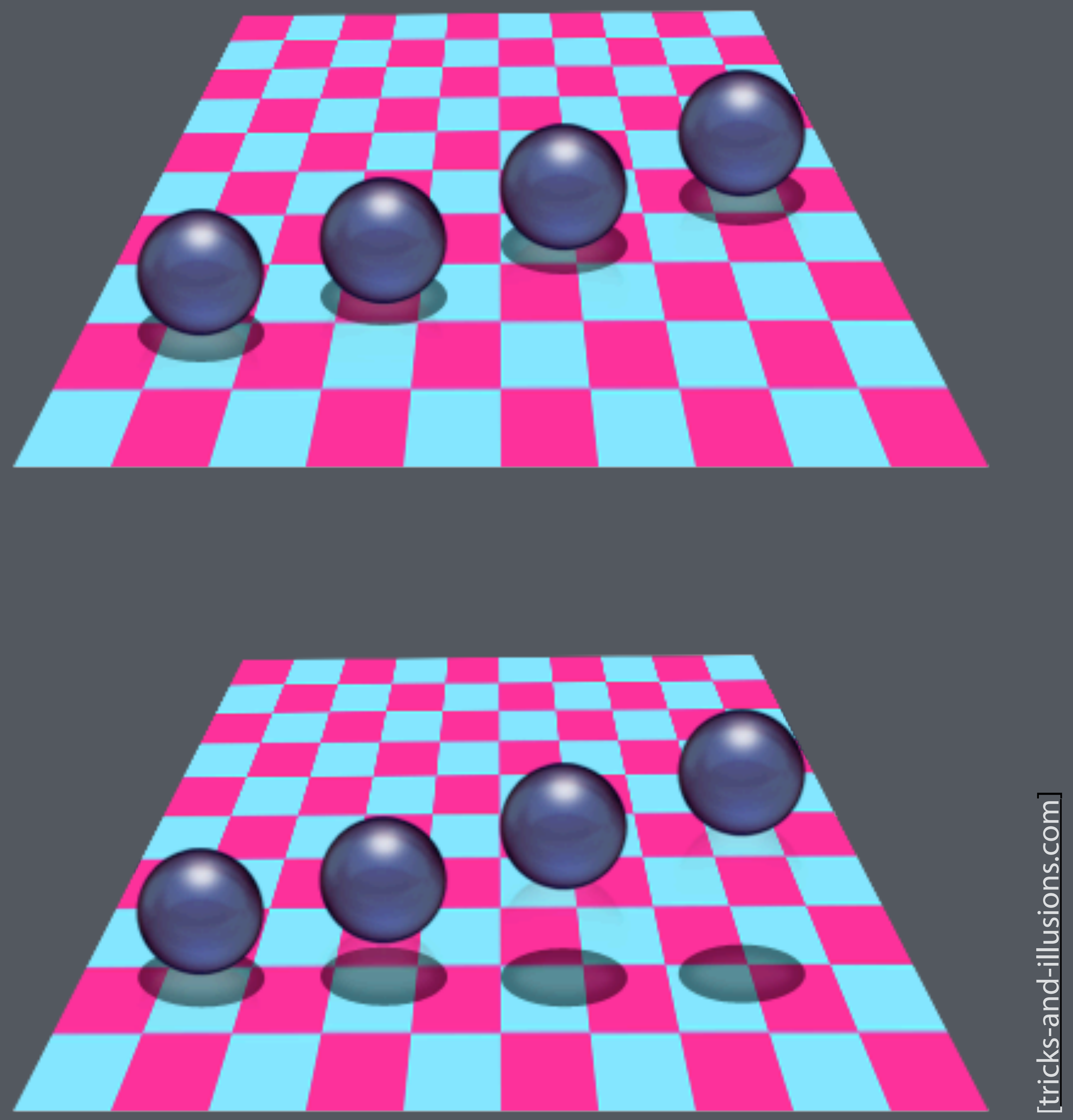
06 Shadow Mapping

Steve Marschner

CS5625 Spring 2022

Thanks to previous instructor Kavita Bala

Shadows as depth cue



Shadows as anchors


REALISTIC RAY TRACING
PETER SHIRLEY


REALISTIC RAY TRACING
PETER SHIRLEY

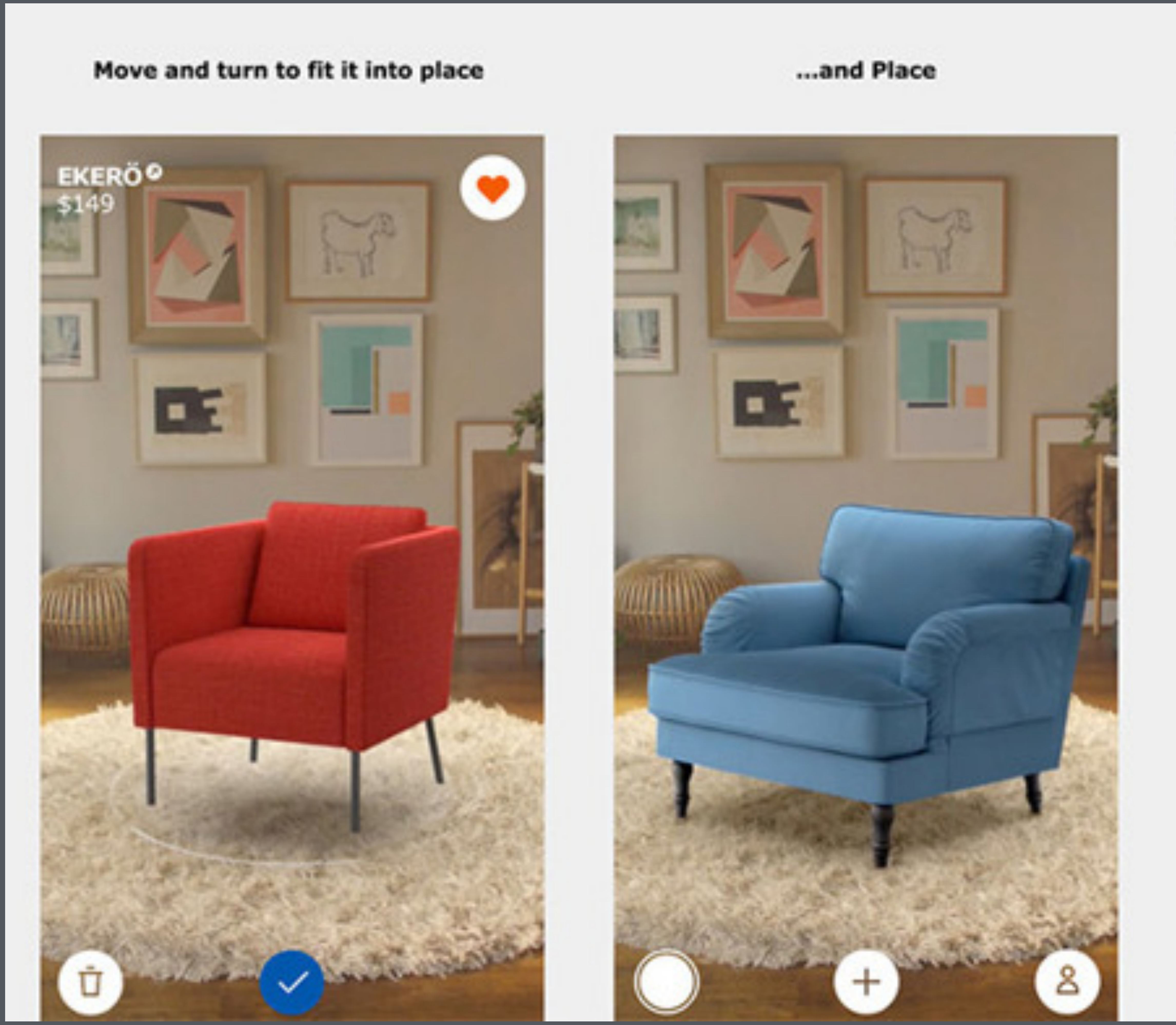
Fake shadows

Before we get into more complex methods…

- if a shadow is just needed to help anchor an object to a plane, very simple techniques can suffice
- classic: project shape of object, blur, use as mask to darken floor

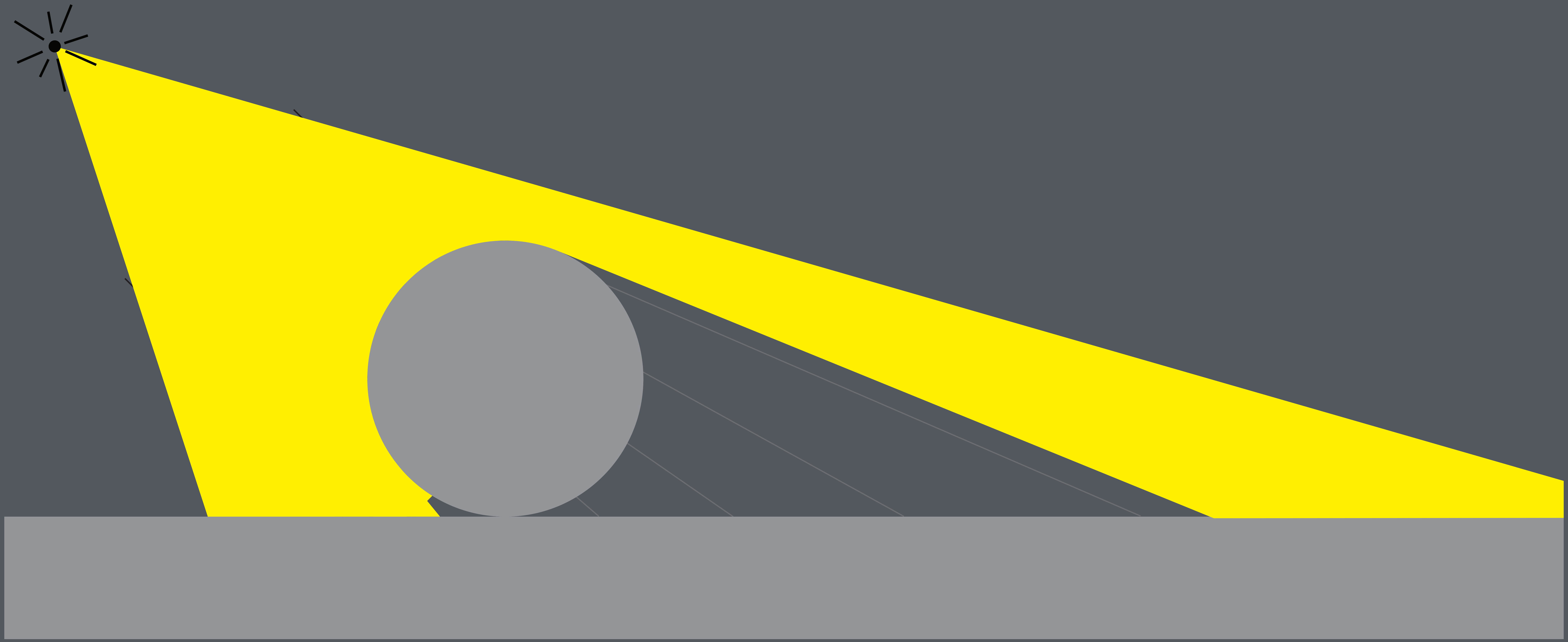
Shadow baking

- a more principled approach
- establish texture coordinates on floor
- for each texel compute irradiance
- perfectly accurate for diffuse receivers when the light and all geometry are static



IKEA Place iOS app

Shadow maps



Shadow maps

Main idea: reuse the z-buffer mechanism to test for light source visibility

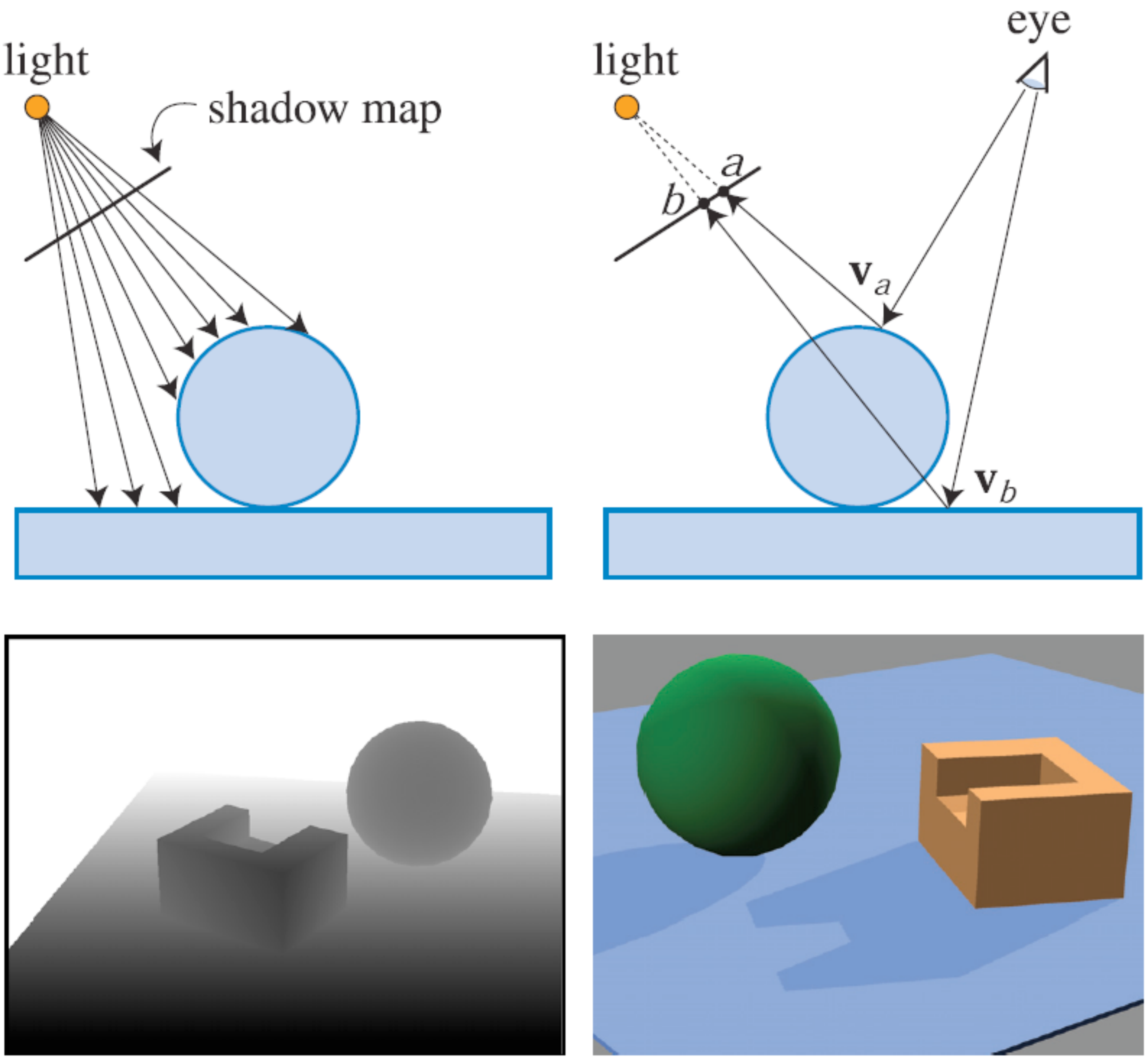
- introduced by Lance Williams in 1978
- very widely used approach for point-like lights

