor do any regular nodes attached to it. However BG Picture or BG Movie do show when plugged into Color, but without any filtering by the swatch.

V3 Poser Lights

The third group of articles is on Lights, which also can have some of their properties managed in Material Room. And next to that, renders result from light interacting with surface material definitions. I can't master the latter without a proper understanding of the other two.

Rendering is not managed from Material Room. That leaves light, as far as color and intensity are concerned.

- 93 How do I set the color and intensity of my light(s)?,
- 94 Can I use a light for diffuse or specular lighting only?,
- 95 Can I use a light for preview only? and
- 96 Can I use a light for a specific object only?.

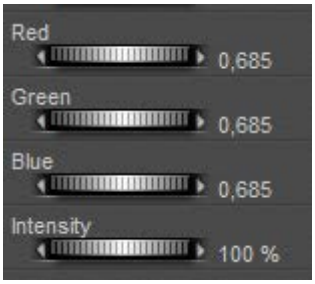
And then Image based Lighting (IBL) and InDirect Lighting (IDL) kick in:

- 97 How are Image Based Lighting, probeLight objects and IDL Sky Domes related?

93 How do I set the color and intensity of my light(s)?

Poser lights can be controlled at various places.

In the Light Controls panel, the dots (bottom left) indicate intensity, I can drag the yellow knob for adjustment. Next to it, mid bottom, is the color indicator. Click it to adjust the color.

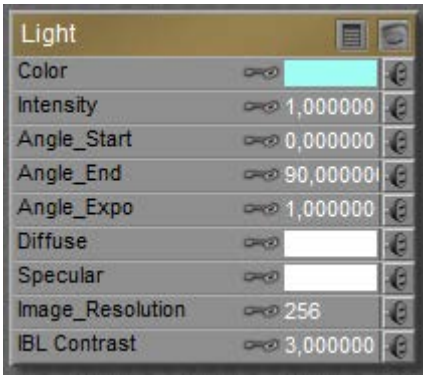


In the properties / parameters of the light, the color and intensity can be adjusted numerically. Red, Green and Blue take values 0..1, but Intensity also takes negative numbers (color inverts the light) and values over 100%.

From Material Room, Simple interface, a color can be assigned, as well as an eventual image-map. The latter is not available in the ways mentioned above. Intensity is not available.

Intermediate

From Material Room, Advanced interface, both Color and Intensity are available and image-maps as well as other kinds of nodes can be attached. Intensity again takes negative numbers and values over 100%.



As usual with all Color/Value pairs in Poser materials definitions, the result is the multiplication (filtering) of both, and their respective inputs:

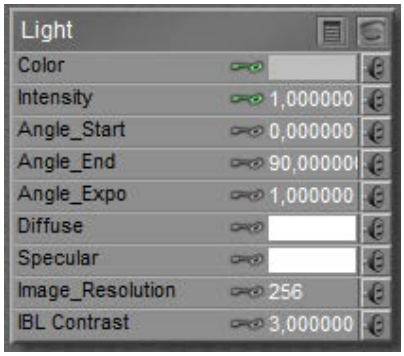
Color * Image-map * Intensity * Intensity (greyscale) map

The Angle-parameters are obsolete, and might serve some compatibility with (very old) Poser versions. The use of the Diffuse and Specular swatches is discussed in 94 CAN I USE A LIGHT FOR DIFFUSE OR SPECULAR LIGHTING ONLY?, they affect the render result only, not the preview.
The last two, Image_Resolution and IBL_Contrast, manage the quality of the IBL (Image Based Lighting) when the light is used as an IBL source, set so in its properties.

94 Can I use a light for diffuse or specular lighting only?

In various cases, it might become handy to have lights that produce highlights only, or just no highlights at all. This can be done easily, but requires the use of the Advanced interface to the Material Room. The options described below are not available through the Simple interface.

Intermediate



When I open the properties of a Light in Material Room, I see

- A generic Color and Intensity
- A Diffuse (color swatch)
- A Specular (color swatch)

The generic Color and Intensity are multiplied to one result as any regular Color/Value pair in the material definitions, and used in preview. The two latter ones are used in the Poser render only, multiplied by the generic Color and Intensity. So:

- When I black out the Diffuse, I'll have a Specular only light in my render.
- When I black out the Specular, I'll have a Diffuse light only in my render
- When I black out both, I'll have no light in my render, but still one in preview.

The first is a necessity in IBL or IDL lit scenes, as these do not produce any specularity themselves (but diffuse light only). Usually, I don't need the extra diffuse part.

The latter is a handy tool in scenes with IDL lighting only. This lighting does not show in preview, so for building the scene I need some lights which won't show at render time.

How the aspects work together:
when the Diffuse is 80% Green, Specular is 60% Blue, Color is 40% Grey and Intensity is 50% than

- When rendering, the light produces a 50% (Intensity) x 40% (Grey) x 80% (=16%) Green diffuse light, plus a 50% (Intensity) x 40% (Grey) x 60% (=12%) Blue specular light
- When previewing, the light produces a 50% (Intensity) x 40% (=20%, Grey) light

Notes:

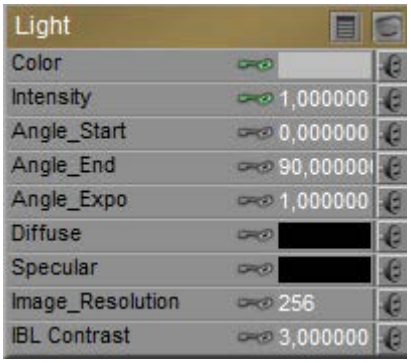
- IBL Lights are supposed to have an image map attached, and do NOT produce any specular lighting despite my settings. IBL lights do show in preview.
- Under IDL (InDirect Lighting) rendering conditions, all objects in the scene behave as a light, but diffuse only. IDL lighting does not show in preview.

95 Can I use a light for preview only?

In various cases, it might become handy to have lights that shine in preview only and do not affect de render result. This can be done easily, but requires the use of the Advanced interface to the Material Room. The options required are not available through the Simple interface.

Intermediate

Open de Light features in Material Room, and black out the Diffuse and Specular entries.



See 94 CAN I USE A LIGHT FOR DIFFUSE OR SPECULAR LIGHTING ONLY? for details.

Note: under IDL (InDirect Lighting) rendering conditions, all objects in the scene behave as a light, but diffuse only (no highlights). IDL lighting does not show in preview. But IBL (image based, direct) lights do show in preview, producing the appropriate local colors as well. So, in order to approximate IDL conditions - like a sky dome – in preview: add an IBL with the same dome image, and have its Diffuse channel blacked out (to have it switched OFF at render time).

