10.2.3.3 Render - Blender Render Engine - Lighting - Shadows

Shadows

- Introduction
  - Lamps: Ray-traced Shadows
  - Lamps: Buffered Shadows
- Common Shadowing Lamps Options
  - See Also
- Lamps Raytraced Shadows
  - What is Quasi-Monte Carlo?

Introduction

Light wouldn’t even exist without its counterpart: shadows. Shadows are a darkening of a portion of an object because light is being partially or totally blocked from illuminating the object. They add contrast and volume to a scene; there is nearly no place in the real world without shadows, so to get realistic renders, you will need them. Blender supports the following kinds of shadows:

- Lamps: Ray-traced Shadows
- Lamps: Buffered Shadows
- *Ambient occlusion*
- *Indirect lighting*

Ambient occlusion really isn’t a shadow based on light *per se*, but based on geometry. However, it does mimic an effect where light is prevented from fully and uniformly illuminating an object, so it is mentioned here. Also, it is important to mention ambient lighting, since increasing *Ambient* decreases the effect of a shadow.

You can use a combination of ray-traced and buffer shadows to achieve different results. Even within ray-traced shadows, different lamps cast different patterns and intensities of shadow. Depending on how you arrange your lamps, one lamp may wipe out or override the shadow cast by another lamp.

Shadows is one of those trifectas in Blender, where multiple things have to be set up in different areas to get results:

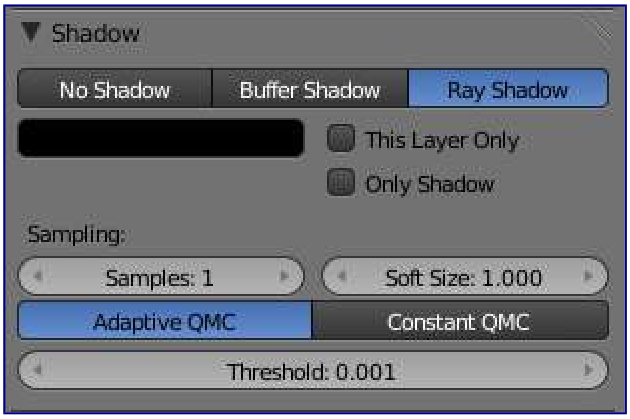
- The lamp has to cast shadows (ability and direction).

- An opaque object has to block light on its way (position and layer).
- Another object's material has to receive shadows (*Shadow* and *Receive Transparent* enabled).
- The render engine has to calculate shadows (*Shadow* for buffered shadows, *Shadow* and *Ray* for ray-traced shadows).

For example, the simple *Lamp*, *Area*, and *Sun* light has the ability to cast ray shadows, but not buffer shadows. The *Spot* light can cast both, whereas the *Hemi* light does not cast any. If a *Sun* lamp is pointing sideways, it will not cast a shadow from a sphere above a plane onto the plane, since the light is not traveling that way. All lamps able to cast shadows share some common options, described *here*.

Just to give you more shadow options (and further confuse the issue), lamps and materials can be set to respectively **only** cast and receive shadows, and not light the diffuse/specular aspects of the object. Also, render layers can turn on/off the shadow pass, and their output may or may not contain shadow information...

Lamps: Ray-traced Shadows



Ray Shadow enabled for a lamp

Ray-traced shadows produce very precise shadows with very low memory use, but at the cost of processing time. This type of shadowing is available to all lamp types except *Hemi*.

As opposed to buffered shadows (Lamps: Buffered Shadows), ray-traced shadows are obtained by casting rays from a regular light source, uniformly and in all directions. The ray-tracer then records which pixel of the final image is hit by a ray light, and which is not. Those that are not are obviously obscured by a shadow.

Each light casts rays in a different way. For example, a *Spot* light casts rays uniformly in all directions within a cone. The *Sun* light casts rays from a infinitely distant point, with all rays parallel to the direction of the *Sun* light.

For each additional light added to the scene, with ray-tracing enabled, the rendering time increases. Ray-traced shadows require more computation than buffered shadows but produce sharp shadow borders with very little memory resource usage.

To enable ray-traced shadows, three actions are required:

- Enable *Shadows* globally in the *Render* menu's *Shading* panel.
- Enable *Ray tracing* globally from the same panel.
- Enable ray-traced shadows for the light using the *Ray Shadow* button in the *Light* menu's *Shadow* panel.

