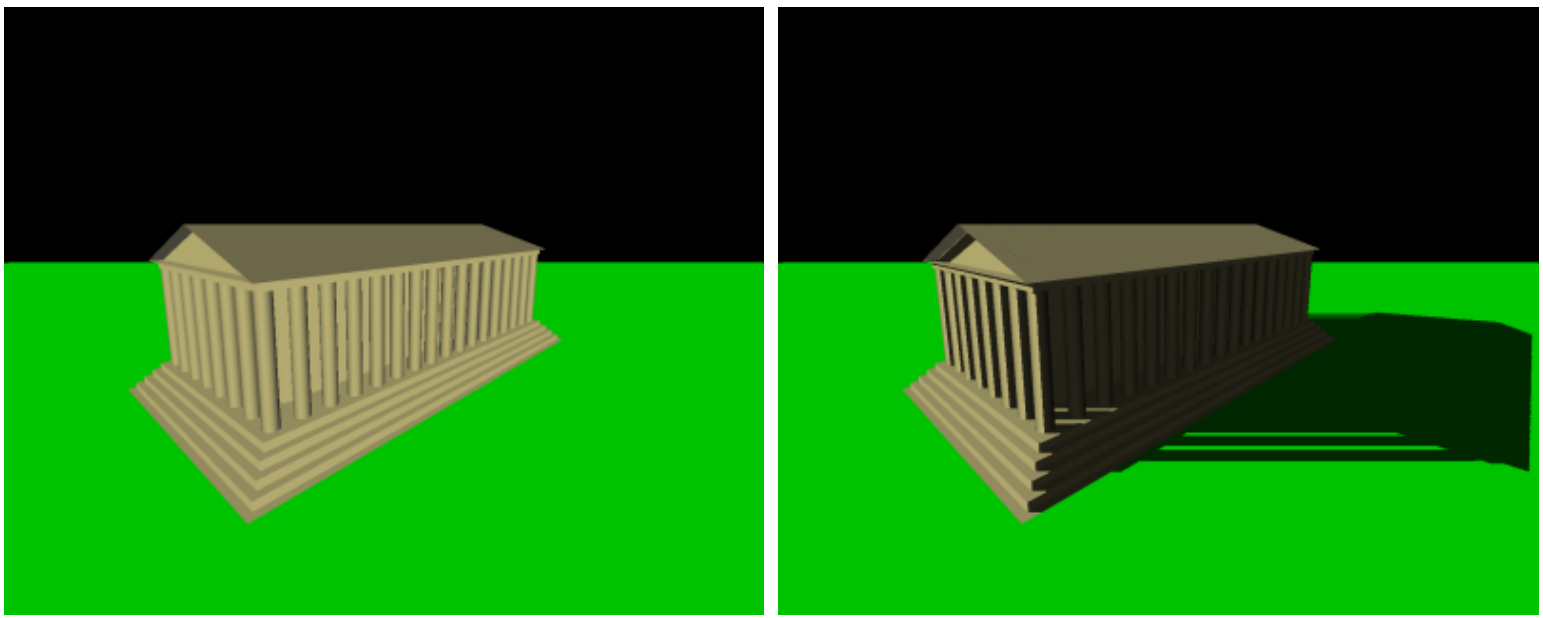
CS 461 - Computer Graphics

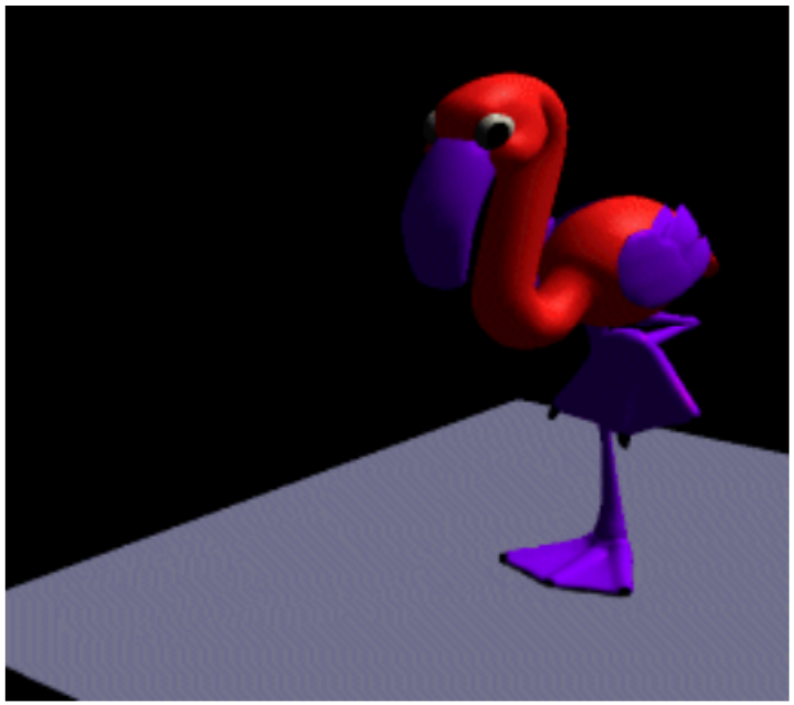
Shadows