Do pay attention to the appropriate orientation of the image, though! See 97 HOW ARE IMAGE BASED LIGHTING, PROBELIGHT OBJECTS AND IDL SKY DOMES RELATED? on this.

96 Can I use a light for a specific object only?

No, Poser does not support selective lighting.

Advanced

However, Poser does support rendering on a per-light basis
(menu Scripts > RenderControl > RenderPasses)

and on a per-object basis.
(menu Scripts > Partners > Dimension3D > Render Content)

This way one can obtain a lot of half-products ("render passes") which in turn can be combined in post. Usually, stacking the images as layers in an image-handling program like Photoshop, with the layers set to Screen mode, is the way forward.

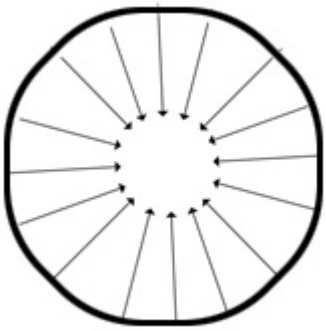
A tutorial on Poser Render Passes can be found at http://www.book.artbeeweb.nl/?p=388 .

97 How are Image Based Lighting, probeLight objects and IDL Sky Domes related?

Although various configurations can be setup via Material Room menus, and can be managed through the Simple interface, this topic is considered Intermediate level.

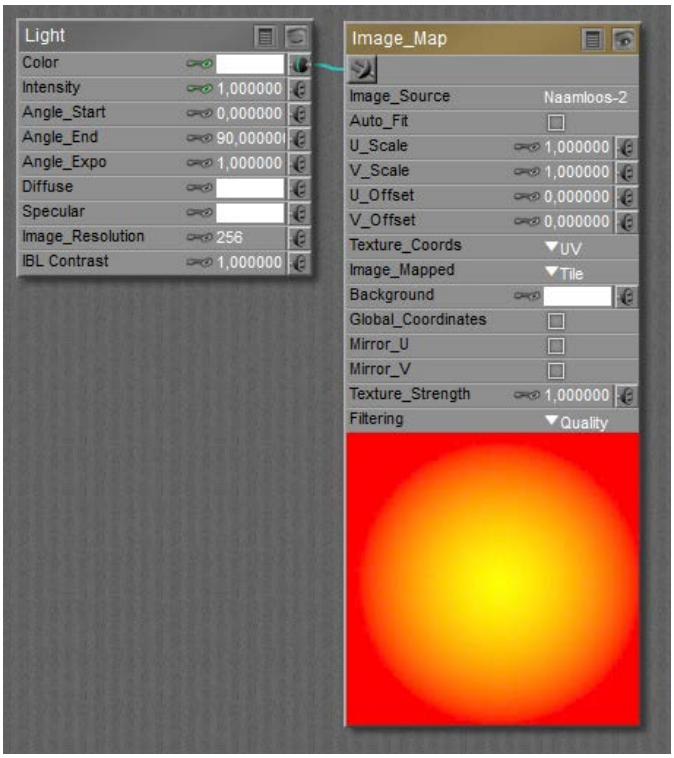
Intermediate

Most scenes require some kind of environmental lighting, and Poser offers various ways to establish that.

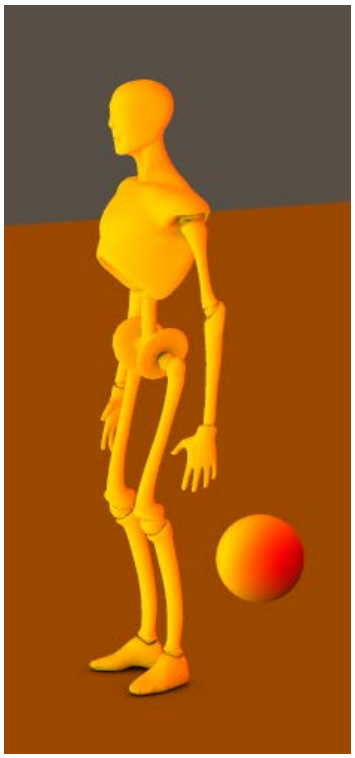


One (recent) way is to enable InDirect Lighting (IDL) and to encapsulate the entire scene by a huge Sky Dome object, with an image mapped onto it, and having its surface normal pointing inward. This method uses extensive raytracing, and is resource intensive (memory, CPU power, render time, user patience). The dome sends its light rays inward into the scene, and from there the light gets bounced all around.

Another way is to put an Image Based Light (IBL) in the scene. An IBL is a kind of point light, with an image (frontally) mapped onto it, but with its light rays traveling inward towards the lamp position. What does that mean? Well,



makes



So the image is wrapped around the point light, and the bright yellow midsection is related to the mid-frontal part of the light. The red edges of the image all fold onto the mid-back side of the light, and all areas top side and bottom get the orange taint. That's: frontally mapped.

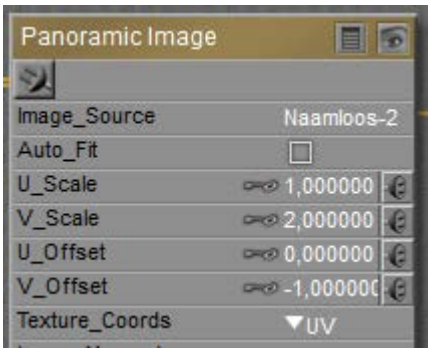
See http://www.book.artbeeweb.nl/?p=3090 for details on translating the environment to IBL images.

In the image above, the IBL light resides at 0,0,0 between Andy's feet. And the light rays are coming in, so Andy is colored yellow at its front, red at its back and orange at his top, sides and bottom.

Tip: for IBL lights, shadows should be OFF. As those lights cannot be placed but can be rotated (directed) only, and because lightrays are considered coming in towards that point of interest, shadows from IBL lights tend to get somewhat unnatural, to say the least.

The ball next to Andy uses the probeLight image mapping (see 47 WHAT'S THE DIFFUSE > PROBELIGHT NODE USED FOR? for details) which offers the same mapping and image as used for the IBL, so the IBL effect is exaggerated in this scene, to illustrate the effects of mapping and of the lighting direction. As far as mapping is concerned, probeLight works well on a basic Poser Ball, and just offers some extras for tuning image appearance and some squeezing and stretching compared to just plugging in the image_map itself. The Hires ball however turns out quite hard to orient properly which becomes apparent when an image is applied to it.