Shadow testing and visibility testing are similar problems

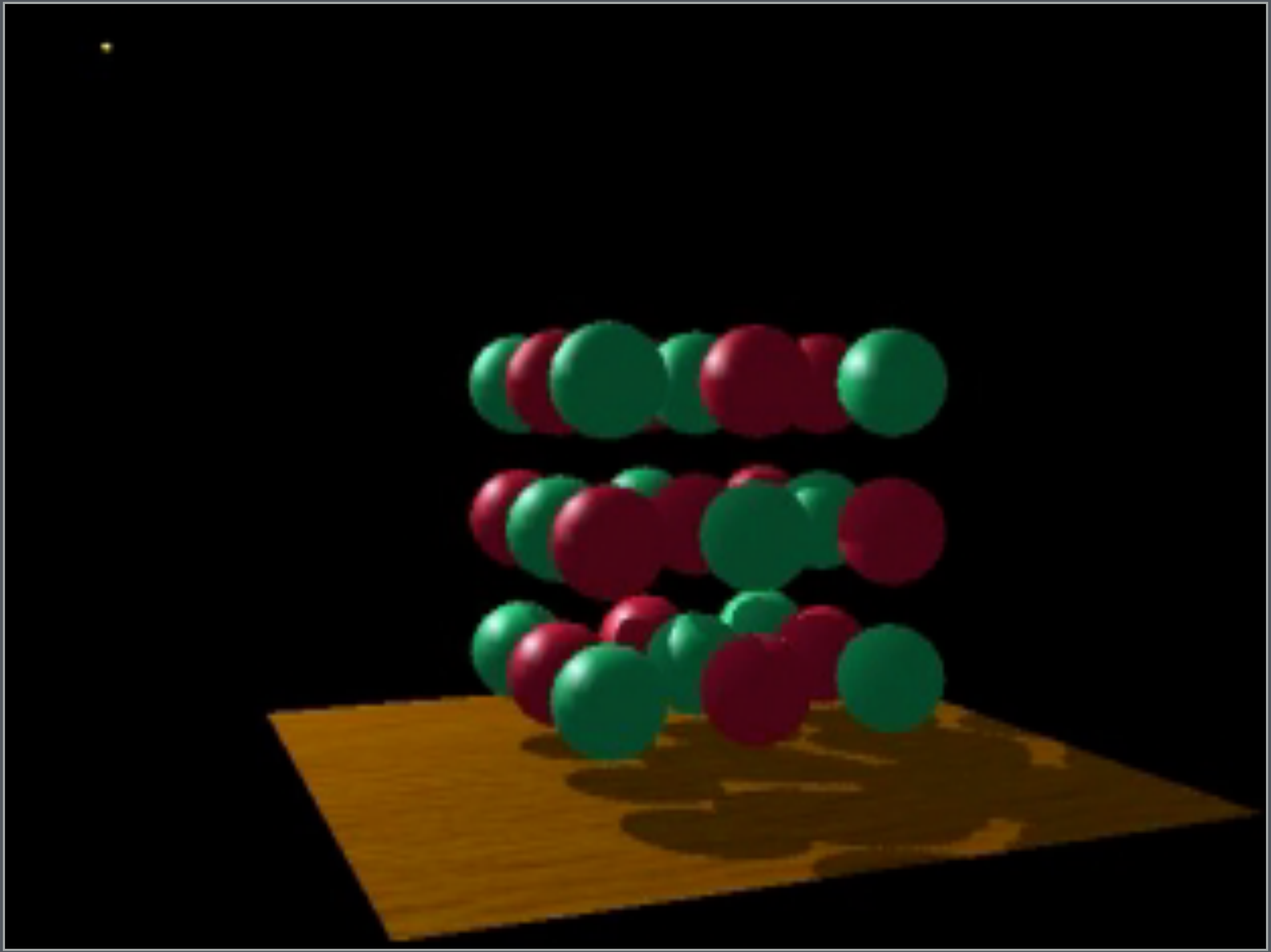
- given a point on a surface, is it visible to an { observer | light } at a fixed location?
- for visibility: interpolate screen-space (x,y,z); consider depth buffer value stored at screen-space (x,y); z <= buffer(x,y) implies visible
- for shadow: compute light-space (x,y,z) of fragment; z <= buffer(x,y) implies illuminated

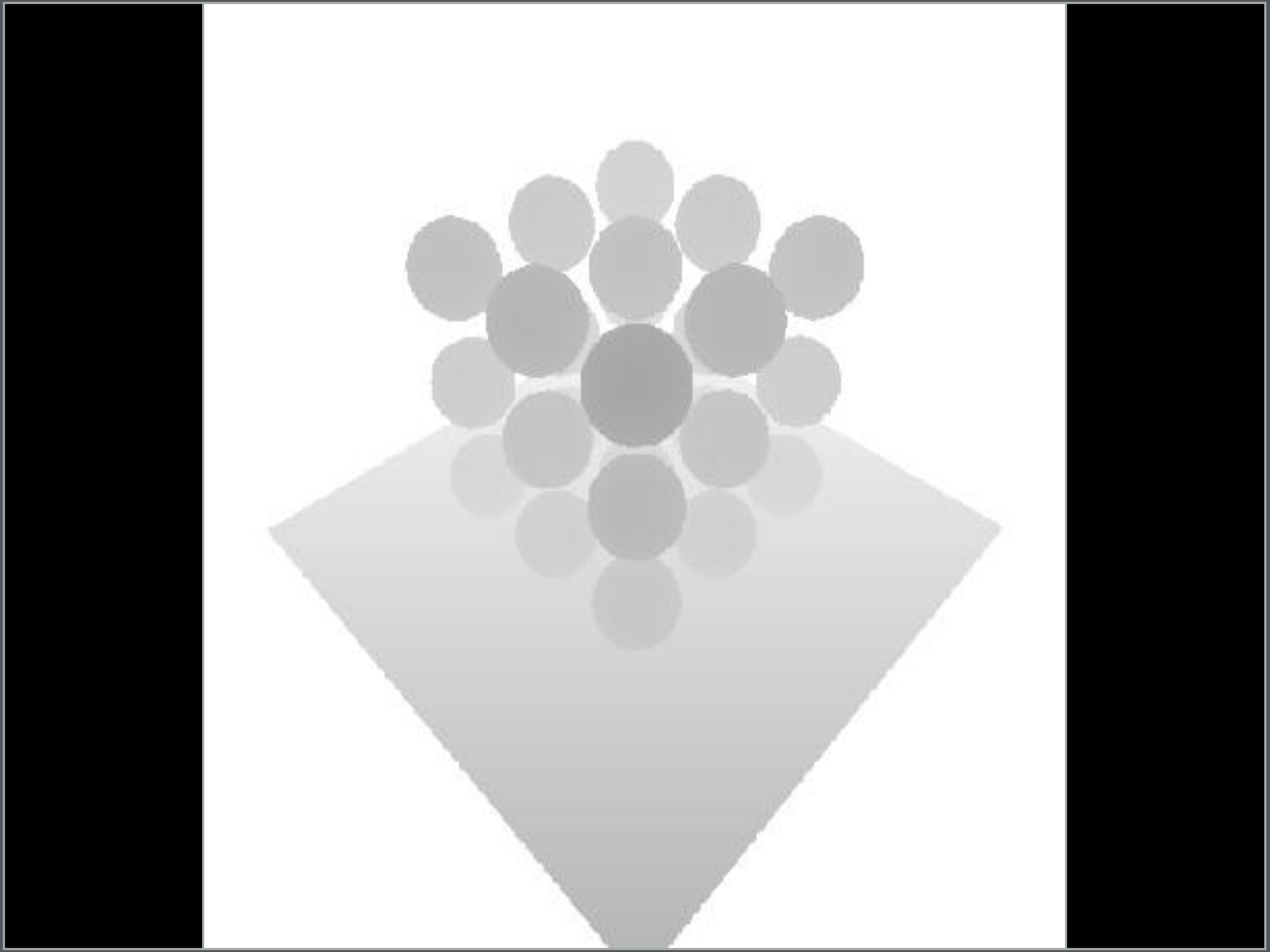
Some serious differences in practice

- most notably: fragments do not line up with depth buffer samples (they are scattered irregularly in light space)



[Möller et al. *RTR*]