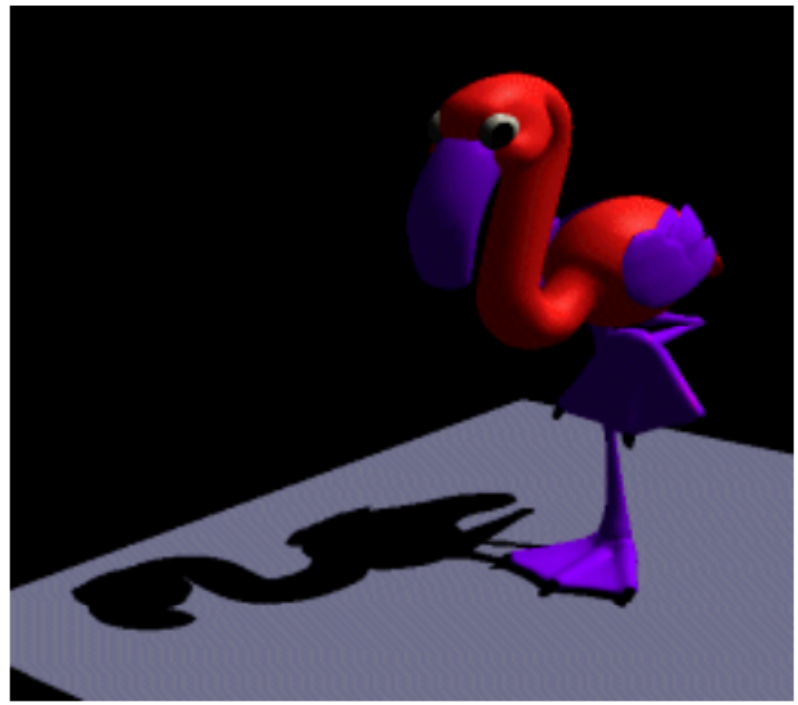
Shadows

- A shadow is a dark area where light from a light source is blocked by an opaque object

