

Presence and Opacity

In RenderMan, you can control the opacity of an object using the Presence Parameter on the materials. Typically designed to work as a 0 or 1 "cutout" effect, values in-between can be used for semi-transparent objects or ghost-like effects.





If using a value between 0 and 1 (or a pattern with middle gray values) you will notice noise by default since we use a stochastic sampling technique. However, integrators allow for smoother effects by specifying a depth in the Stochastic Opacity Depth parameter. Where each intersection counts as a depth limit. This will be explained below.

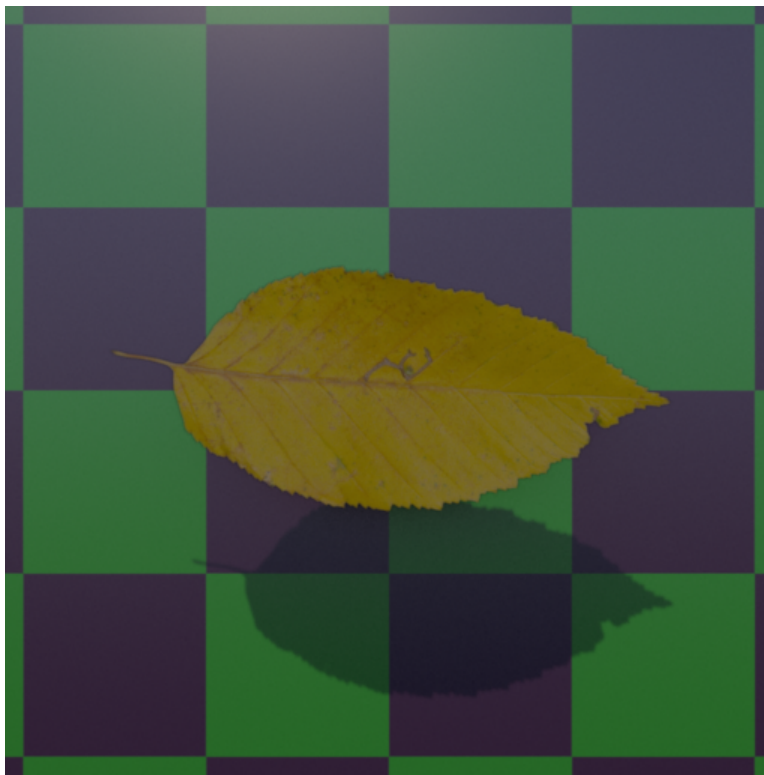
If rendering thick glass, presence is ignored as we need to know when we enter and exit a volume. A cutout here could cause issues with that condition.

The simplest example is that of a leaf. Rather than physically model a leaf which can be painful for the artist and increase geometric complexity in a scene, typically a plane is used to represent the leaf and a map is used to provide the shape. This map is then connected to the Presence Parameter.

Without using presence, you may end up with something like the image below.



This won't work, so we apply a map to the Presence parameter where the leaf area is 1.0 and the empty area we want to remove is 0.0. It's a simple black and white map without any greyscale values.



Notice several things:

- The shadow is now correct
- The area outside the leaf is now completely ignored by the renderer
- There's a small bit of darkening around the leaf, this is handled by bleeding the leaf color beyond the border of the *color* texture

This is the traditional way to use the parameter. But you may wish to use partial presence values in rendering for something semi-opaque. By default we use a stochastic method to determine presence. This means we randomly sample at varying levels of presence when the value is not a 0 or 1. This can generate noise and require more samples to clear up or converge. Below we have a set of overlapping planes set to a Presence value of 0.15. Notice that /



Some production integrators support a solution to this with a parameter called "Stochastic Opacity Depth". This value defaults to 0, meaning "off". However, at each increment, it tells the renderer to composite a semi-opaque color instead of sample randomly at that specified depth and no further. Below are examples at depth 1, 2, and 3



Depth 1, notice the areas of the plane where there is no stacking are now free of noise.



Depth 2, notice that planes overlapping one another once are now free from noise.



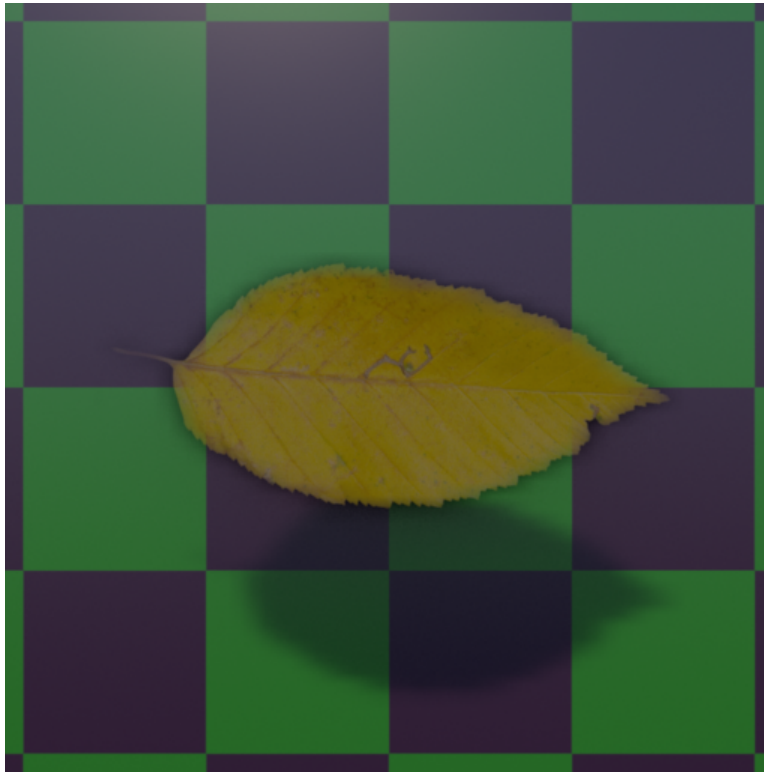
The render is now noise-free at depth 3.



As a performance consideration, you may notice that overlapping objects with high levels of opacity will create less noise. You may only need to set the Stochastic Opacity Depth to where the noise is agreeable relative to the rest of the scene instead of literally counting the overlapping surfaces and using that value as the setting.

Cached versus Uncached

To avoid some expensive work, we cache the presence value with the micropolygon of the geometry. Typically this is fine as we default to a dicing scheme that creates micropolygons the size of a pixel or smaller. However, you may notice that poorly tessellated objects have fringing or incorrect values.



Notice here the edges of the leaf are semi-transparent and incorrect. There are two ways to solve this:

- If using cached, decrease the micropolygon length to create smaller micropolygons and therefore more detail
- You may switch from cached to non-cached and instead evaluate with each shading sample, this could be expensive but it is "easiest" as a solution