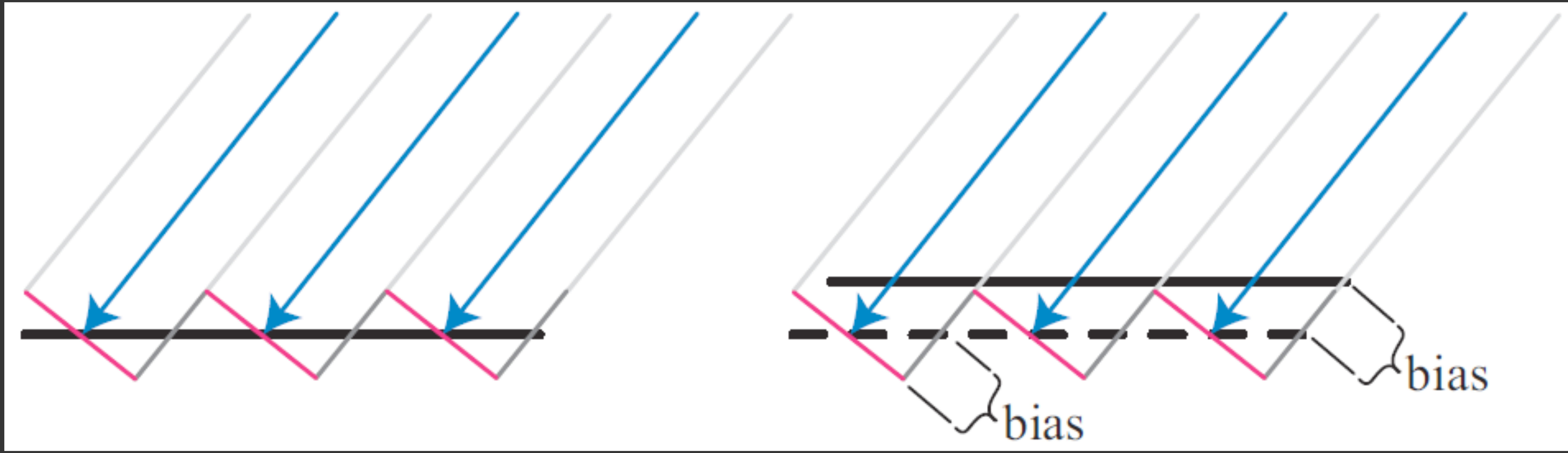

Mark Kilgard

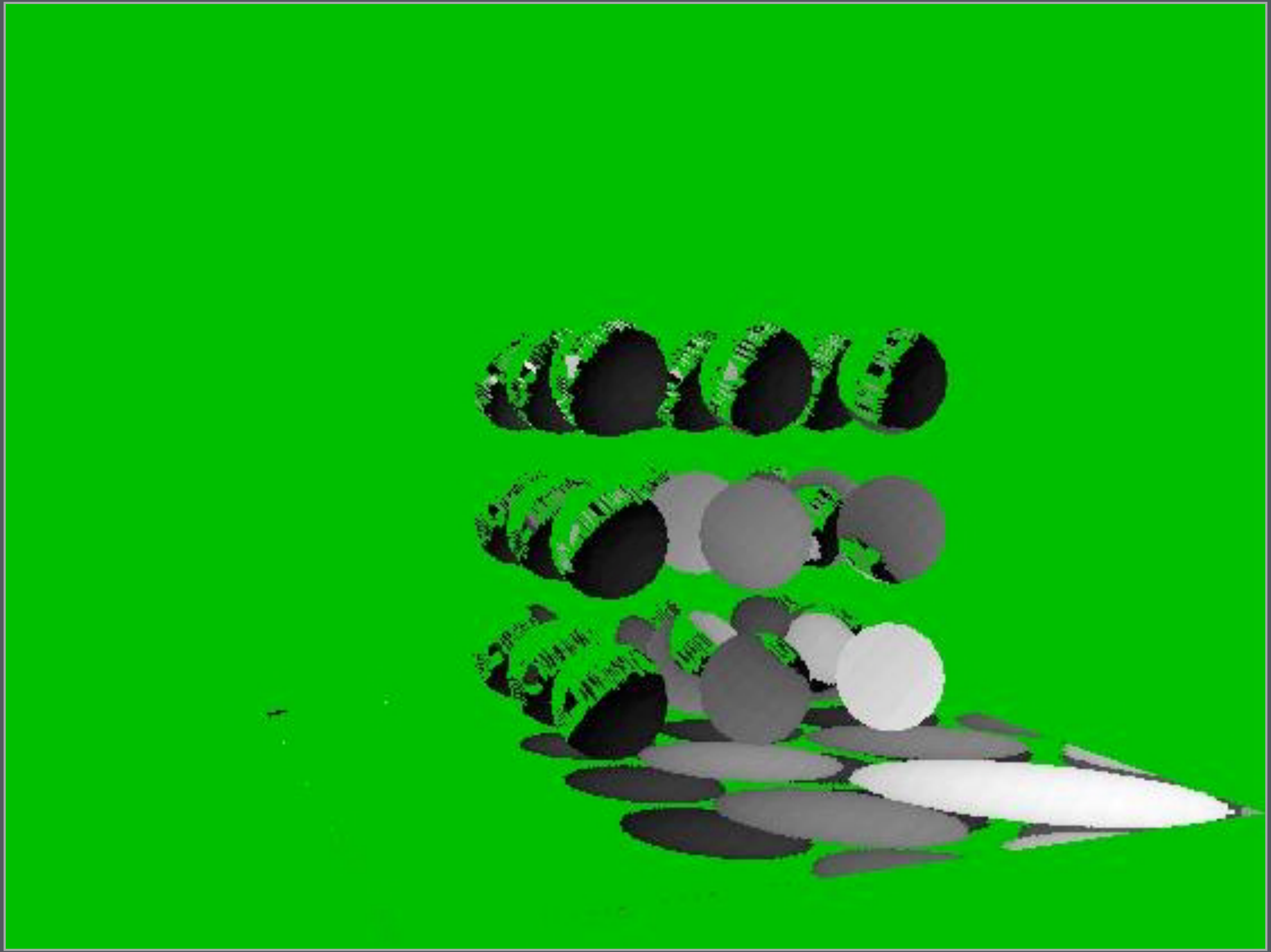


Mark Kilgard

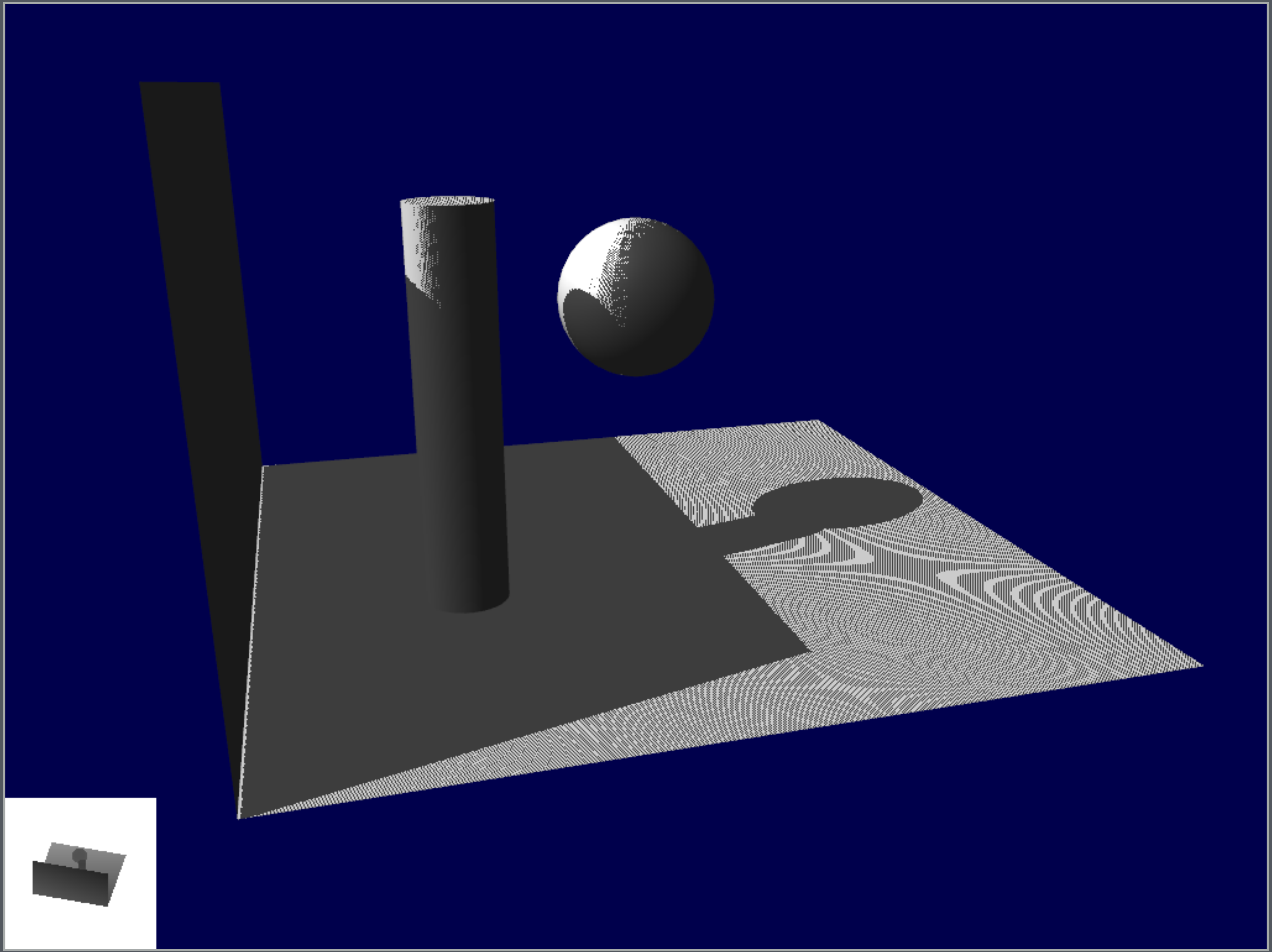
Shadow Map Issues

- if A and B are approximately equal?
- Speckling

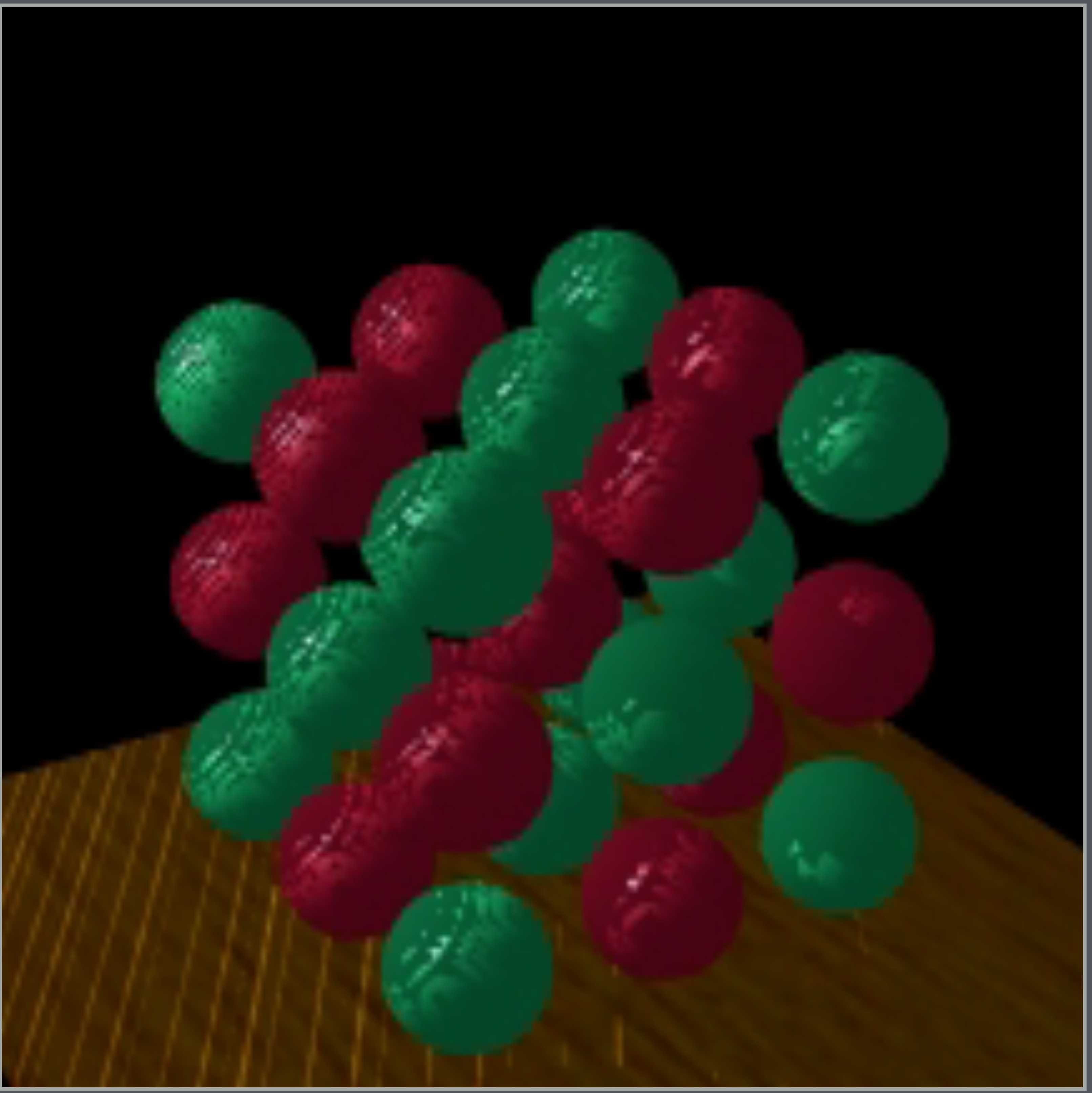




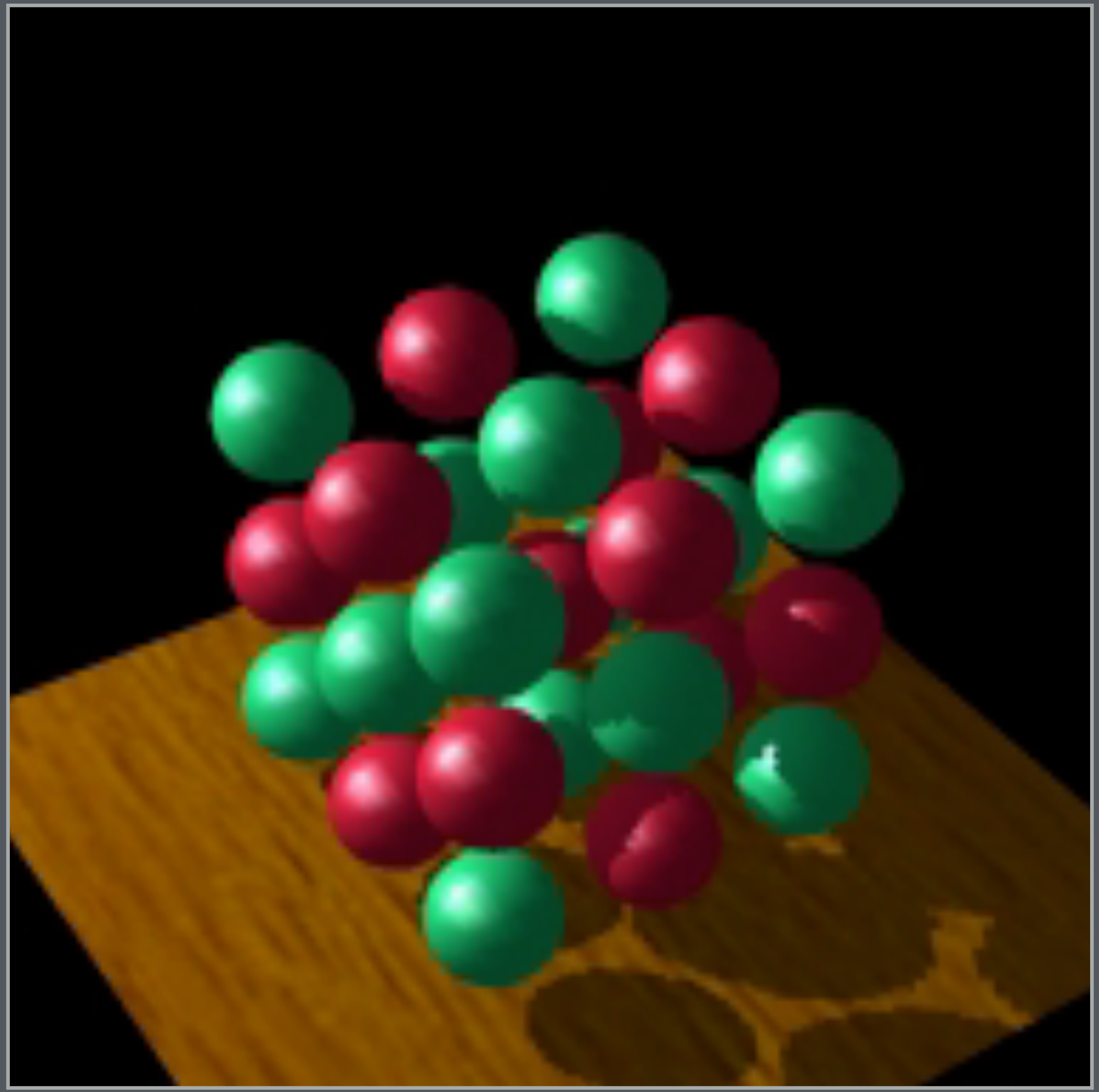
Mark Kilgard



first try at shadow mapping

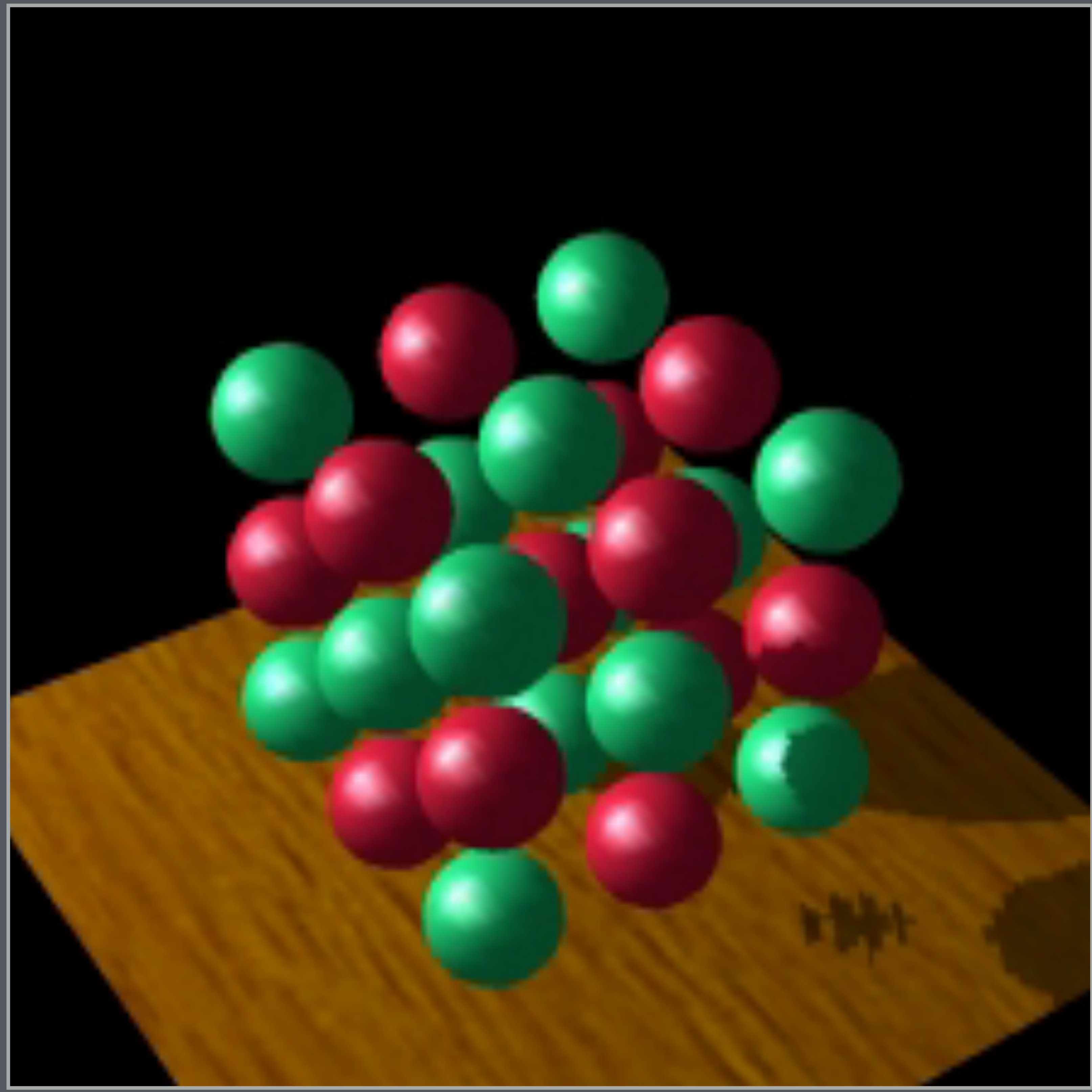


