White Paper

Soft Shadows

February 2007
WP-03016-001_v01

Document Change History

Version	Date	Responsible	Reason for Change
01	February 21, 2007	KD, SM	Initial release

Go to sdkfeedback@nvidia.com to provide feedback on Soft Shadows.

Soft Shadows

Abstract

Soft Shadows is an extremely useful feature in any computer game. According to the physics of light transport, sharp shadows can be created only by an ideal point light source, which means that the light source size is zero. In real life, lights of zero size don't exist. Therefore, shadows we see every day are soft and their softness varies depending on the light source.

This demo presents two algorithms for soft shadow rendering. The common advantage of the presented algorithms is that in contrast with standard PCF (Percentage Closer Filtering) algorithms, they do not use pseudo-random samples to sample the shadow map. That is why partially shadowed areas lack noise typically introduced by PCF algorithms.

The first algorithm tries to be physically accurate. It is more expensive, but gives results close to what one would see in real life. It accounts for the fact that shadow softness depends on ratio of two distances:

- From light source to light occluder
- From light source to shadow receiver

The second algorithm is more like a fixed kernel percentage closer filter. It softens the shadow by a constant user defined amount.

Kirill Dmitriev
NVIDIA Corporation

How It Works

The basic structure used by both algorithms is a min-max depth mipmap. A min-max mipmap is created from a standard shadow map. Suppose we have a 1024 × 1024 shadow map. On the first pass we create from it a 512 × 512 texture with two channels. The first channel of the coarser mip level contains the min of the four corresponding pixels in the finer mip level. The second channel contains the max of these four pixels.

Figure 1 shows the min-max mipmap creation, from four pixels of the finer level and we get one pixel on the coarser level.

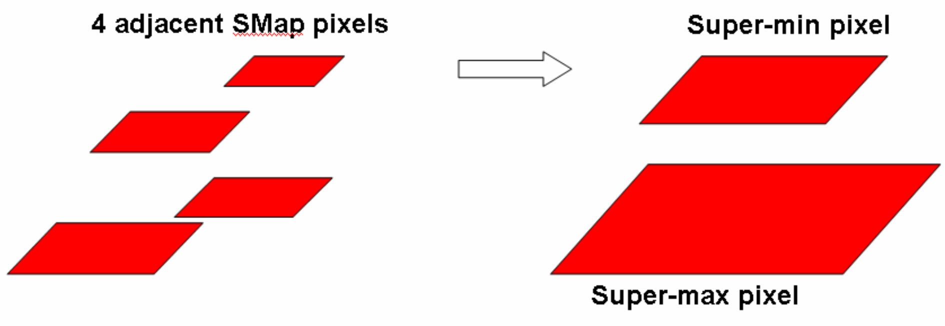

Figure 1. Min-max Mipmap Creation

Fast Algorithm

The fast soft shadow algorithm implements standard PCF with a fixed kernel. For example, for shaded fragment:

- It projects this fragment on to the shadow map
- It surrounds the point where it projects by square kernel K × K pixels
- It computes what is the number C of pixels that are closer to light source than the fragment

Shadowing is then equal to C/(K^2). If the kernel contains too many shadow map pixels, it is computationally expensive to compute the shadowing. Here is where the structure of the min-max mipmap comes to the rescue. Let's denote the z value of the current fragment as Zf. We start looping through shadow map pixels at the most coarse 2 × 2 mip level of the mipmap. If (Zf < min), none of the shadow map pixels can potentially cover the fragment and shadowing is 0. Alternatively, if (Zf > max), all of shadow map pixels occlude the fragment from light and shadowing is 1. Third possible case is when (min < Zf < max). In this case we can't conservatively answer the shadow query and we have to descend down to finer levels of the min-max mipmap. We do it recursively until we reach the level where we can answer the shadow query conservatively. Worst case scenario, we reach the finest mipmap level where the situation (min < Zf < max) is never possible because at that level min = max.