As IBL serves either as an extra or as a replacement of an IDL sky dome, the remaining question is: can the IBL-image be mapped onto such a dome. Yes it can, and it works immediately on the geosphere and hemisphere from the regular Poser content, and on Bagginsbill's EnvSphere and EnvHemisphere. Bagginsbill's EnvDome offers a slight shift in mapping, which has to be compensated for.
(V_Scale is doubled, and V_Offset is set to -1).



98 What's the fuzz about Gamma Correction?

Although various configurations can be setup via Render Settings (Poser 10, Pro 2010 and up), this topic is considered Intermediate – and sometimes Advanced – level.

Intermediate

When it comes down to Gamma Correction, there are a lot of stories. About gamma, images and devices, and on the application of the gamma function in Poser rendering. Some of those are close connected, other stories are hardly related. Unfortunately, all stories tend to get mixed up in forum threads, manuals and alike. Let's tell them all, in a sort of organized way.

Gamma and Devices

Output devices (at the end of any workflow, step A in the scheme above), like TV sets, PC monitors as well as newspaper pages all have in common that when they display an image, the midtones are darkened, the shadows lose detail while the brights gain some. From electronics, and from image handling as well, this loss of brightness is known as "gamma distortion" of "gamma effect". Lots of people find this reduction of image brightness rather unpleasant for various reasons, some of which even go back to our natural roots having hunters and prey hidden in the darks.

A way to overcome this is to adjust the image before it gets displayed, by an "inverse gamma" or "gamma correction" effect (start of the workflow, step B in the scheme above). This brightens the midtones, and enhances the details in the darks at the cost of losing some in the brights. When such a corrected image is displayed, the "gamma inversion" embedded into the image and the "gamma distortion" of the device cancel each other out, and the image is displayed as it was intended, and as we are pleased to see it. This is how JPG (for pictures) and MPG (for movie footage) came to life: these formats have the gamma correction embedded as prescribed by their format standard; such in contrast to BMP, and raw image data. And so, this way of work proves fine for newspaper and magazine printers as well as TV sets and movie projectors which don't have the processing power ('intelligence') on board to adjust images or movie frames on the fly.

TV sets etc. have to work with images and frames that have a standardized correction embedded into them. But they only approach that standardized gamma effect themselves. So they only can display the images as being "not too far off". PC's and the like have an advantage. They can open the image and correct for the standardized correction before (or when) sending the image data to any user application. This is step D in the scheme, called linearization.

And then the application can send its results to the video driver / video card which then communicate with the monitor. This offers the opportunity to establish correction curves for a specific monitor. Either by loading a brand and type specific profile from the manufacturer or even by measuring the monitors input/output response periodically, to compensate for aging as well. On a multi-monitor system, each one even can have its own profile, so an image keeps its exact appearance when dragged from one device to another. This is step C in the scheme, usually referred to as Color Management.

Summarizing: given the behavior of my monitor (A), image formats (B) are made up to compensate and with a dumb device in the middle, both cancel out to give me the proper image in result. With an

intelligent device in the middle, both actions can be compensated for (monitor A by driver C, image format B by image handler D) so also the application in the middle can make proper results based on proper ('linear') inputs, and without having to worry about image formats and device characteristics.

So, in my PC, **if** the operating system / application deals with image formats and **if** the driver/card deals with monitor profiles, every type of image will behave 'pure' from an application input/output point of view. Great!
Unfortunately, there are two ifs in this.

On the output side, PC video driver/cards have the profile handling capacity for say 10 years now (from 2005 on), but native (motherboard) video and laptops can only do so for say 5 years (2010 and on), while smartphones and tablets hardly do, and whether my laptop is feeding gamma-corrected streams to my TV-set is quite questionable. So images might look different from my PC monitor.

In cases the output handling abilities fall short, some applications offer the adjustment themselves, while others don't. Some applications put in some adjustment anyway. In either case, a single image is shown in different ways by different applications onto the same screen.
Unfortunately, there is no way to tell which one is right, although Photoshop can come close once I master its Color Management settings. And there are not that many ways for user-adjustment.
Poser creates render results in 16-bit per color EXR format (see 67 How does Poser handle my images for texturing?, and does quite a nice job in showing it on screen. Anyway, there are no settings or dials to affect that.

On the input side, some versions of operating systems do offer routines for proper image handling, some don't. Even if they do, it's up to the application to call for them instead of reading out the image data itself. I don't have to figure out the details of this, it's easy to test: make an image, save it in bmp as well as jpg format, and open both in my application at hand (with the same settings etc.). Do both show the same result? Then all is fine. If not, the jpg needs its embedded gamma-correction taken out, or has so but not to the right amount yet. Sometimes I can correct for that, either by opening it with adjusted settings, or by correcting for it in pre-production, with my image editor. This is called: **linearizing the workflow**. Poser (7 and up) users can rest assured: an image will show the same way whatever format is used.

Some test results, with a BMP and a JPG with an embedded profile:

- In Poser (Pro 2014 SR3.1), Photoshop (CC 14.1), Internet Explorer (11.0) and Windows (7) Photo Viewer both BMP and JPG are the same.
- In Chrome (33.0) and FireFox (27.0), the JPG is slightly darker than the BMP. So these applications seem to take out the gamma correction as well, but take more out than the profile put in in the first place.

Gamma and Poser

Overview

All in all, Poser works fine with all image formats and all output devices. I do not need any adjustments for that, Poser offers a "linear workflow", especially in the design stage. But the great story is: Poser offers an optional non-linear rendering as well.

In that, it first darkens all images, footage and colors around (step E in the scheme), and then it renders (step R). This will create quite a dark render, including shading and shadows. And with dark highlights as well, with quite a reduced risk on overlighting. Then it un-darkens the render result, step F in the scheme. From that, I can export the image for any further handling as required.

This sandwiched rendering will make all surfaces reappear at about their original colors and brightnesses as steps E and F are intended to cancel out. But all darkening that occurred during the rendering itself, like the shading and shadowing, will be brightened up considerably as these receive the post-render treatment from step F only. Highlights will be brightened a bit as well, but not to an overlighting level. In short: this sandwich process of darkening first and un-darkening later will just soften the darks and highlights, contributing to a more pleasant and more realistically looking image.

So, how to do the (un)darkening? Poser offers the gamma function for this, and that was just an arbitrary choice. They could have used another function as well, like exposure, but they didn't. That gamma function is driven by a single ("gamma") parameter. Each value larger than 1 will soften shadows and highlights, but values over 4 will create very artificially looking results. Values around 2 do very well for photorealistic renders. Values smaller than 1 (and preferably larger than 0.25) will have the opposite effect, which might be desired in comic styles.