not enough shadow bias

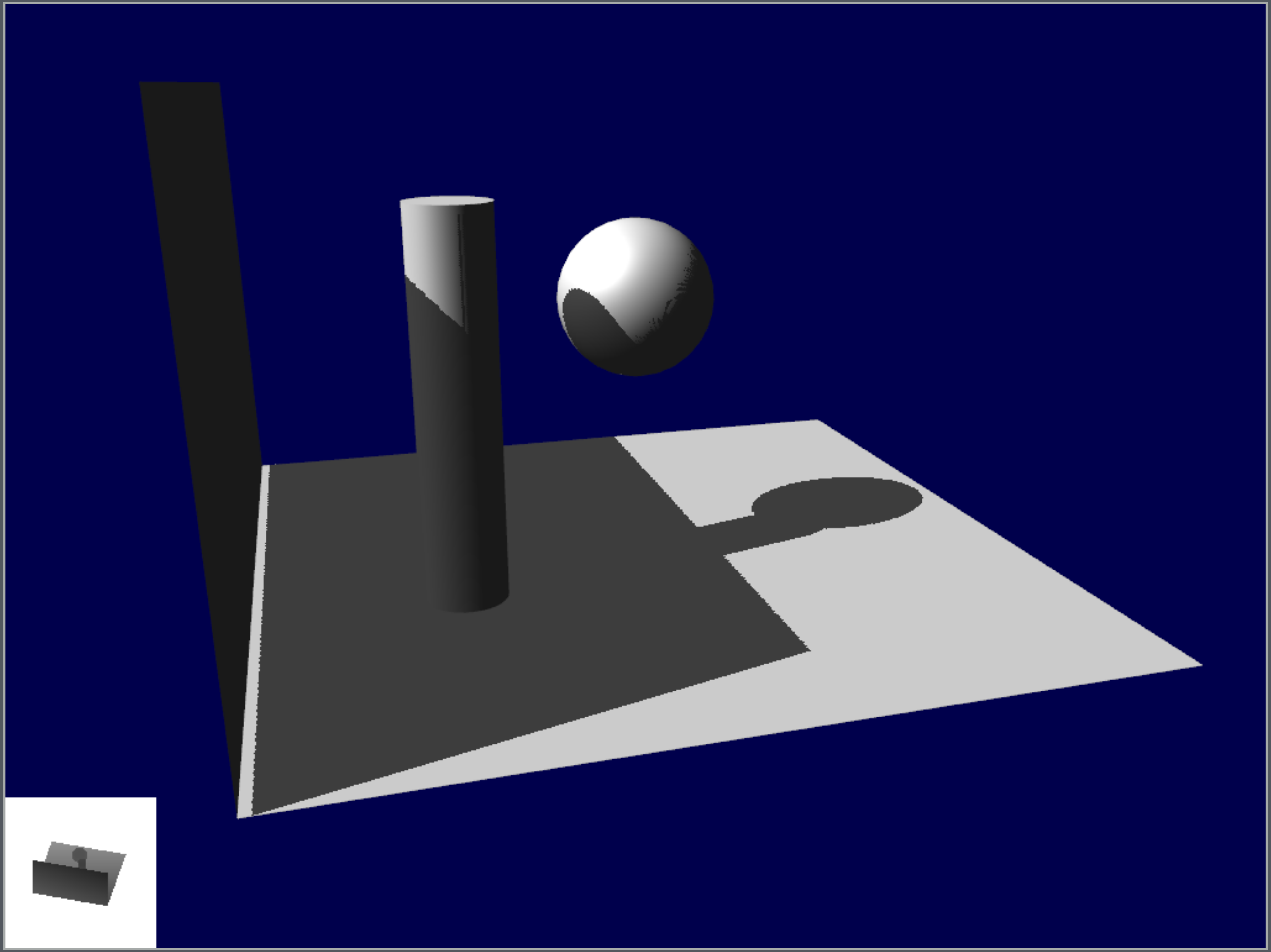


good shadow bias

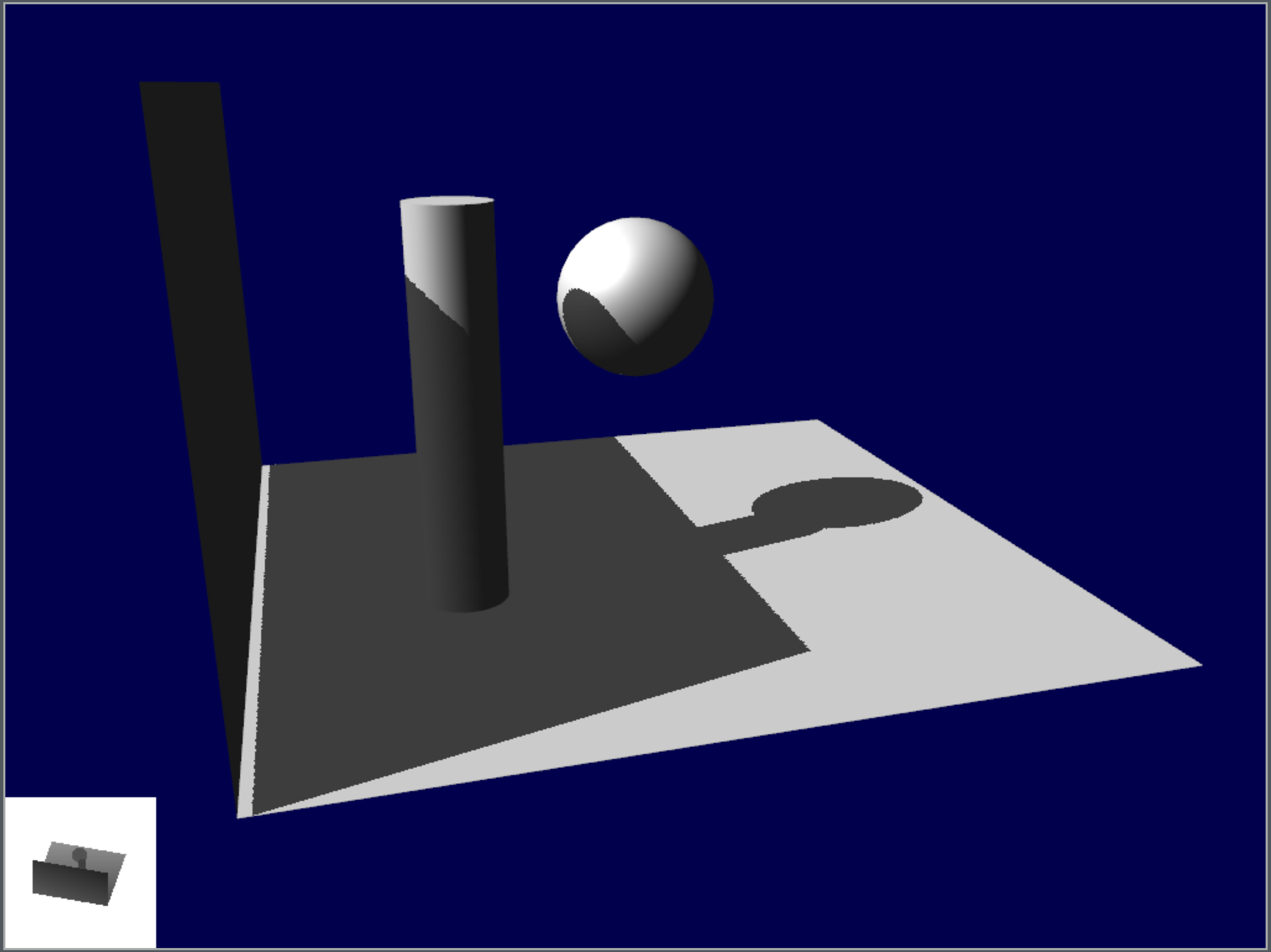
Mark Kilgard



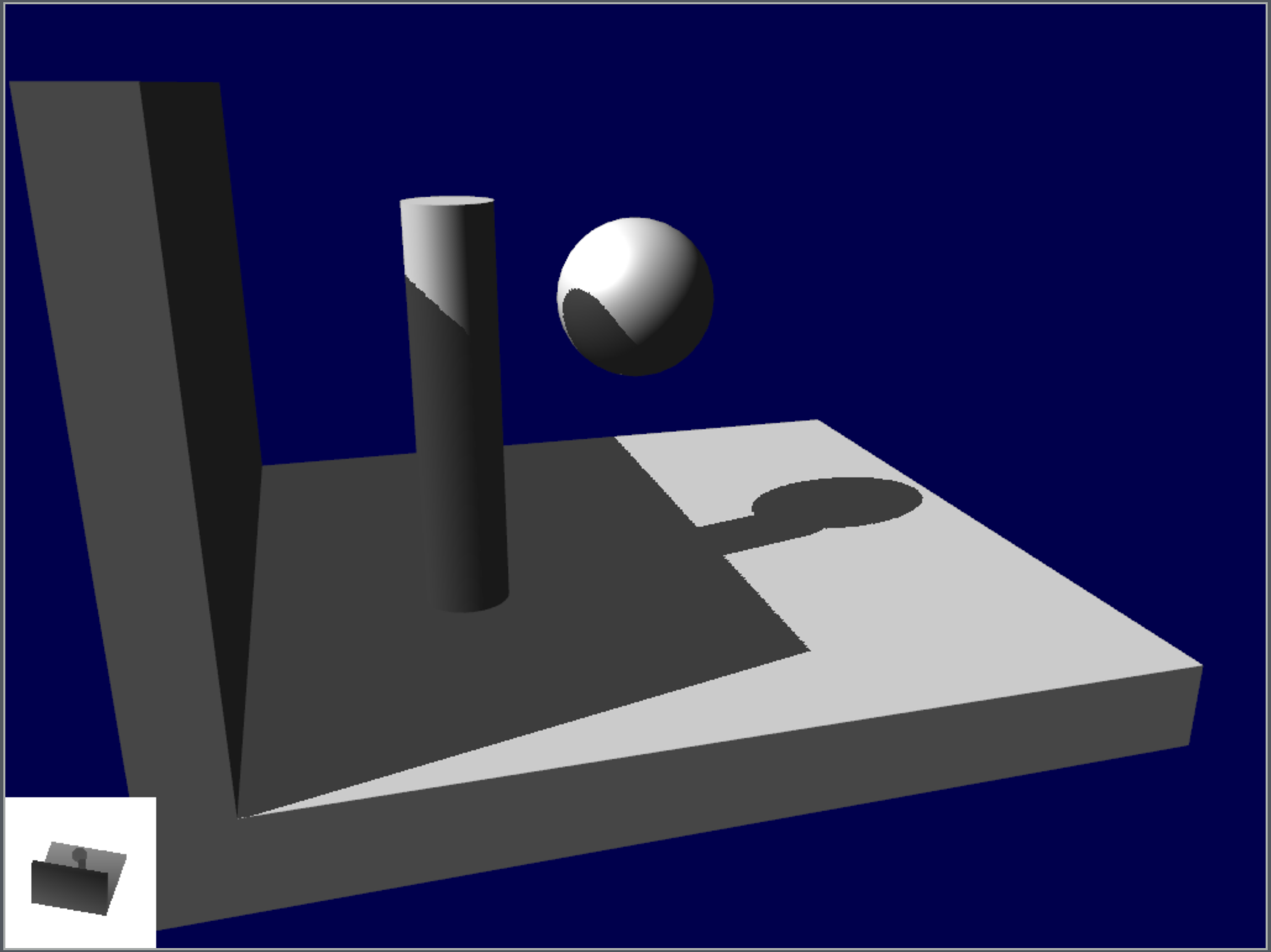
too much shadow bias



shadow mapping with constant bias



shadow mapping with slope-dependent bias



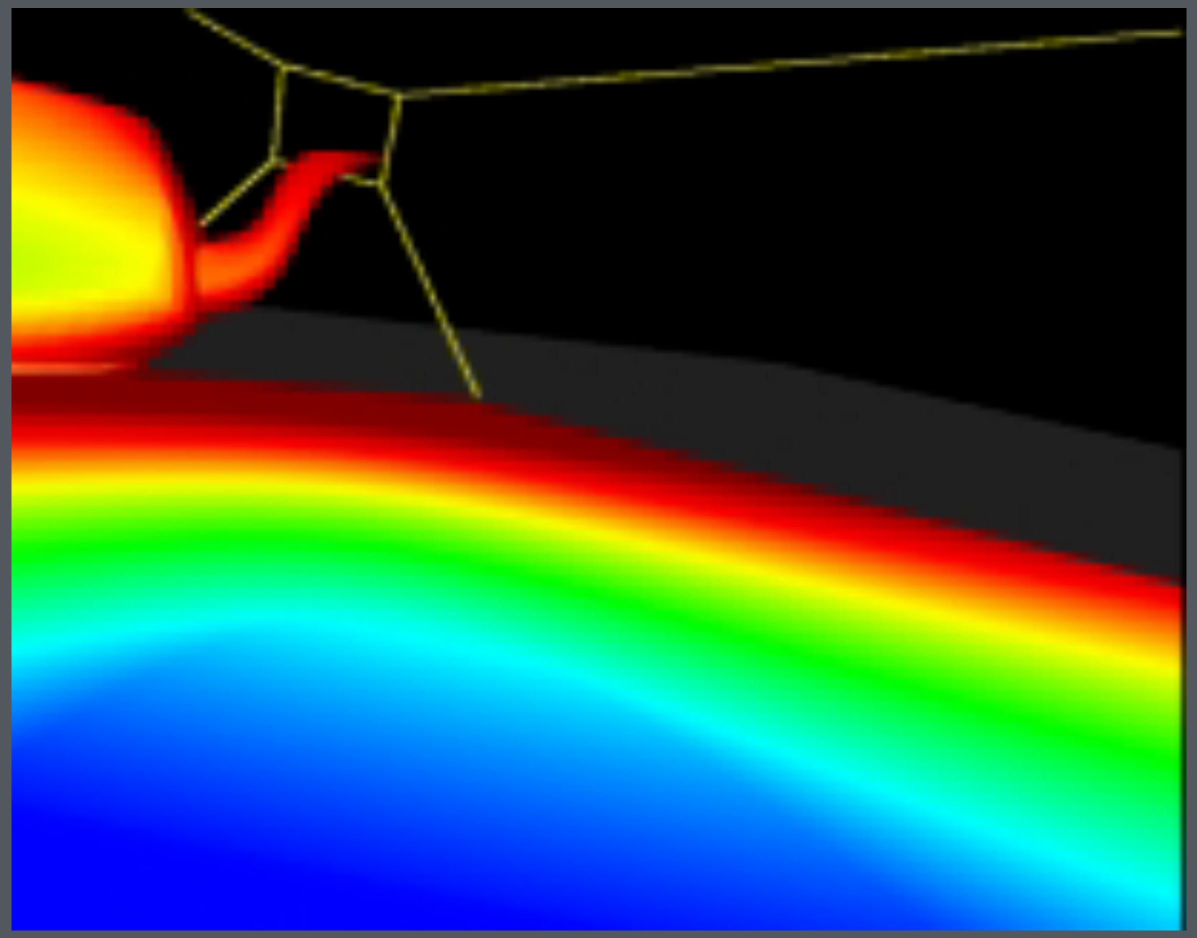
closed surfaces and slope-dependent bias