Accurate Algorithm

The accurate algorithm works by back projecting the shadow map pixels on to the light source plane[1] (Figure 2). It assumes that light source is a square of user-specified size hanging in 3D space. After back projection, it is straightforward to compute shadowing. We simply need to subtract areas of all projected pixels from the light source area. This is a very simple geometric task when performed in the light source coordinate system. In this system both light source and projected pixels are 2 dimensional AABBs (axis-aligned bounding boxes).

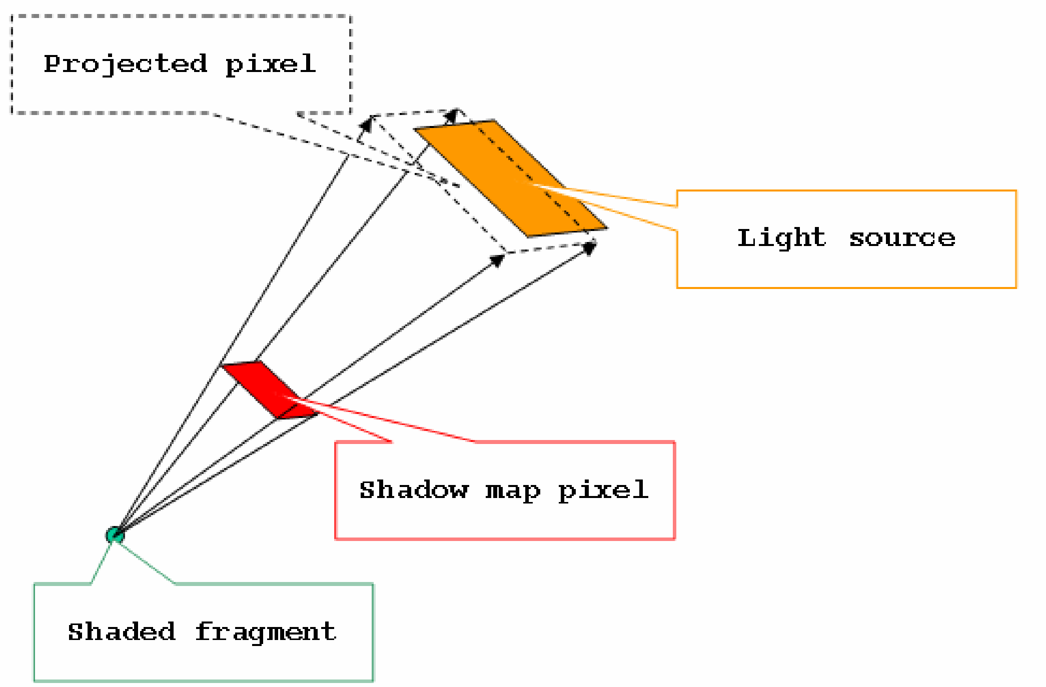

Figure 2. Back Projection of Shadow Map Pixel on to the Light Source Plane

Even though the task of back projecting each shadow map pixels is geometrically simple, doing it for each 1 M pixels (if we have 1024 × 1024 shadow map) and for each fragment would be prohibitively expensive. The min-max mipmap comes to the rescue again. The idea is that if for some super-min pixel (see Figure 1 for super-min definition) on the coarse mip level we find that it completely shadows the fragment and (Zf > super-max depth), then we do not have to descend to finer mip levels. In this case we can immediately say that shadowing is 1. Alternatively, if we find that shadowing by super-min is 0, then we can conservatively say that shadowing by all corresponding sub-pixels on the finer mip-levels is also 0.

[1] Gael Guennebaud, Loic Barthe and Mathias Paulin. *Real-Time Soft Shadow Mapping by Back Projection*. Europgraphics Symposium on Rendering 2006, Nicosia, Cyprus.

NVIDIA Corporation
2701 San Tomas Expressway
Santa Clara, CA 95050
www.nvidia.com