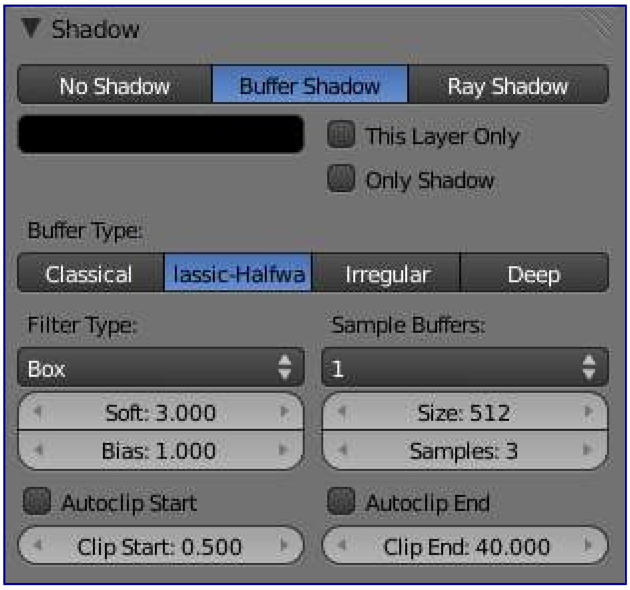
This panel varies depending on the type of light.

- All lamps able to cast ray-traced shadows share some common options, described in *Ray-traced Properties*.

Ray-traced shadows can be cast by the following types of lamp:

- *Point lamp*
- *Spot lamp*
- *Area lamp*
- *Sun lamp*

Lamps: Buffered Shadows



Buffer Shadow enabled for a Spot lamp

Cast Buffer Shadows enabled for a material

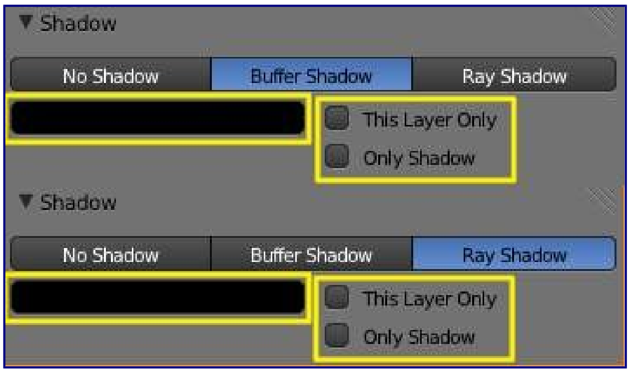
Buffered shadows provide fast-rendered shadows at the expense of precision and/or quality. Buffered shadows also require more memory resources as compared to ray tracing. Using buffered shadows depends on your requirements. If you are rendering animations or can’t wait hours to render a complex scene with soft shadows, buffer shadows are a good choice.

For a scanline renderer - and Blender’s built-in engine *is*, among other things, a scanline renderer - shadows can be computed using a *shadow buffer*. This implies that an “image”, as seen from the spot lamp’s point of view, is “rendered” and that the distance - in the image - for each point from the spot light is saved. Any point in the “rendered” image that is farther away than any of those points in the spot light’s image is then considered to be in shadow. The shadow buffer stores this image data.

To enable buffered shadows these actions are required:

- Enable shadows globally from the *Scene* menu’s *Gather* panel by selecting *Approximate*.
- Enable shadows for the light using the *Buffer Shadow* button in the *Lamp* menu’s *Shadow* panel.
- Make sure the *Cast Buffer Shadows* options is enabled in each *Material* ‘s *Shadow* panel.
- The *Spot lamp* is the only lamp able to cast buffered shadows.

Common Shadowing Lamps Options



Common shadowing options for lamps

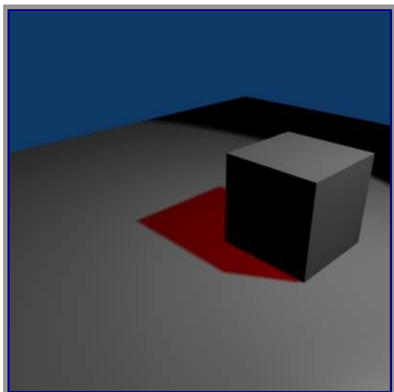
All lamps able to cast shadows (*Lamp*, *Spot*, *Area* and *Sun*) share some options, described below:

This Layer Only
: When this option is enabled, only the objects on the same layer as the light source will cast shadows.