Note on the method:

> This Gamma-sandwiching approach is one of the available strategies. It works about as required for shadows as well as for highlights, and it's easy to implement as it just adds a pre-render and a post-render step but leaves the rendering itself as is. Other 3D software (like Vue) works likewise. Very recommended, discussed in various sections of this tutorial and in 98 WHAT'S THE FUZZ ABOUT GAMMA CORRECTION? and 99 GAMMA, TRANSPARENCY AND BLENDS specifically.

Other ways are:

- Tone mapping. Not available in Poser (yet) but available in external renderers like LuxRender. That way the internal High Definition (HDRI) render result is mapped to the most feasible visual range. It corrects for (too) low lighting levels too, but brighter lights will make darker shadows, so although this method has far less artefacts, shadows are not catered for. PoserPro users can do it manually, by exporting the HDRI result first. Poser (all versions) can adjust Exposure as well, as a substitute.
- Screening. Available in Photoshop and alike, and considered the industry standard in combining the effects of multiple light sources (in post-production). Shadows stay as dark as they were, however, and need additional (manual) adjustment.
  PoserPro users can do it manually, by exporting the result as a layered PSD file. This is the recommended way of work when integration Poser renders into a larger, professional workflow. Gamma Correction (as well as Exposure correction) should be OFF then.

Note on the Gamma value:

> *The Gamma value applied in Poser is a trade-off. For scenes with mainly or only direct lighting, a value below but close to 2.0 will produce the best shadow softening but when IDL gets the upper hand, lower values (towards 1.0) will produce better results. This is because the Gamma Correction includes compensation for radiosity of ceilings, walls etc. and with IDL, the lighting itself is catering for that itself. On the other hand, adding up very dissimilar light sources / surface components works best with gamma values just below 2.0 while adding up quite similar lights or about equally bright components works best with values even over 3.0. But… there is only one gamma to serve all. As a result, the industry standard 2.2 for device*

handling is considered a reasonable trade-off for Poser as well, even though both are completely unrelated.

Gamma and Confusion

So there are two rather unrelated processes. One for putting images onto output devices and for taking their effects out to enable applications to serve normally (represented by steps A, B, C and D in the scheme). And one for softening shadows and highlights in Poser renders to establish a more (photo)realistic look and feel, represented by steps E and F in the scheme.

The first utilizes a "gamma function" with a parameter value around 2 as this represents most output devices quite reasonably. In formal image format standards the value 2.2 is prescribed.
The second utilizes the same gamma function, and a parameter value around 2 happens to present a pleasant result, and supports photorealism quite well.

So both processes happen to have the same mathematical function and the same optimal input value for it, in common. That's their only relationship. Hence, don't bother about all gamma-fuzz in forums, google, Wikipedia and the like, and don't bother about monitors, calibration, and other output devices. They address the steps A, B, C and D from the scheme quite well.
But I can freely choose whether or not to use the Gamma process in my Poser rendering, and I can freely choose my gamma value with it, as steps E and F in the scheme are separate ones. However, for photoreal'ish results, using the Gamma process with the default 2.2 value is much recommended.

Poser Details

The overview above mentioned the benefits of using the Gamma Mechanism offered from Poser 10 and Poser Pro 2010 up. The mechanism however has some drawbacks as well:

- Greyscale swatches and images which for bump/displacement should be excluded from the mechanism. For images, their gamma should be set to 1.0 explicitly in the Texture Manager. When this can't be done for whatever reason the Gamma node can be used for additional adjustment. See 84 WHAT ARE THE HSV AND GAMMA (MATH) NODES USED FOR? on details.
- The mechanism is applied to all images and color swatches, but not to numericals. So 50% white at 100% value, and 100% white at 50% value do refer to the same shade of grey but will end up differently in the render result when the Gamma Mechanism is applied.
  For that reason, the various Value slots in PoserSurface should either be 0 or 1, but any intermediate value should be addressed in the accompanying Color slot. At least when Gamma Correction is used for rendering, although it doesn't harm to turn this into a generic way of work.
- When using the HSV node (see 84 WHAT ARE THE HSV AND GAMMA (MATH) NODES USED FOR?, be aware that the image inputted into it will get Gamma corrected **before** the HSV adjustments are applied. You might have to adjust the Texture Manager settings for these.
- Since values are not adjusted by the Gamma mechanism before rendering, the User_Defined node (see 83 WHAT ARE THE USER_DEFINED, SIMPLE_COLOR AND COLORRAMP NODES USED FOR?) which produces a color on value inputs will not be affected. You might want to add the Gamma node itself for explicit adjustment.
- Various (older) object textures offer a neat balance of color settings and adjustments, which work out nice when used in Poser 9 and earlier. But these produce awful results, when rendered in more recent Poser versions that have the Gamma mechanism enabled. All color settings etc. are distorted, are any balance is completely gone. Those textures have to be re-evaluated (or the mechanism should be turned OFF for those renders).

A well-known example is the "smoking zombie" look of the V4 character when rendered with Gamma Correction switched ON. See 99 GAMMA, TRANSPARENCY AND BLENDS for details.

- The various blend-nodes, like Edge_Blend, Fresnel_Blend and the Fresnel node itself, but also the Blend option in Color_Math and the ColorRamp node as well present some deviations when rendering under Gamma. Like the previous point, not only brightnesses but also the resulting hues can be affected, reflection-to-refraction rations might get distorted, and the like.
  For short: they should participate in the Gamma mechanism as least as possible, and bypass it as much as possible. See 99 GAMMA, TRANSPARENCY AND BLENDS for details.
- Poser Transparency acts as some kind of 'super blender', and may cause a blend of light bouncing from the front of the object (reflection, diffusion), light bouncing from the backside of the object (internal reflections), light from behind the object passing through , and in the meantime it's casting shadows onto the scene behind the object too. Should transparency itself participate in the Gamma mechanism (like Diffuse etc.), or should it bypass it (like Bump/Displacement?). For short: it should participate as least as possible, and bypass as much as possible. See 99 GAMMA, TRANSPARENCY AND BLENDS for details.

For Poser 9 and earlier, the Gamma node is available as well. This is meant for those cases where Poser is not taking any embedded brightness corrections out of the image properly. The node cannot be used to build the Gamma Mechanism myself, as that is a sandwiching process of which the nodes can only affect the input-part. One can however do the output-part of it in post. It's quite elaborate though, to add the gamma node for every image map and color swatch in all material definitions in the entire scene.

The math of Gamma

Okay, let's go for it. This is the Advanced part.