Shadow map sample rate—bad case

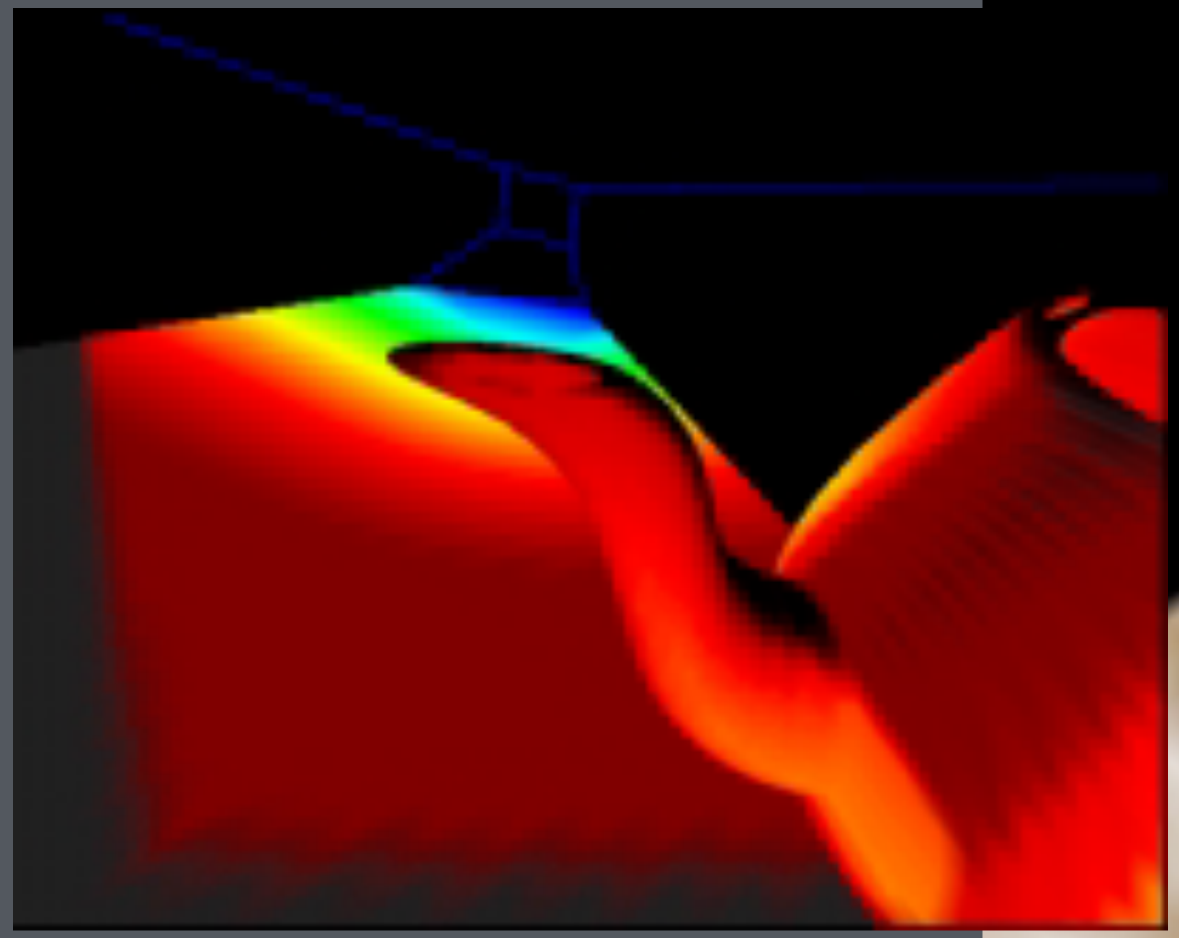
Light behind object

Light's "view direction" almost opposite the eye's view direction

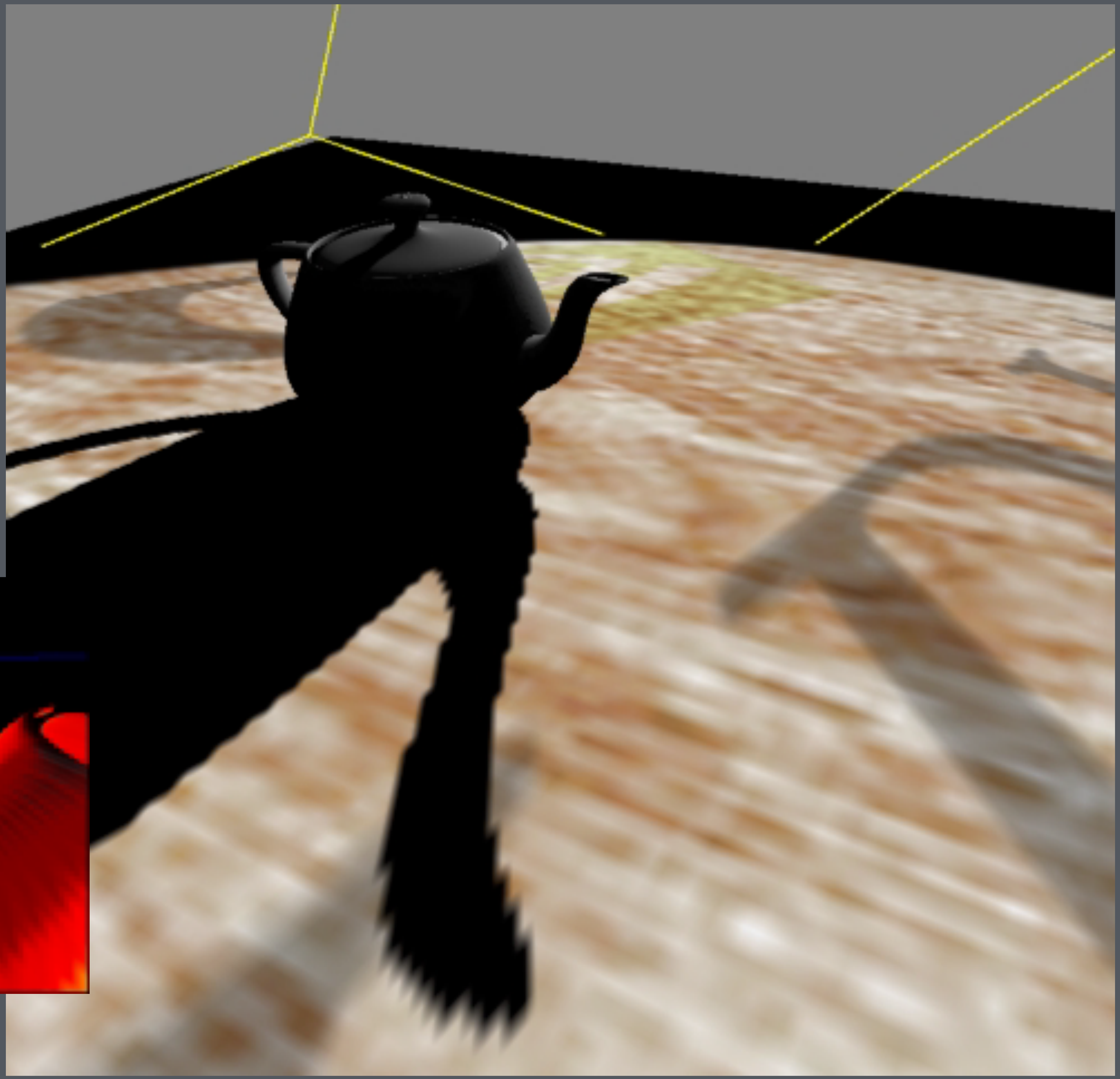
"Dueling frusta"



eye view



light view



Mark Kilgard

Cascaded shadow maps (aka. parallel-split SM)

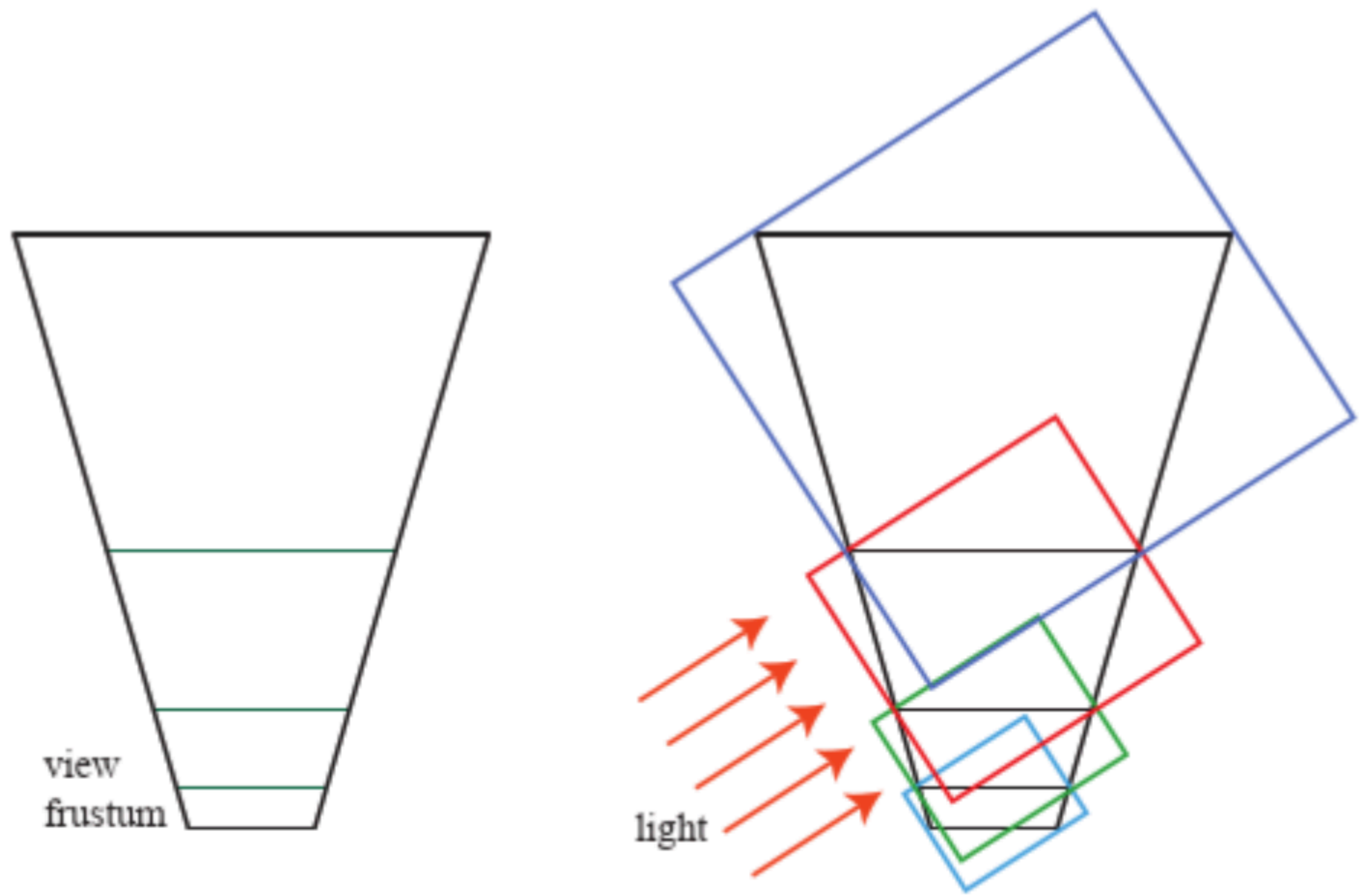


Figure 7.18. On the left, the view frustum from the eye is split into four volumes. On the right, bounding boxes are created for the volumes, which determine the volume rendered by each of the four shadow maps for the directional light. *(After Engel [430].)*

[Möller et al. *RTR*]