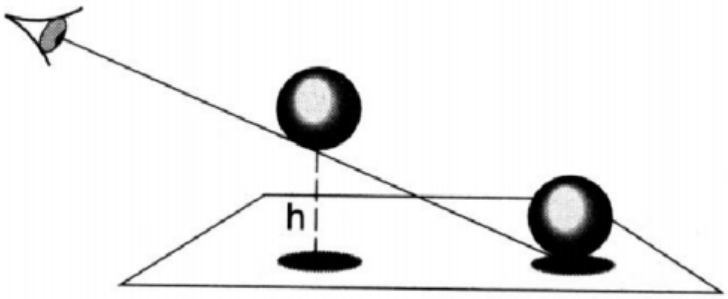


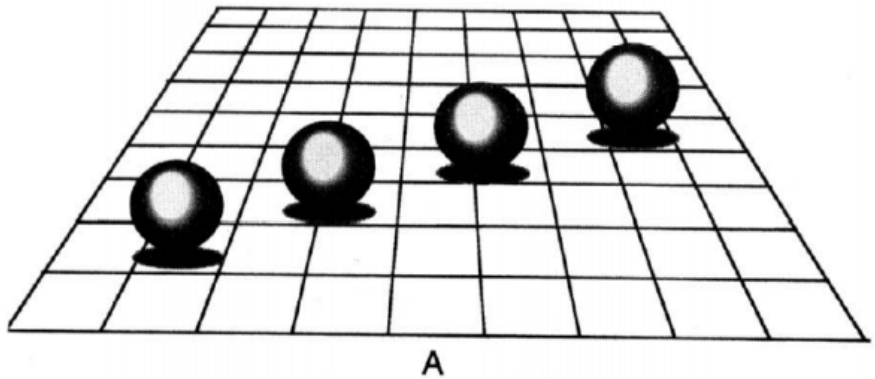
Shadow - Relevance



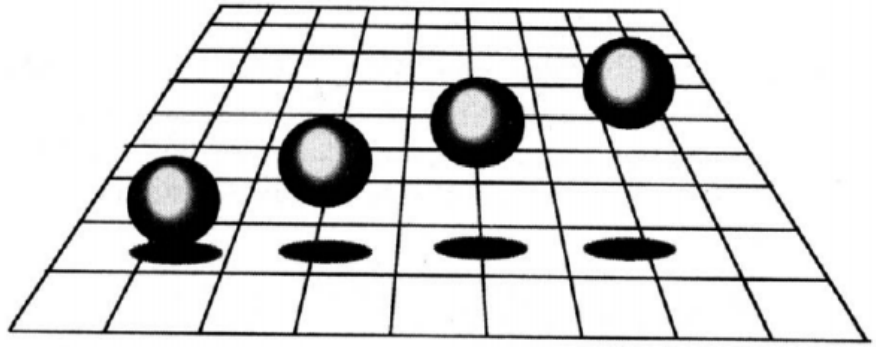


Shadow - Relevance