Shadows

Say, an object in the scene has a color D. That is, I'm not interested in its hue, but the brightness is its Diffuse_Color or D between 0..100%. Let's pick 70%.

With the Gamma mechanism enabled, the pre-render pass will turn that into D^g and for g>1, by default g=2.2, that will result in a darkening as for instance $0.7^{2.2} = 0.46$.
When rendering, shading (like Lambert as in 30 CAN I GET A (SIMPLIFIED) EXPLANATION ON LAMBERT AND DIFFUSE SHADING?) darkens the sides of the object with respect to the lights, and any other object standing in the light casts shadows onto my object as well. This shading and shadowing actually multiplies areas of the object with S, between 0..100% also, say 20% at specific points. That turns the brightness of those spots to $S*D^g$ or 0.2*0.46=0.092 in the example.
The post-render pass then will turn the final result into $(result)^{1/g}$ which is a brightening, from the formula $(S*D^g)^{1/g} = S^{1/g} *D$. The spots on the object without shading and shadow result in $0.46^{0.45}$=0.70 while the spots with it result in $0.092^{0.45}$=0.34.
Without the Gamma passes, the spots with shading would have resulted in 70%*20%=0.14 – a much darker result.

So from the formula and the example figures I learn that the object color will go untouched (70% remains 70%) while the shading and shadowing gets brightened (34% instead of 14% in the result, which is like multiplying with 0.485 ($=0.20^{0.45}$) instead of 0.20).

Colors, Images and Values

Now let's look at the result of a diffuse component, with color D, an image map M attached to it adjusted with "render gamma", and with value d, having an image map m attached to it adjusted with a custom gamma value h (1 or anything else).
Then the final result will be (S* Dg * M^g * d * m^h)1/g = $S^{1/g}$ * D * M * $d^{1/g}$ * $m^{h/g}$ as the shading and shadowing S will get brightened up as required, the value d went untouched by the pre-render pass and hence will appear brightened up in the result as well, and the image m is assigned its own adjustment.
So the total effect is different from just having a brightened up shadow on an unaffected object. The color swatch D and its image map M went unharmed, the result for the other image map is my own choice, but the value part results in a brightening up as it was ignored in the pre-render pass.
Hence, when rendering with Gamma ON, I'd better set all value parts in PoserSurface to either 0 or 1, or face an additional brightening. This also implies that when I do want a brightness reduction, I have to blend that into the color swatch.
For instance, a metal reflects with an orange color RGB (100%, 50%, 0) and reflectivity 80%. Then I should not set the color to the Reflection_Color and the 80% to the Reflection_Value, but I should assign the overall RGB (80%, 40%, 0) to the color swatch instead and set the Value to 1.0. Otherwise, my reflections will be too bright when rendering with Gamma. And without Gamma, it doesn't matter.

Most components of PoserSurface have a color swatch. Alternate_Diffuse and Alternate_Specular even don't have a value slot. Bump and Displacement do have a Value only, but anything plugged in those should not have any Gamma mechanism assigned. Bump maps for instance should have their Gamma = 1.0 in Texture Manager.

The only components requiring explicit attention are the 'blenders', like Edge_Blend, Fresnel, but Transparency too. These components blend colors, images, scene reflections and the like based on a value or image map. Should they participate in the Gamma mechanism (like Diffuse etc.), or should they bypass it (like Bump and Displacement?).
For short: they should participate as least as possible, and bypass as much as possible. See 99 GAMMA, TRANSPARENCY AND BLENDS for details.

Combining Components

So far so good, for components considered separately, and for the effects on shading and shadows. But what happens when components are combined and added up? That's like two projectors, each presenting its own image on the wall.

In nature, our eyes will adjust to the increased lighting levels. In image handling such a thing is not possible, and for ages programs like Photoshop blend layers using the Screening method, which mimics the behavior of photographic film or its modern electronic equivalents.
Film behaves about as follows: when a ray hits the film, there is a possibility, a change p_1, that the film responds to it, and so there is change 1-p_1 that the film remains unlit. When a second ray, either from a later moment, or from another source, or from opening the diaphragm further, hits the same spot on the film, there again is a change p_2 of lighting the film, provided that it was unlit before that. So the change that the spot on the film is still unlit after two rays is (1-p_1)*(1-p_2) and hence the formula for light addition is: 1-(1-p_1)(1-p_2).

This is the way Screening works on PSD image layers: the inverses of the layers are multiplied and inverted again. For instance: one layer represents 30% brightness and the other 80%, then it results

in 1- (1-0.3)*(1-0.8)=86% brightness. So lighting levels will increase but will not exceed 100%. Screening is considered the (industrial standard) way of work for adding light to an existing image (as multiplication is considered the right way to apply shadows and the like).

In Poser, without the Gamma mechanism, components will just be added up. In the above case, 30%+80%=110% and overlighting artefacts will ruin the render result. With the Gamma mechanism however, it will brighten the addition of the darkened components, that is: $(0.3^{2.2}+0.8^{2.2})^{0.45}=0.84$.

So, with Gamma about 2.2, the Gamma mechanism not only prevents overlighting and dims the highlights when specularity is added to regular diffuse. It also approximates the industry standard screening method very well, and should be considered the way forward to combine any components in the render result. Whether they are diffuse and specular, diffuse and reflection, reflection and refraction, you name it. So they all have to respond to the Gamma mechanism to enable that. With exceptions for Bump and Displacement, which simulate or realize an alteration of the object geometry and do not “add light” or “subtract shadow” to the resulting render in any way. And with exceptions for the ‘blenders’ (Edge_Blend, Fresnel, Transparency, …) which also should avoid the Gamma mechanism as much as possible.

99 Gamma, Transparency and Blends

In the previous section 98 WHAT'S THE FUZZ ABOUT GAMMA CORRECTION? it's discussed how the Poser Gamma mechanism not only brightens up shadows and shades to compensate for the lack of radiosity and atmospheric scattering as can be found in real-life scenes, but also helps to add up lighting components like diffuse, specular, reflection and the like, especially when such components can be considered rather independent.