Cascaded shadow maps

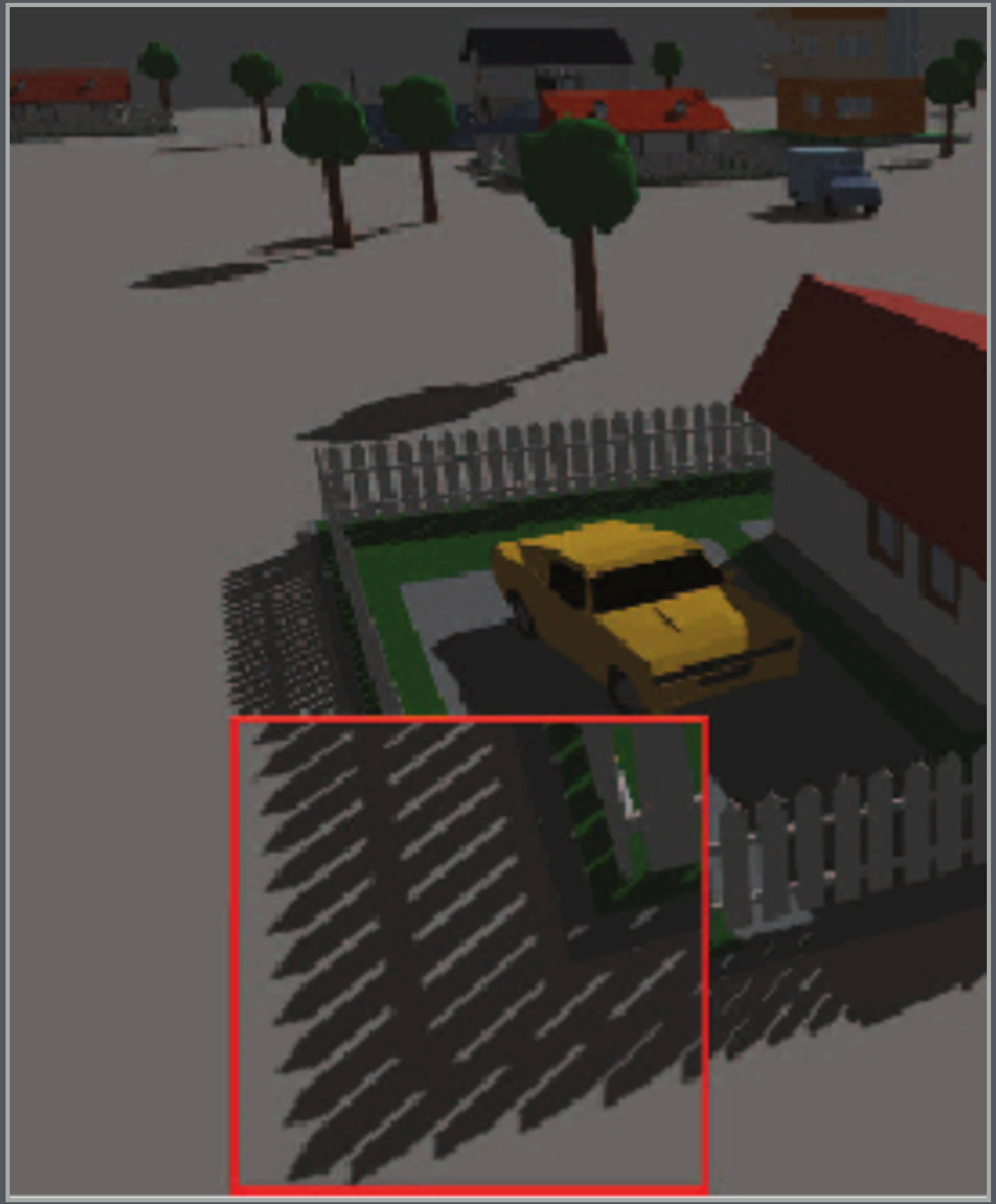
Idea: split the view volume

- cut into several slabs by depth
- handle shadows in each slab with a separate shadow map
- compute shadow frusta to exactly bound each piece
- use fragment depth to decide which map to sample

Design choices

- how to split the depth range (often logarithmically)
- set near distance with great care (has big effect on resolution of shadows)
- can be smarter about bounds: only need to bound objects, not whole view volume…


Single shadow map, 2048x2048


Four 1024x1024 shadow maps (equal memory)

Fan Zhang, Chinese U. Hong Kong

Filtering shadow maps

Shadow map lookups cause aliasing, need filtering

As with normal maps, pixel is a nonlinear function of the shadow depth

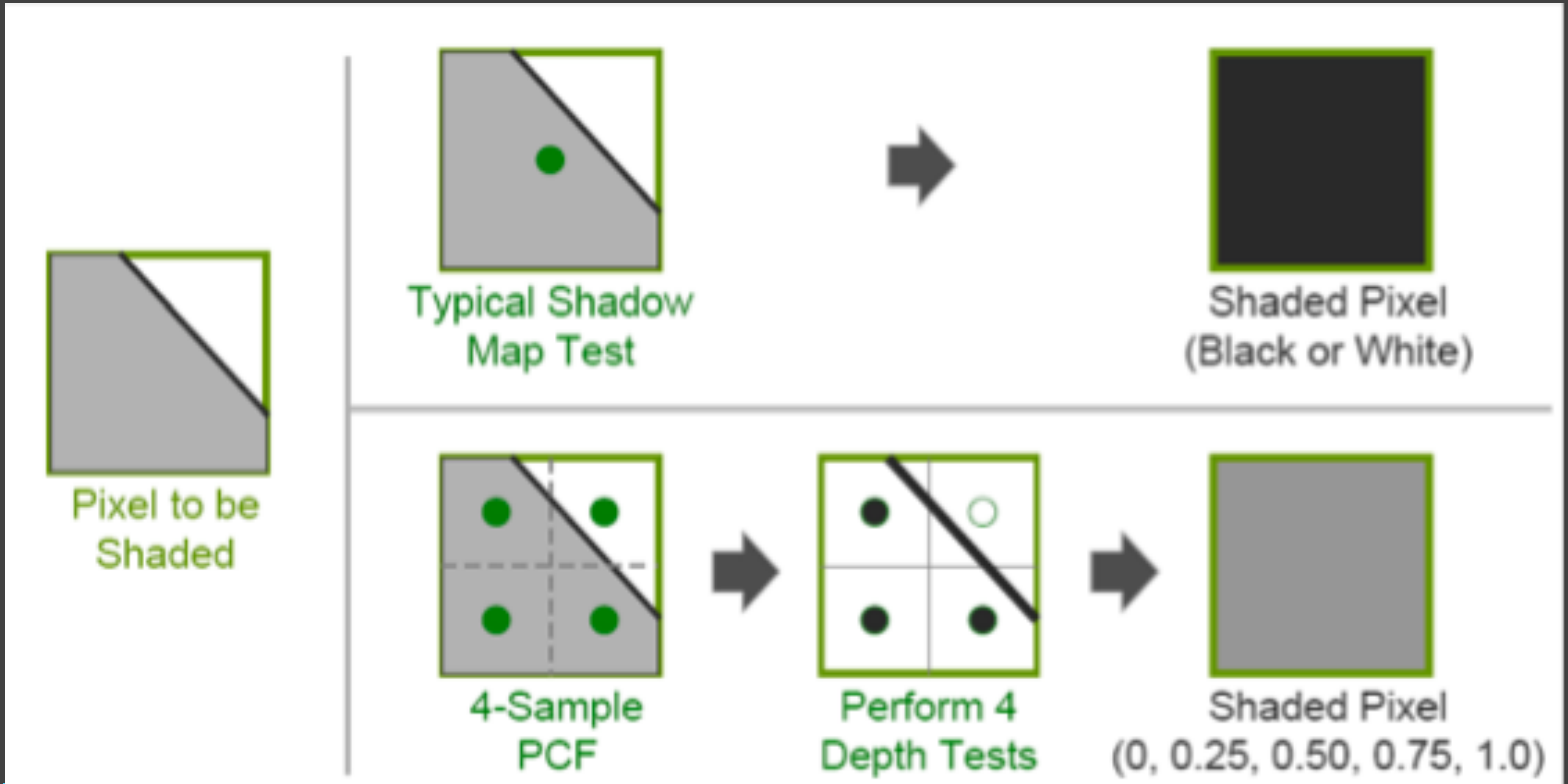
- this means applying a linear filter to the depth is wrong

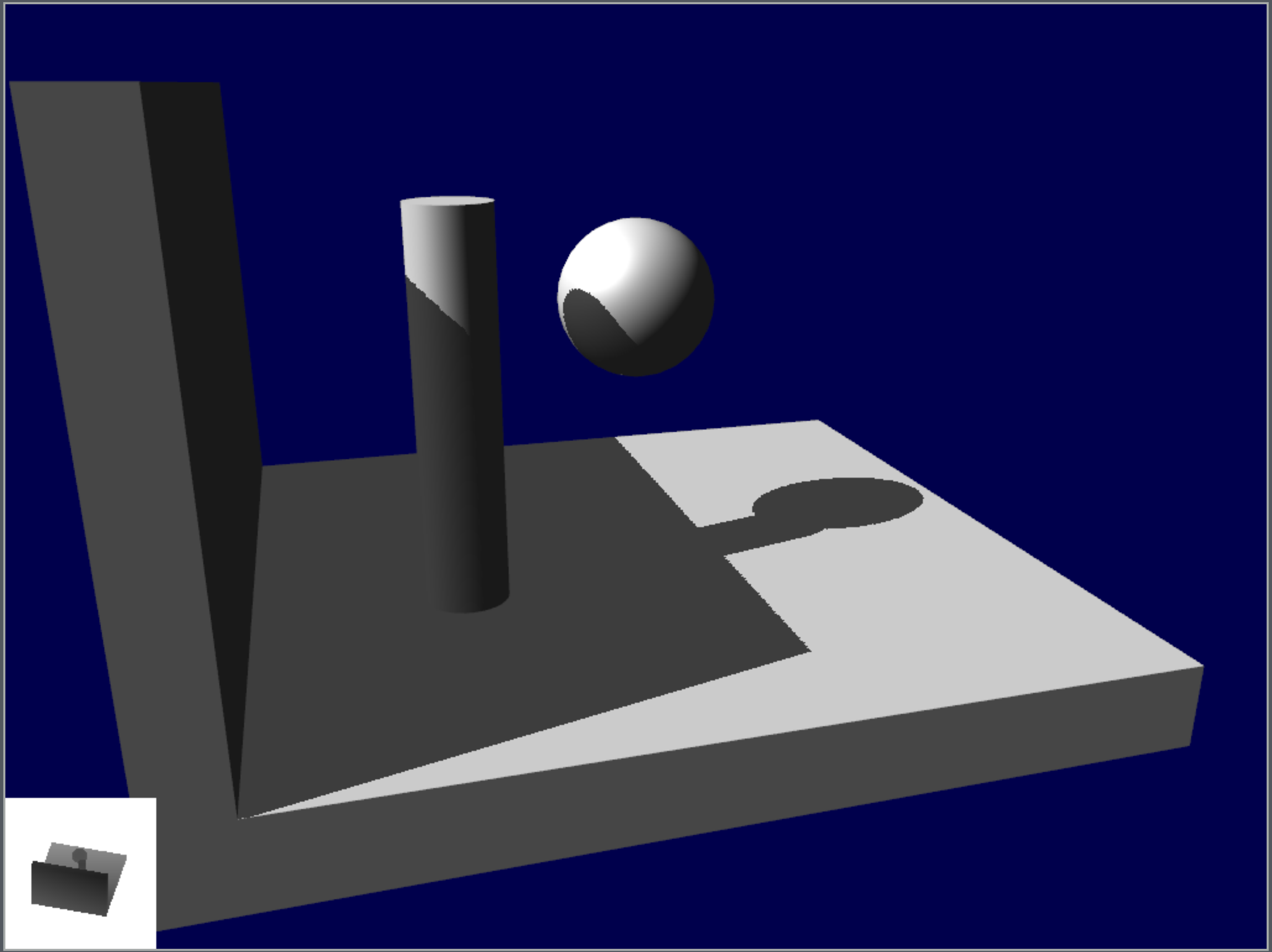
We want to filter the output, not the input, of the shadow test

- what fraction of samples pass the test
- samples pass the test if they are closer than the shadow map depth
- therefore “percentage closer filtering” or PCF

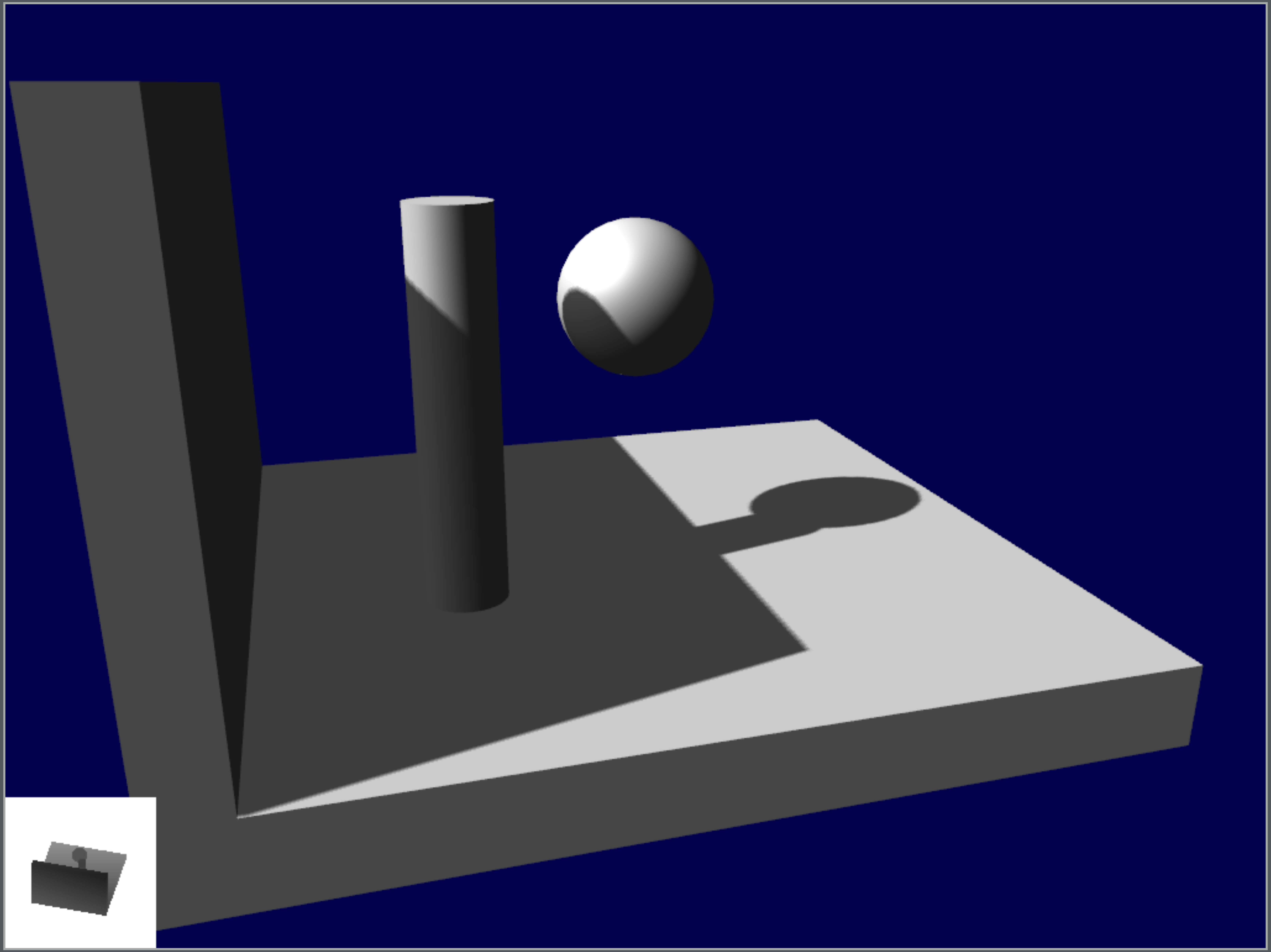
Percentage Closer Filtering

- Soften the shadow to decrease aliasing
  - Reeves, Salesin, Cook 87
  - GPU Gems, Chapter 11





closed surfaces and slope-dependent bias



adding percentage-closer filtering

Soft shadows from small sources

Main effect is to blur shadow boundaries

- PCF can do this
- …but how wide to make the filter?