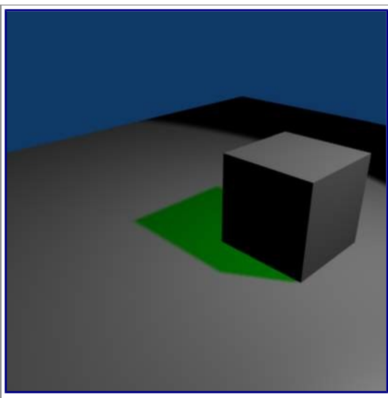
Only Shadow
: The light source will not illuminate an object but will generate the shadows that would normally appear. This feature is often used to control how and where shadows fall by having a light which illuminates but has no shadow, combined with a second light which doesn’t illuminate but has *Only Shadow* enabled, allowing the user to control shadow placement by moving the “Shadow Only” light around.

Shadow color

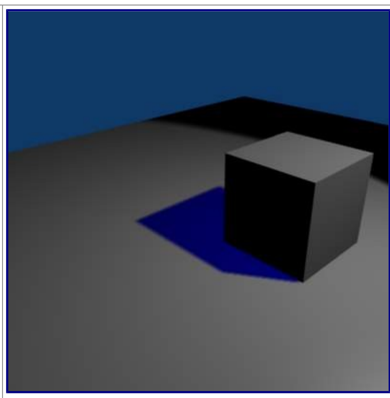
This color picker control allows you to choose the color of your cast shadows (black by default). The images below were all rendered with a white light and the shadow color was selected independently.



Red colored shadow example



Green colored shadow example



Blue colored shadow example

Although you can select a pure white color for a shadow color, it appears to make a shadow disappear.

See Also

- *Shadows*
- *Common Raytraced Options*
- *Lamp Light Raytraced Shadows*
- *Spot Light Raytraced Shadows*
- *Area Light Raytraced Shadows*
- *Sun Light Raytraced Shadows*
- *Spot Light Buffered Shadows*

Lamps Raytraced Shadows

Ray shadowing options for lamps

Most lamp types (*Lamp*, *Spot* and *Sun*) share the same options for the ray-traced shadows generation, which are described below. Note that the *Area* lamp, even though using most of these options, have some specifics described in its *own ray-traced shadows page*.

Ray Shadow

The *Ray Shadow* button enables the light source to generate ray-traced shadows. When the *Ray Shadow* button is selected, another set of options is made available, those options being:

Shadow sample generator type

Method for generating shadow samples: Adaptive QMC is fastest, Constant QMC is less noisy but slower. This allows you to choose which algorithm is to be used to generate the samples that will serve to compute the ray-traced shadows (for now, mainly two variants of Quasi-Monte Carlo, see What is Quasi-Monte Carlo?):

Constant QMC

The *Constant QMC* method is used to calculate shadow values in a very uniform, evenly distributed way. This method results in very good calculation of shadow value but it is not as fast as using the *Adaptive QMC* method; however, *Constant QMC* is more accurate.

Adaptive QMC

The *Adaptive QMC* method is used to calculate shadow values in a slightly less uniform and distributed way. This method results in good calculation of shadow value but not as good as *Constant QMC*. The advantage of using *Adaptive QMC* is that it is in general much quicker while being not much worse than *Constant QMC* in terms of overall results.

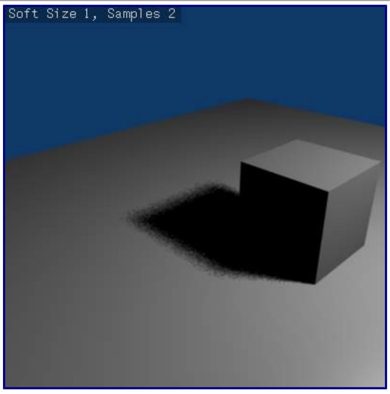
Samples

Number of extra samples taken (samples x samples). This slider sets the maximum number of samples that both *Constant QMC* and *Adaptive QMC* will use to do their shadow calculations. The maximum value is `16` - the real number of samples is actually the square of it, so setting a sample value of `3` really means $3^2 = 9$ samples will be taken.