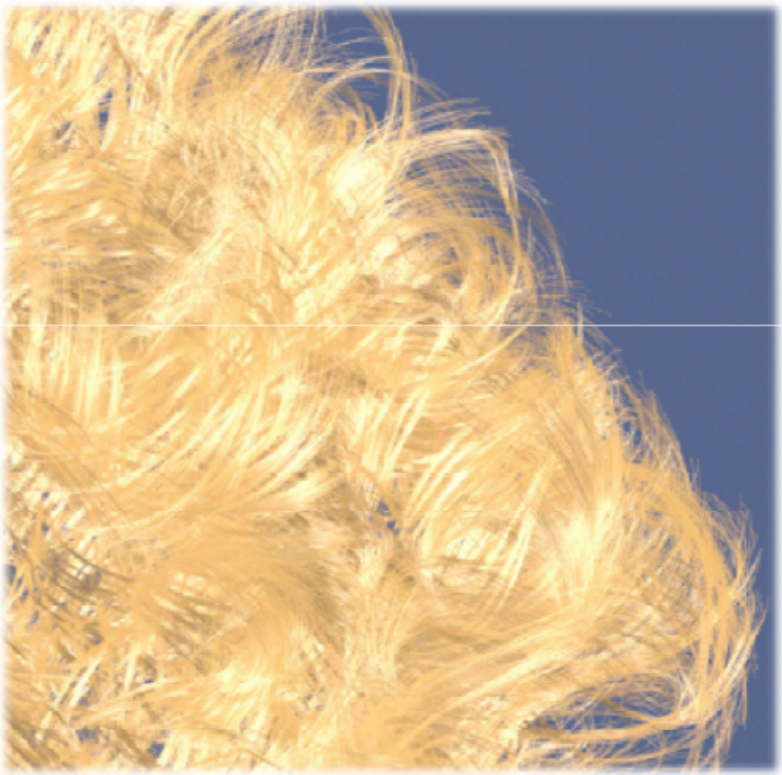




A



Shadow - Relevance



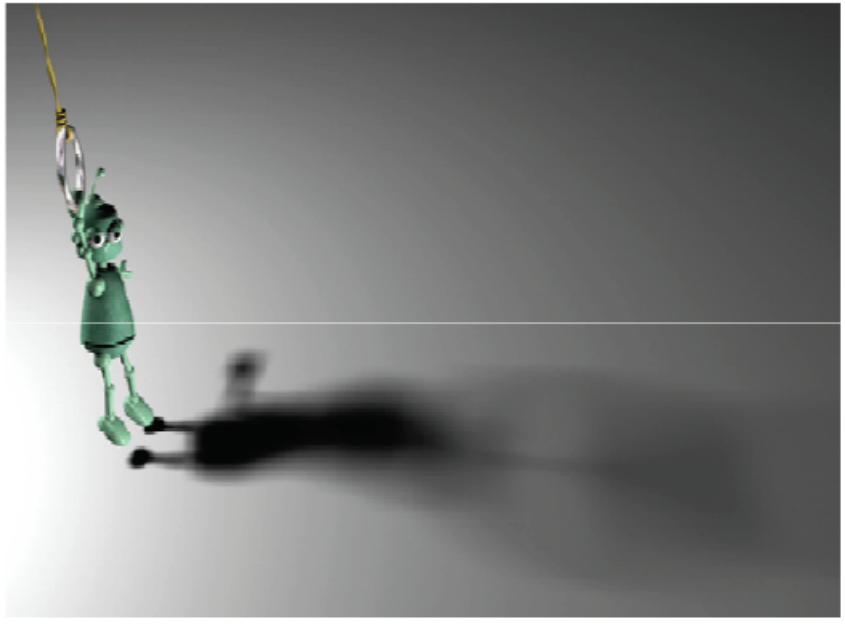
Without self-shadowing

With self-shadowing

Shadow - Relevance

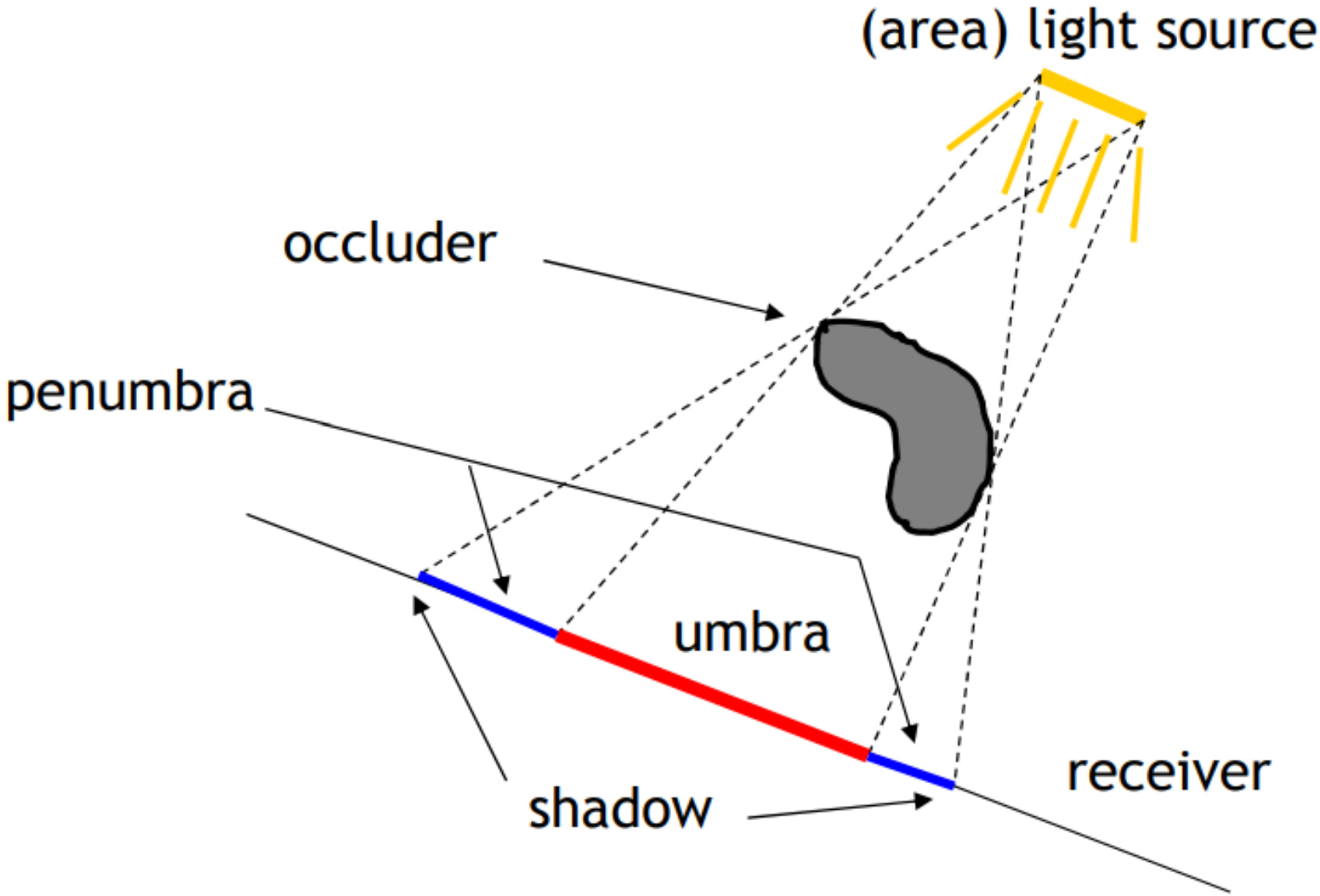