In some cases however, such components come to us in a '' blend', that is: more of one implies less of the other, which makes them far from independent. For example:

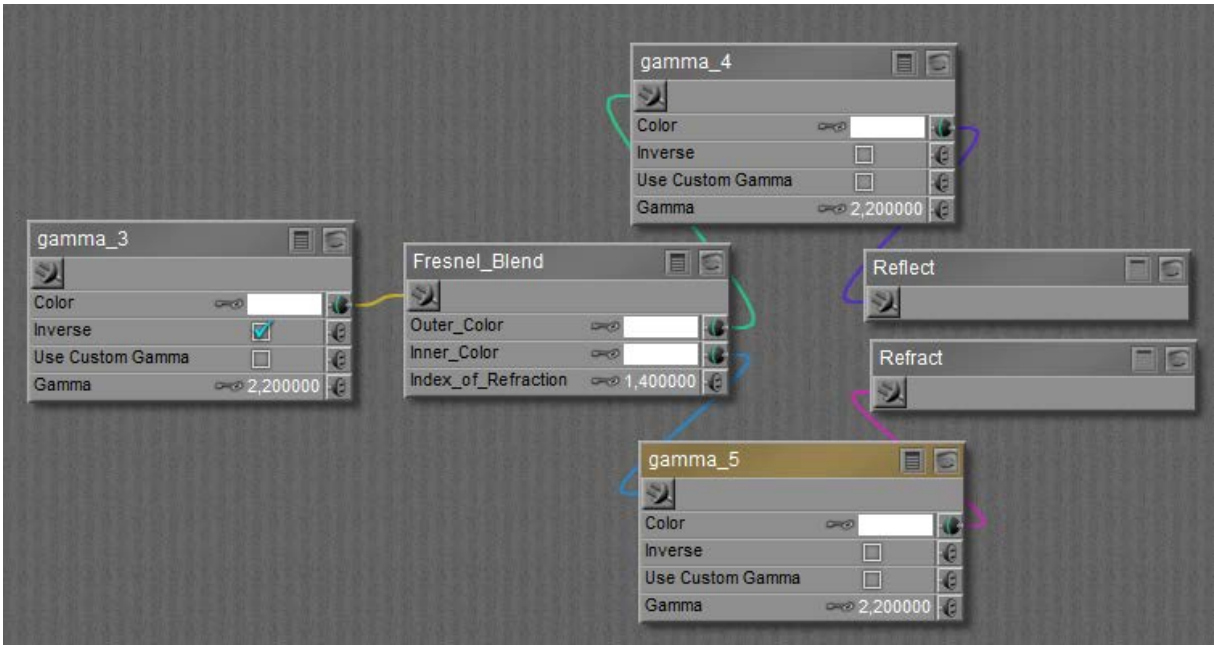
- I use a blending node which, from a value or image brightness, blends x% of one input with (as a consequence) 1-x% of the other input.
- I apply Transparency, which blends reflections and diffuse from the front of the object with the scene behind the object shining through. The more of the first, the less of the latter.
- By manual action: I take some red out of Diffuse and put that amount of red back in via Ambient.

In those cases, the Gamma mechanism turns into a burden, instead of a benefit, when it includes the blending itself. I went through all the math (see last part of this chapter), and through a load of test renders, and my findings are:

- If possible, the best solution is to revert the Gamma mechanism for the blend. That is: get the Gamma out of all inputs including the blending, apply the blend, and put the Gamma back into the result. This will apply the proper blending to the proper colors, so to say. This works for nodes.
- Else, accept the Gamma mechanism for the inputs but avoid any Gamma effects on the blending itself. This will distort the proportions that the inputs contribute to the result, and always brightens up the object somewhat compared to the previous way of work. It applies the proper blending to the adjusted colors. This works for Transparency.
- Else, accept the Gamma mechanism but compensate for it where possible. This can be tedious but might solve the remaining cases, like the blending invoked manually.
- As said, having Gamma to affect the blending itself is to be avoided, as it applies an adjusted blending to adjusted colors. The results will be off again, but far more (as least twice as much) as in the method mentioned above, and in either direction: too dark or too bright, depending on the lighting balances in the scene itself.

Reverting the Gamma Sandwich

Just like this:



The Fresnel_Blend is packed between two Gamma's at the input side to take the effect out of the Reflect and Refract, and an inverse Gamma at the output side to put the effect back in again. Note that the Gamma node is not applying Gamma (darkening), but its inverse: Gamma Correction (brightening).

This can be applied to all kinds of blending nodes: Fresnel_Blend, Edge_Blend, Color_Math (Blend), ColorRamp, you name it. It cannot be applied to the Fresnel node itself, as one cannot take the Gamma out of the reflections / refractions before they get blended.

Just leave Gamma out

The second best option, it distorts but not that much. Just apply the mentioned nodes as they come, apply transparency as it comes, it feels like "doing nothing". But it also means: use Values for blending amounts and transparency, avoid using Color-swatches at any cost, and when using image-maps do set their Gamma values to 1.0 to bypass the mechanism.

This is the method followed automatically when the Fresnel node is used.

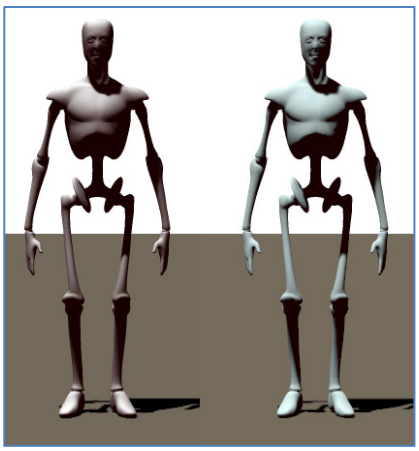
The use of Gamma = 1.0 for Transparency in all PoserSurface definitions in the scene can be enforced with the changeGamma scripts, discussed in 66 HOW CAN I ASSIGN AN IMAGE TO A MATERIAL?.

Manual adjustment

Older textures for characters offered a way to mimic advanced skin effects, like translucency and scattering. A small portion of red was removed from Diffuse (turning white into (245,255,255) a very mild cyan) and put back in again in the Ambient slot (turning black into (10,0,0) a very deep red). When rendering without Gamma, as was common in those days (Poser before version 10) this resulted in white (multiplied by any image map) plus a mild reddish glow.