Real shadows depend on area of light visible from surface

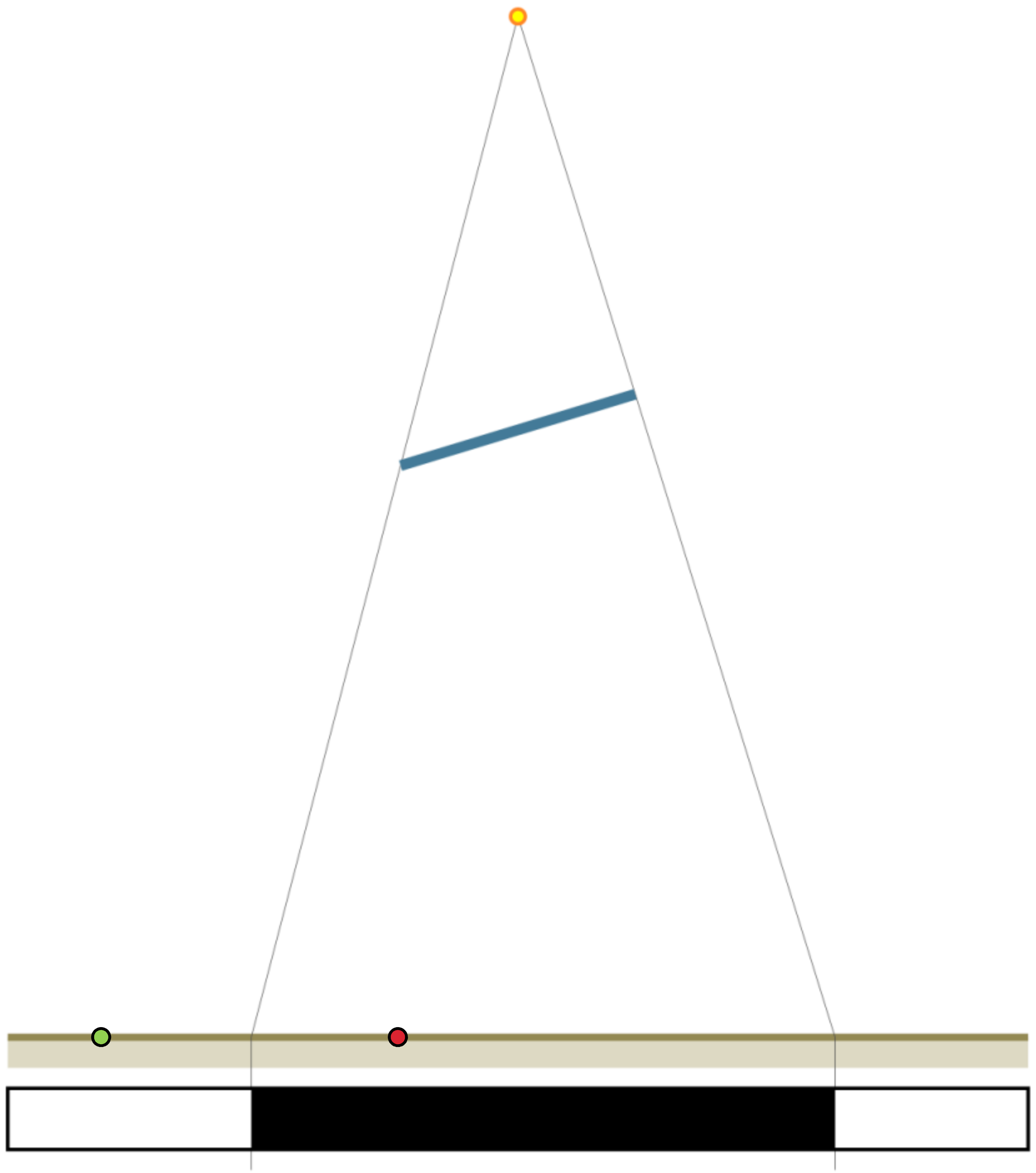
- this can vary in complex ways
- example: sun viewed through leafy trees

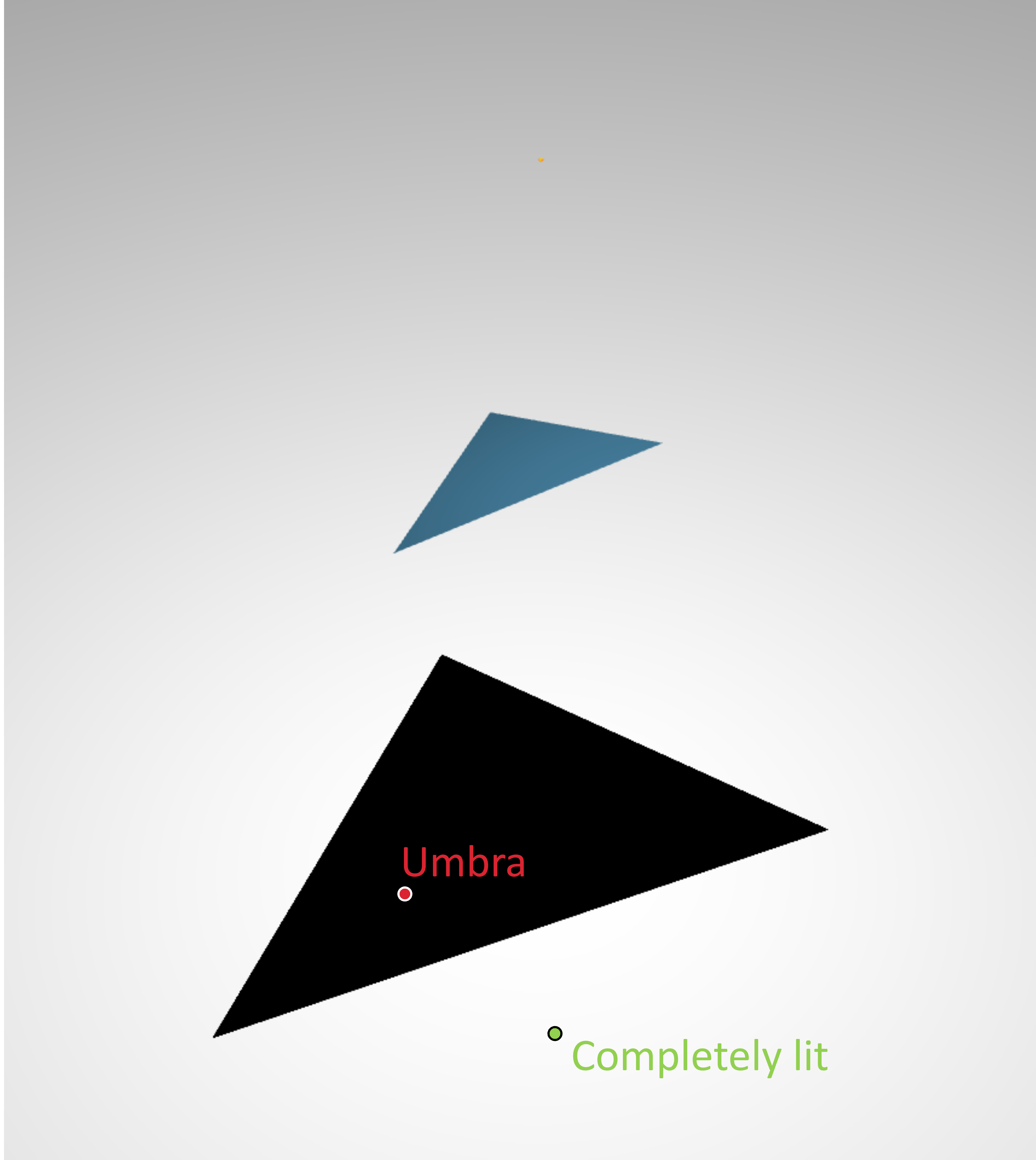
Useful approximation: convolution

- shadows are convolutions when the blocker and source are parallel and planar
- occluder fusion: approximating some occluding geometry as a planar blocker

Hard Shadows

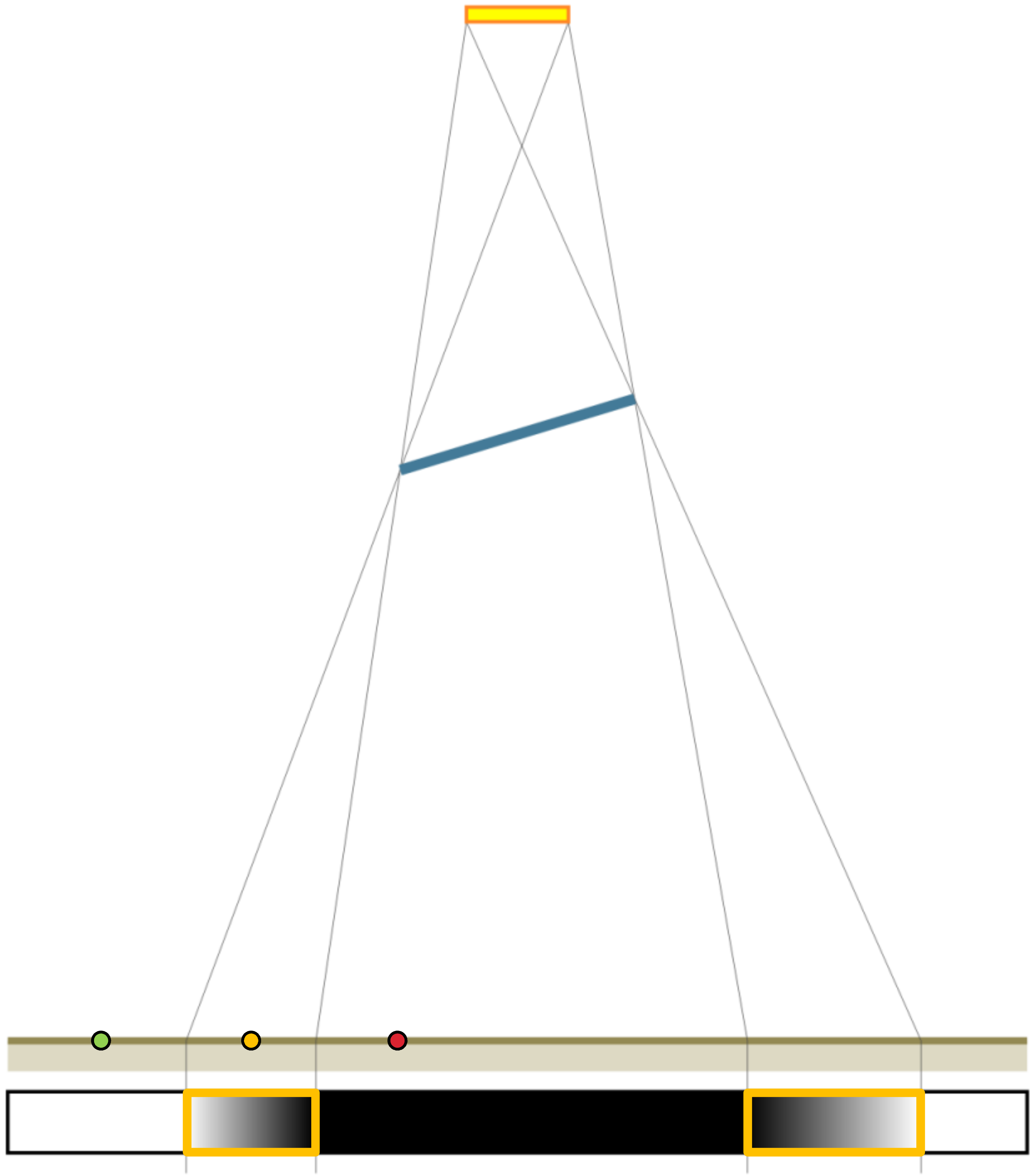
Michael Schwarz, SIGGRAPH 2013 Real Time Shadows course