Hard and Soft shadows

Hard shadow from point light source



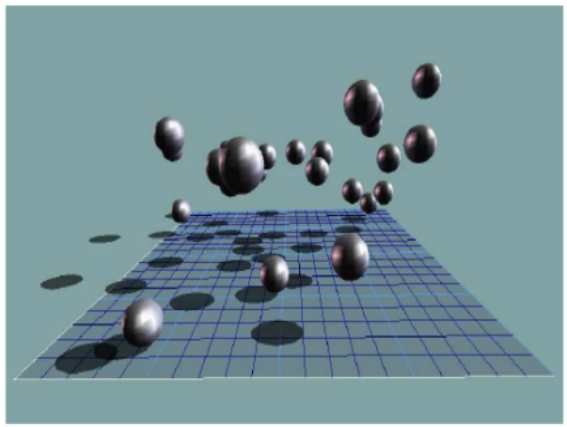
Soft shadow from area light source

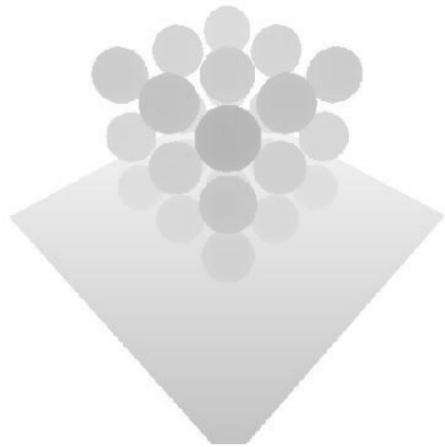