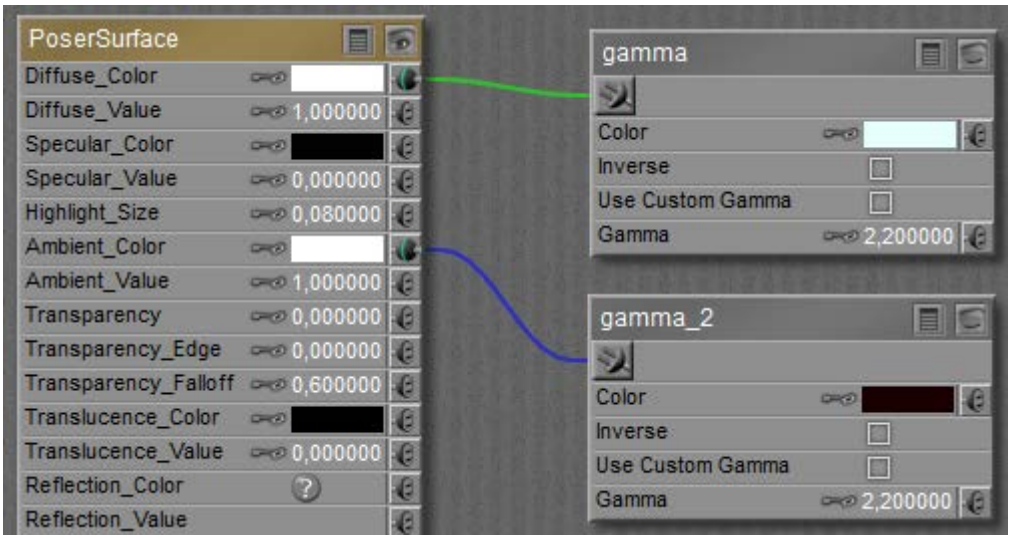
However, when Gamma is turned on, the ratio gets distorted, the brightest component – the cyan in here – gets the upper hand, and all characters got the "smoking zombie look".

Like this, left Andy without Gamma, right Andy with Gamma, same material and lighting:



Solutions:

- Rebuild the material completely and switch to modern alternatives applying scattering and the like. Set the Diffuse slot to white and remove all Ambient from the PoserSurface definition. This however will not work for early Poser versions not supporting those modern features.
- Set the color swatches apart, and get the Gamma out:



  It will, by the way, increase the reddish glow so some additional adjustment might be required. This is the most flexible solution, leaving all other settings intact, and works in all Poser versions for all Gamma settings, including none at all.
- Correct the swatches in PoserSurface manually. Instead of taking out 5% red, take out just 1%, and instead of adding 5% Ambient, add 10%. These adjustments however will vary with different settings for the Render Gamma. It leaves the structure of all PoserSurface definitions intact, but it's the least flexible and most tedious of all.

A closer look at Transparency

Consider an object, having a (surface) transparency of T=90%. When a direct (say 100%, white, point) light is illuminating the scene, this object will show the following behavior:

- The light is passing through a surface twice, object-front and back, and so it produces a shadow of intensity T^2 (90%x90%=81%) assuming Gamma is not applied to the transparency itself. Then this shadow is brightened up in the post-render pass, to $T^{2/g}$ ($=0.81^{0.45}$ = 91%) like all other shadows in the scene.
  When Gamma is applied to transparency too, it gets reduced to T^g ($0.9^{2.2}$= 79%) in the pre-render pass, then it casts shadows T^{2g}, and then these get brightened to $T^{2g/g}=T^2$=81%. In other words: when I include transparency into the Gamma mechanism, the shadows from the object will be darker related to other shadows in the scene, as if the Gamma mechanism was not applied but to those shadows specifically.
- However, the scene behind the object shining through will follow the same logic. When no Gamma is applied to transparency, that scene will get dimmed to 81% and brightened up in the post-render pass to 91%, while applying Gamma to transparency will produce a neat 81% result.

So at first sight I've got a choice: either the object is too bright and the shadows are right (no Gamma to transparency), or the object is fine and the shadows are too dark (Gamma on transparency too). At second sight, more effects have to be considered:

- Since the object is only partially transparent, a portion of the (light of the) scene in front of the object will bounce as well, as diffuse or reflection. At a surface transparency of 90%, this will put 10% into the mix. When Gamma is allowed to reduce the transparency (from 90% to 79% as calculated above) this frontal portion will increase to 100-79=21%.
- When the surface transparency is 90%, then that amount of light will pass the frontal surface (10% bounces as discussed above), and at the backend surface again 90% of that will pass, and 10% will bounce. The latter implies that 10% of 90% = 9% starts to participate in an internal bouncing around, and each time a surface is hit some light passes through and some bounces again.
  When Gamma is allowed to reduce the transparency, more bouncing will occur and actually the 9% mentioned will rise to 17%.
- That light bouncing around will not only leave the object at the front, a part of it leaves the object at the back instead. But… from the light from the scene behind the object, some portion bounces around internally as well, leaving the object at the front, and to some extent at the back instead.

Actually, all the four elements mentioned above (all except the shadow mentioned first) add up to make one result. So how can be predict what happens when a gamma sandwich is applied to all that? We can be sure something gets distorted, as it would be a miracle when all things turned out equal. I do need math, figures and a lot of tests to get it sorted.

That's for the next part of this chapter. For short: the Gamma mechanism, even when not applied to transparency itself, will produce a somewhat distorted result as it alters the balance between the 'front' and 'behind' scenes in the result. But when transparency itself is excluded from Gamma, then 1) I'll get the best shadows which behave in sync with the other shadows in the scene and 2) I'll get the least (and eventually most manageable) distortion, compared to when Gamma is included into the transparency.

So, I consider it best to treat Transparency as a value 'thing', I'll avoid using color swatches to determine it, any image map involved will get Gamma=1.0, and I'll use the various scripts to enforce that all over my scene materials.

The math and the figures

For those who want to check out on my findings themselves.

Plain Blending

Consider two inputs, A and B, blended together in an x-to-(1-x) ratio. More of A implies less of B and vice versa. For instance, reflection blended with diffuse (a shiny surface), of with refraction (Fresnel).

- I without gamma: $A*x + B*(1-x)$
- II with Gamma on A and B only: $\{ A^g*x + B^g*(1-x) \}^{1/g}$
- III with Gamma on all: $\{ A^g*x^g + B^g*(1-x^g) \}^{1/g}$

Some results, for Gamma 2.2 and blend ratio x set to 70%:

A	B		I	II	III
80%	**80%**		80%	80%	80%
80%	**60%**		74%	75%	70%
80%	**40%**		68%	71%	62%
80%	**20%**		62%	69%	57%
60%	**80%**		66%	67%	72%
60%	**60%**		60%	60%	60%
60%	**40%**		54%	55%	50%
60%	**20%**		48%	52%	44%
40%	**80%**		52%	56%	65%
40%	**60%**		46%	47%	52%
40%	**40%**		40%	40%	40%
40%	**20%**		34%	35%	31%
20%	**80%**		38%	49%	62%
20%	**60%**		32%	38%	47%
20%	**40%**		26%	28%	33%
20%	**20%**		20%	20%	20%

Transparency

Consider A the lighting level in front of the transparent object, B behind it, for transparency T. More transparency means more of B (transmission/refraction), less transparency means more of A (diffuse/reflection). Transparency induces internal reflection/diffusion within the object. For a simple illustration: at transparency 90%, 10% with be reflected/diffused from the front while 90%x90%=81% shines through, passing front and back surfaces. The remaining 100-10-81=9% results from internal reflection / diffusion, leaving the object while bouncing around.