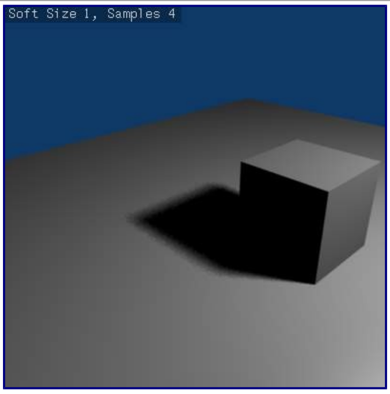
Soft Size

Light size for ray shadow sampling. This slider determines the size of the fuzzy/diffuse/penumbra area around the edge of a shadow. *Soft Size* only determines the width of the soft shadow size, not how graded and smooth the shadow is. If you want a wide shadow which is also soft and finely graded, you must also set the number of samples in the *Samples* field higher than `1`; otherwise this field has no visible effect and the shadows generated will not have a soft edge. The maximum value for *Soft Size* is `100.0`.

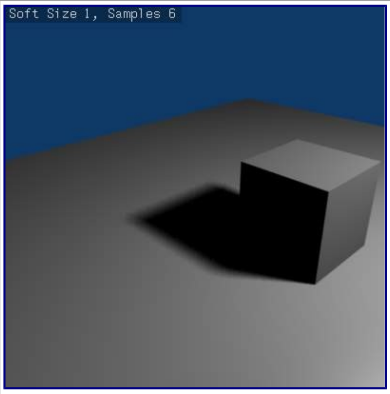
Below is a table of renders with different *Soft Size* and *Samples* settings showing the effect of various values on the softness of shadow edges:



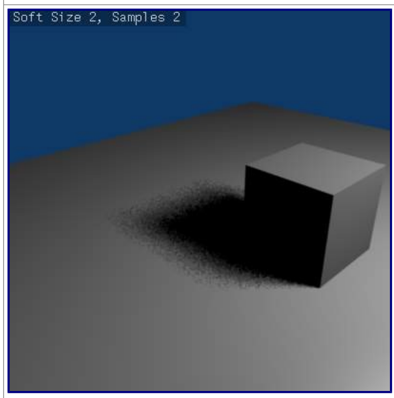
Soft Size: 1.0, Samples: 2.



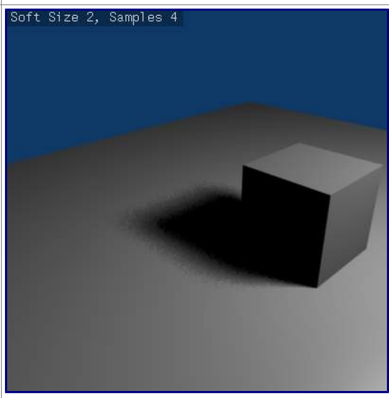
Soft Size: 1.0, Samples: 4.



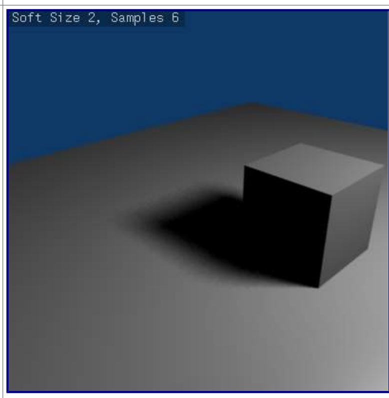
Soft Size: 1.0, Samples: 6.



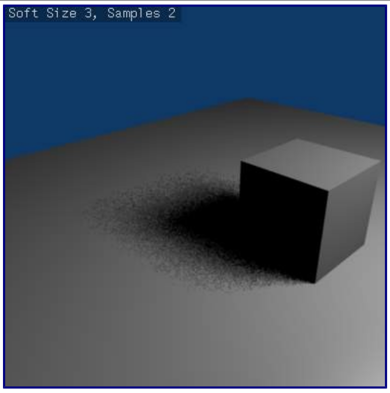
Soft Size: 2.0, Samples: 2.



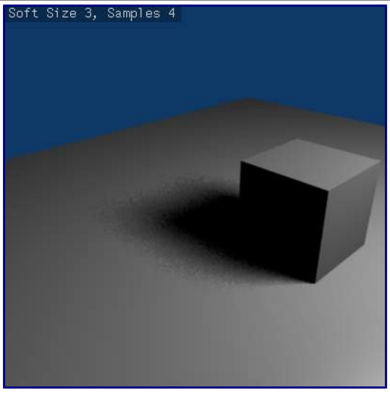
Soft Size: 2.0, Samples: 4.