Umbra and Penumbra

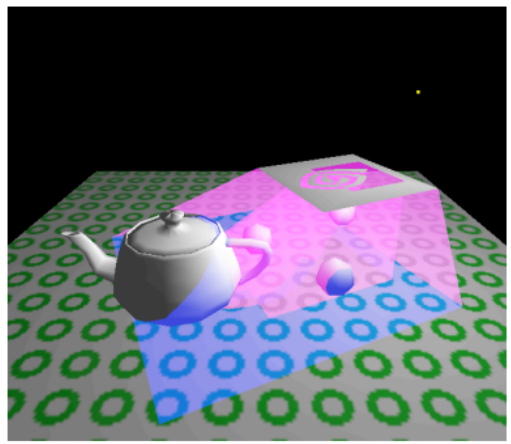


Shadows

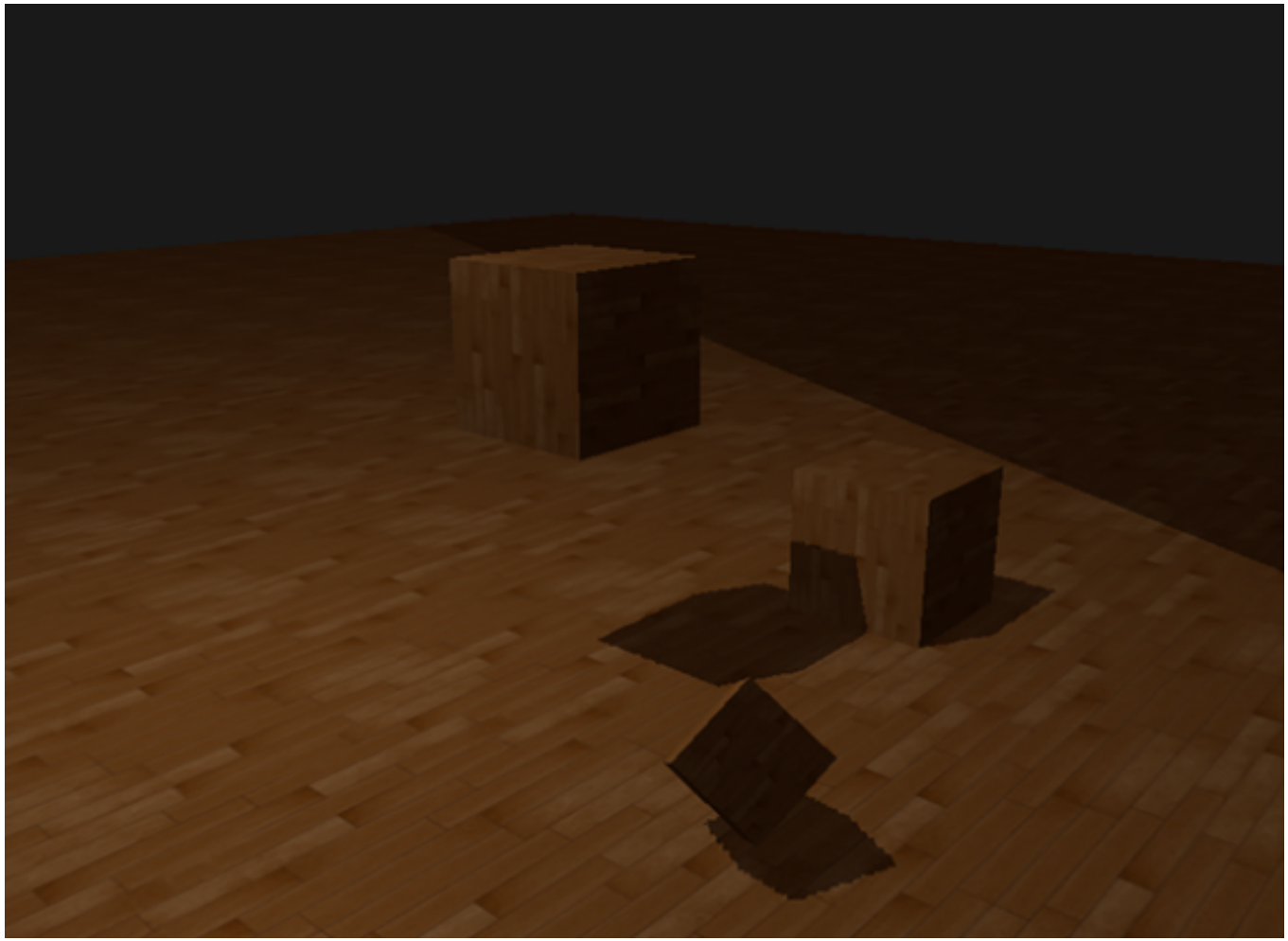
- Ground planes
- Shadow volumes
- Depth buffer shadows