- I without Gamma: A*(1-T)*(1+T/(2-T)) + B*T*(T+(1-T)^2/(2-T))
- II with G on A and B only: $\{ A^{g}*(1-T)*(1+T/(2-T)) + B^{g}*T*(T+(1-T)^{2}/(2-T)) \}^{1/g}$
- III with G on all: $\{ A^{g}*(1-T^{g})*(1+T^{g}/(2-T^{g})) + B^{g}*T^{g}*(T^{g}+(1-T^{g})^{2}/(2-T^{g})) \}^{1/g}$

Some results, for Gamma 2.2 and Transparency set to 70%:

A	B		I	II	III
80%	80%		80%	80%	80%
60%	80%		71%	72%	67%
40%	80%		62%	65%	56%
20%	80%		52%	61%	48%
80%	60%		69%	70%	75%
60%	60%		60%	60%	60%
40%	60%		51%	52%	47%
20%	60%		42%	47%	38%
80%	40%		58%	62%	71%
60%	40%		49%	50%	55%
40%	40%		40%	40%	40%
20%	40%		31%	33%	28%
80%	20%		48%	58%	69%
60%	20%		38%	44%	52%
40%	20%		29%	31%	35%
20%	20%		20%	20%	20%

Conclusions:

- When A and B are about equivalent, the method does not really matter, and distortions from Gamma Correction are minimum in any way. This situation occurs for instance when a semi-transparent object resides in a quite homogeneously (say IDL) lit environment.
- In all cases, the difference between method III (all gamma) and method I (no gamma) is at least at large, and in varying directions, compared to the difference between method II (no gamma on blending /transparency) and method I. Therefore, I prefer method II over method III as presenting the least distortions, when method I is not available.
- So in case of blending / transparency, I prefer values, I avoid color swatches, and when applying images I set their Gamma to 1.0.

Appendix: Poser nodes / settings and versions

This appendix presents lists the Material Room nodes as well as various Render Settings, and their availability in recent Poser versions.

Nodes

Group	Node	In P7	In P8/Pro2010	In P9/Pro2012	In P10/Pro2014
Math	Blender	Yes	Yes	Yes	Yes
	Edge_Blend	Yes	Yes	Yes	Yes
	Fresnel_Blend	No	No	Yes	Yes
	Component	Yes	Yes	Yes	Yes
	Math_Functions	Yes	Yes	Yes	Yes
	Color_Math	Yes	Yes	Yes	Yes
	User_Defined	Yes	Yes	Yes	Yes
	Simple_Color	Yes	Yes	Yes	Yes
	Colorramp	Yes	Yes	Yes	Yes
	HSV	Yes	Yes	Yes	Yes
	Gamma	No	Yes	Yes	Yes
Lighting \Specular	Anisotropic	Yes	Yes	Yes	Yes
	Phong	Yes	Yes	Yes	Yes
	Glossy	Yes	Yes	Yes	Yes
	Blinn	Yes	Yes	Yes	Yes
	Specular	Yes	Yes	Yes	Yes
	Ks_Microfacet	No	No	Yes	Yes
Lighting \ Diffuse	Clay	Yes	Yes	Yes	Yes
	Diffuse	Yes	Yes	Yes	Yes
	ProbeLight	Yes	Yes	Yes	Yes
	Toon	Yes	Yes	Yes	Yes
Lighting \ Special	Skin	Yes	Yes	Yes	Yes
	Subsurface Skin	No	No	Yes	Yes
	Velvet	Yes	Yes	Yes	Yes
	Hair	Yes	Yes	Yes	Yes
	Scatter	No	No	Yes	Yes
	Custom Scatter	No	No	Yes	Yes
	Fastscatter	Yes	Yes	Yes	Yes
Lighting \ Raytrace	Reflect	Yes	Yes	Yes	Yes
	Refract	Yes	Yes	Yes	Yes
	Ambient Occlusion	Yes	Yes	Yes	Yes
	Gather	Yes	Yes	Yes	Yes
	Fresnel	Yes	Yes	Yes	Yes
Lighting \ Env Map	Sphere Map	Yes	Yes	Yes	Yes
Variables	N, P	Yes	Yes	Yes	Yes
	Framenumber	Yes	Yes	Yes	Yes
	U,V	Yes	Yes	Yes	Yes
	Du, Dv	Yes	Yes	Yes	Yes
	dPdu, dPdv	Yes	Yes	Yes	Yes
	dNdu, dNdv	Yes	Yes	Yes	Yes

Group	Node	In P7	In P8/Pro2010	In P9/Pro2012	In P10/Pro2014
3D textures	Fractal Sum	Yes	Yes	Yes	Yes
	fBm	Yes	Yes	Yes	Yes
	Turbulence	Yes	Yes	Yes	Yes
	Noise	Yes	Yes	Yes	Yes
	Cellular	Yes	Yes	Yes	Yes
	Clouds	Yes	Yes	Yes	Yes
	Spots	Yes	Yes	Yes	Yes
	Marble	Yes	Yes	Yes	Yes
	Granite	Yes	Yes	Yes	Yes
	Wood	Yes	Yes	Yes	Yes
	Wave3D	Yes	Yes	Yes	Yes
2D textures	Wave 2D	Yes	Yes	Yes	Yes
	Image Map	Yes	Yes	Yes	Yes
	Brick	Yes	Yes	Yes	Yes
	Tile	Yes	Yes	Yes	Yes
	Weave	Yes	Yes	Yes	Yes
	Movie	Yes	Yes	Yes	Yes
Compound	Compound	No	No	No	Yes

Render Settings

	In P7	In P8/Pro2010	In P9/Pro2012	In P10/Pro2014
Subsurface Scattering	No	No	Yes	Yes
Raytracing	Yes	Yes	Yes	Yes
Irradiance Caching	Yes	Yes	Yes	Yes
Indirect Light	No	Yes	Yes	Yes
Post filter	Yes	Yes	Yes	Yes
Tone Mapping	No	Yes	Yes	Yes
HDRI opt output	No	Pro only	Pro only	Pro only
Gamma Correction	No	Pro only	Pro only	Yes
Aux render data	No	Pro only	Pro only	Pro only