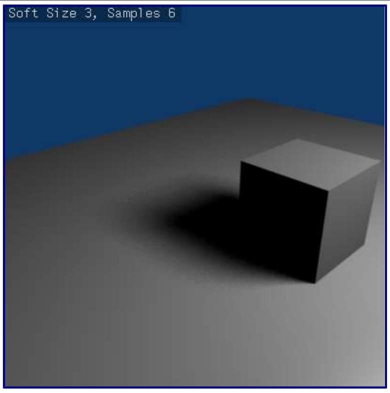
Soft Size: 2.0, Samples: 6.



Soft Size: 3.0, Samples: 2.

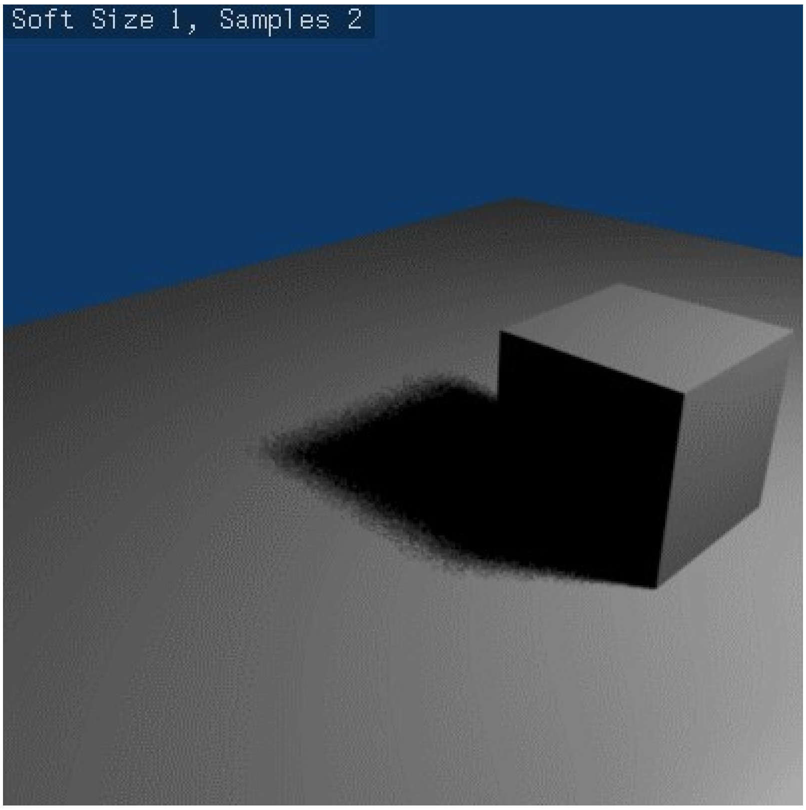


Soft Size: 3.0, Samples: 4.



Soft Size: 3.0, Samples: 6.

Below is an animated version of the above table of images showing the effects:



Animated version renders with different Soft Size and Samples settings showing the effect of various values on the softness of shadow edges.

Threshold

Threshold for Adaptive Sampling. This field is used with the *Adaptive QMC* shadow calculation method. The value is used to determine if the *Adaptive QMC* shadow sample calculation can be skipped based on a threshold of how shadowed an area is already. The maximum *Threshold* value is `1.0`.

What is Quasi-Monte Carlo?

The Monte Carlo method is a method of taking a series of samples/readings of values (any kind of values, such as light values, color values, reflective states) in or around an area at random, so as to determine the correct actions to take in certain calculations which usually require multiple sample values to determine overall accuracy of those calculations. The Monte Carlo method tries to be as random as possible; this can often cause areas that are being sampled to have large irregular gaps in them (places that are not sampled/read). This in turn can cause problems for certain calculations (such as shadow calculation).

The solution to this was the Quasi-Monte Carlo method.

The Quasi-Monte Carlo method is also random, but tries to make sure that the samples/readings it takes are also better distributed (leaving less irregular gaps in its sample areas) and more evenly spread across an area. This has the advantage of sometimes leading to more accurate calculations based on samples/reading.