Soft Shadows

Michael Schwarz, SIGGRAPH 2013 Real Time Shadows course

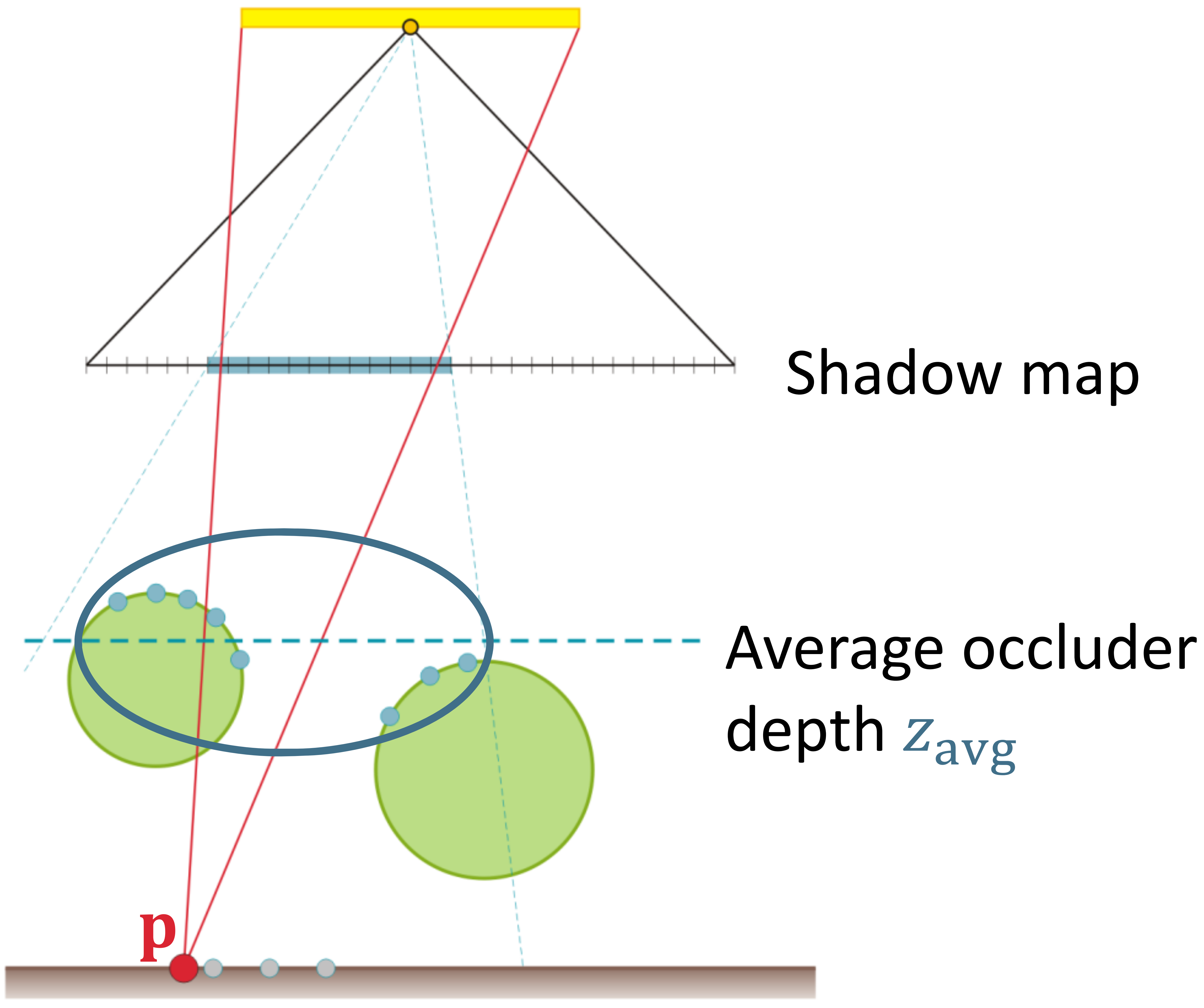


Shadow Hardening on Contact

Michael Schwarz, SIGGRAPH 2013 Real Time Shadows course

Percentage-Closer Soft Shadows

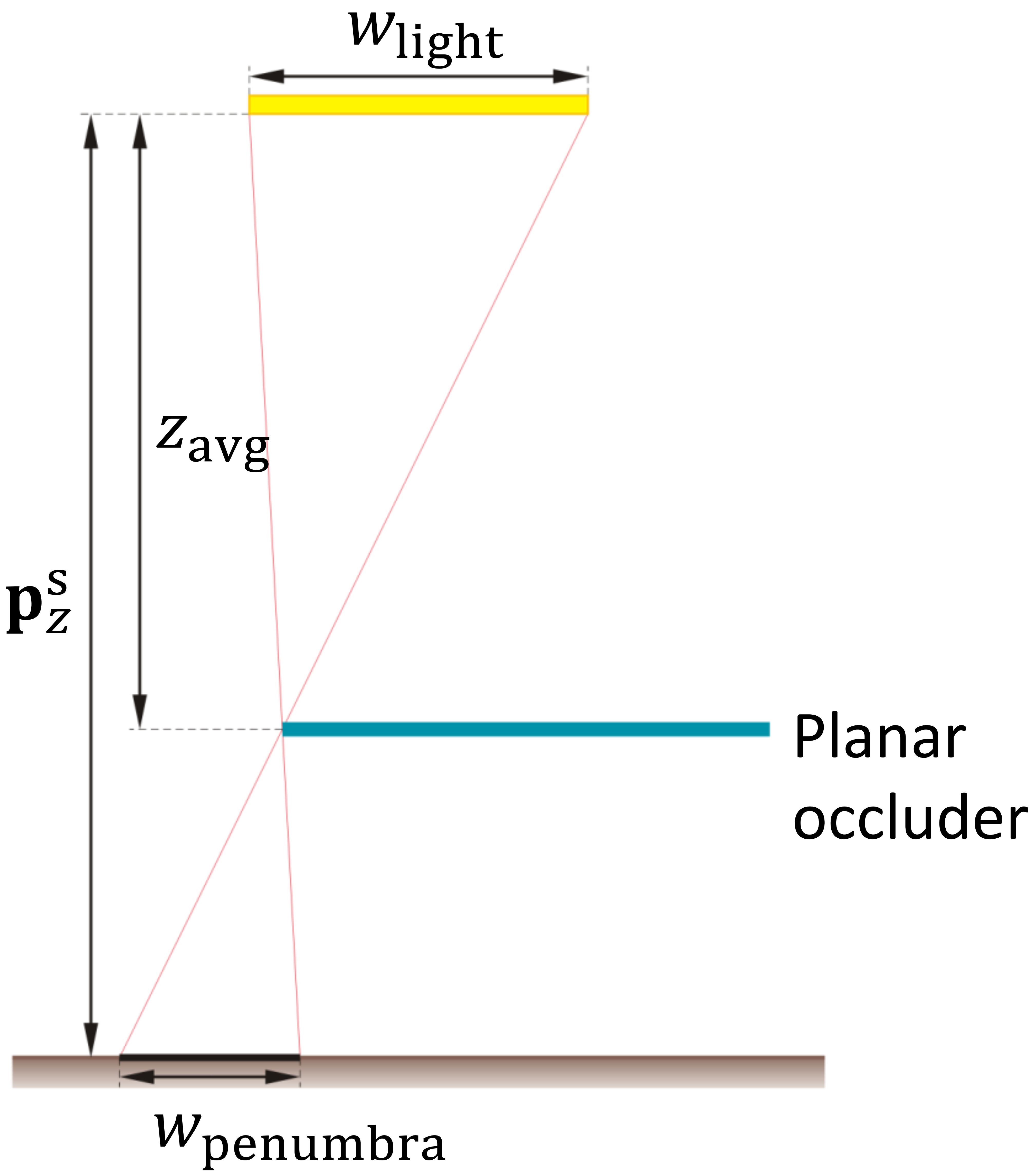
1. Blocker search



Percentage-Closer Soft Shadows

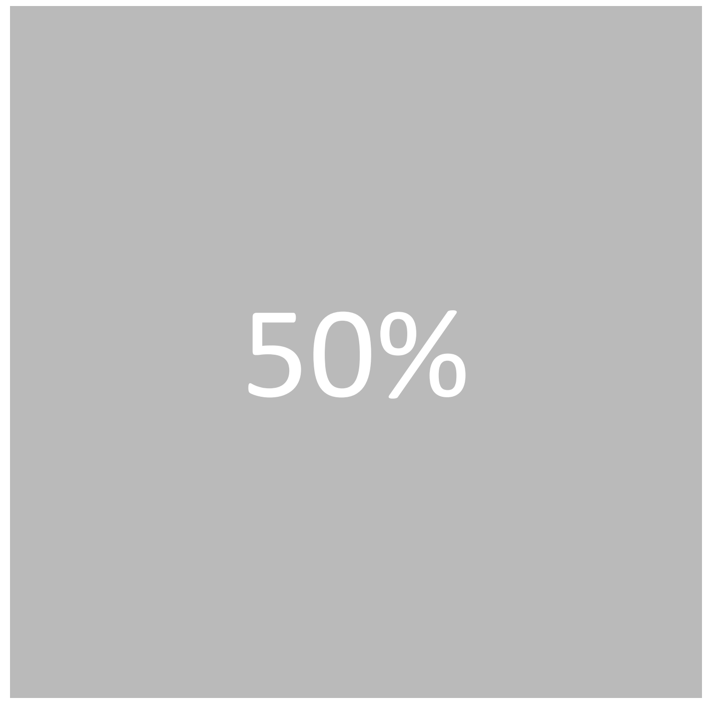
1. Blocker search
2. Penumbra width estimation

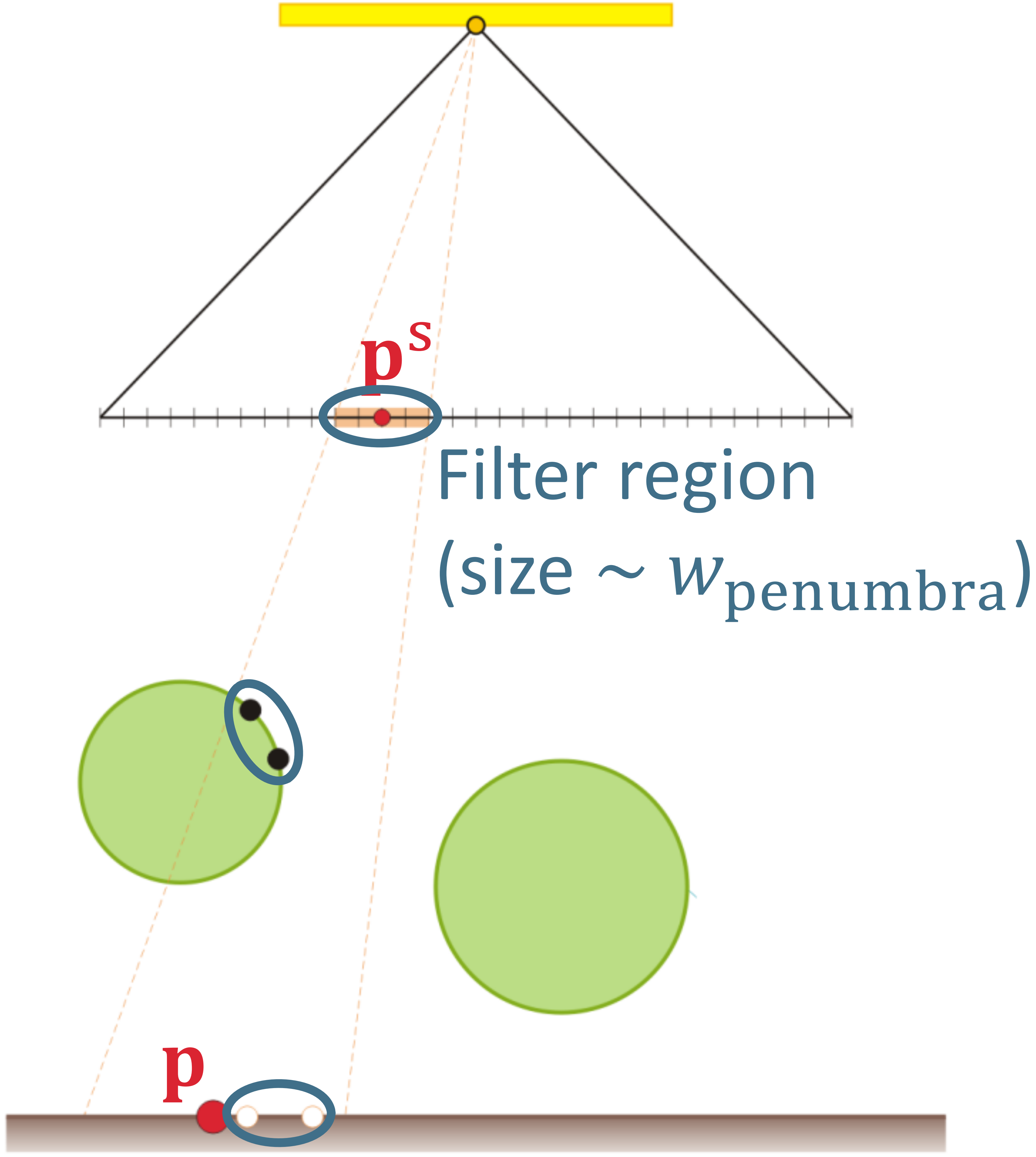
$$w_{\text{penumbra}} = \frac{\mathbf{p}_z^s - z_{\text{avg}}}{z_{\text{avg}}} w_{\text{light}}$$



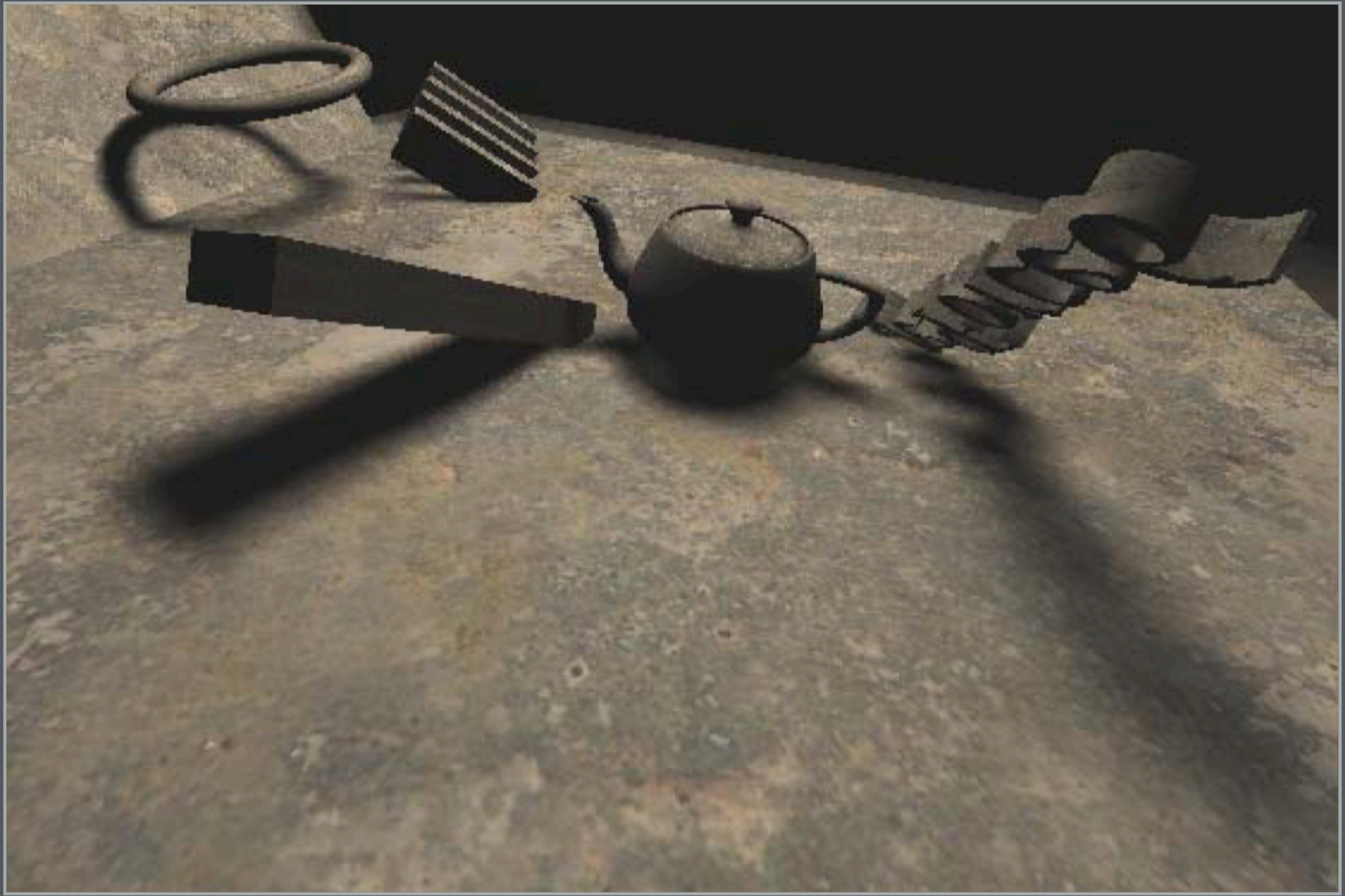
Percentage-Closer Soft Shadows

1. Blocker search
2. Penumbra width estimation
3. Filtering





Percentage-closer soft shadows



Fernando, NVidia whitepaper ~2005