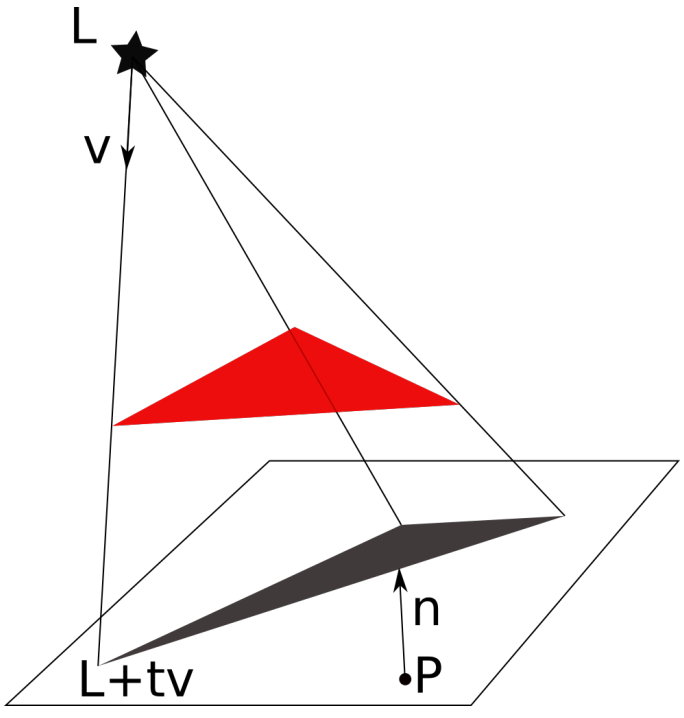




Ground planes

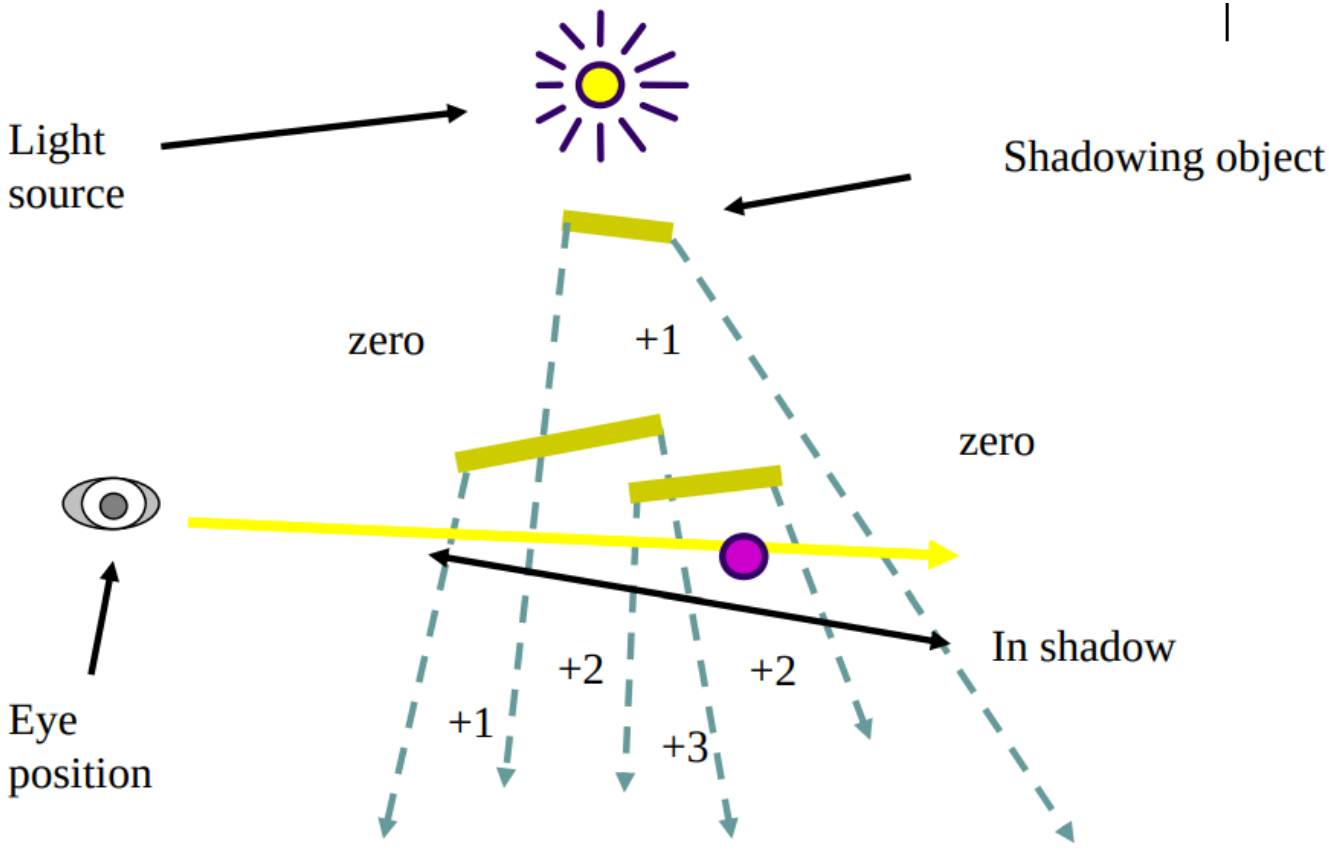


Ground planes

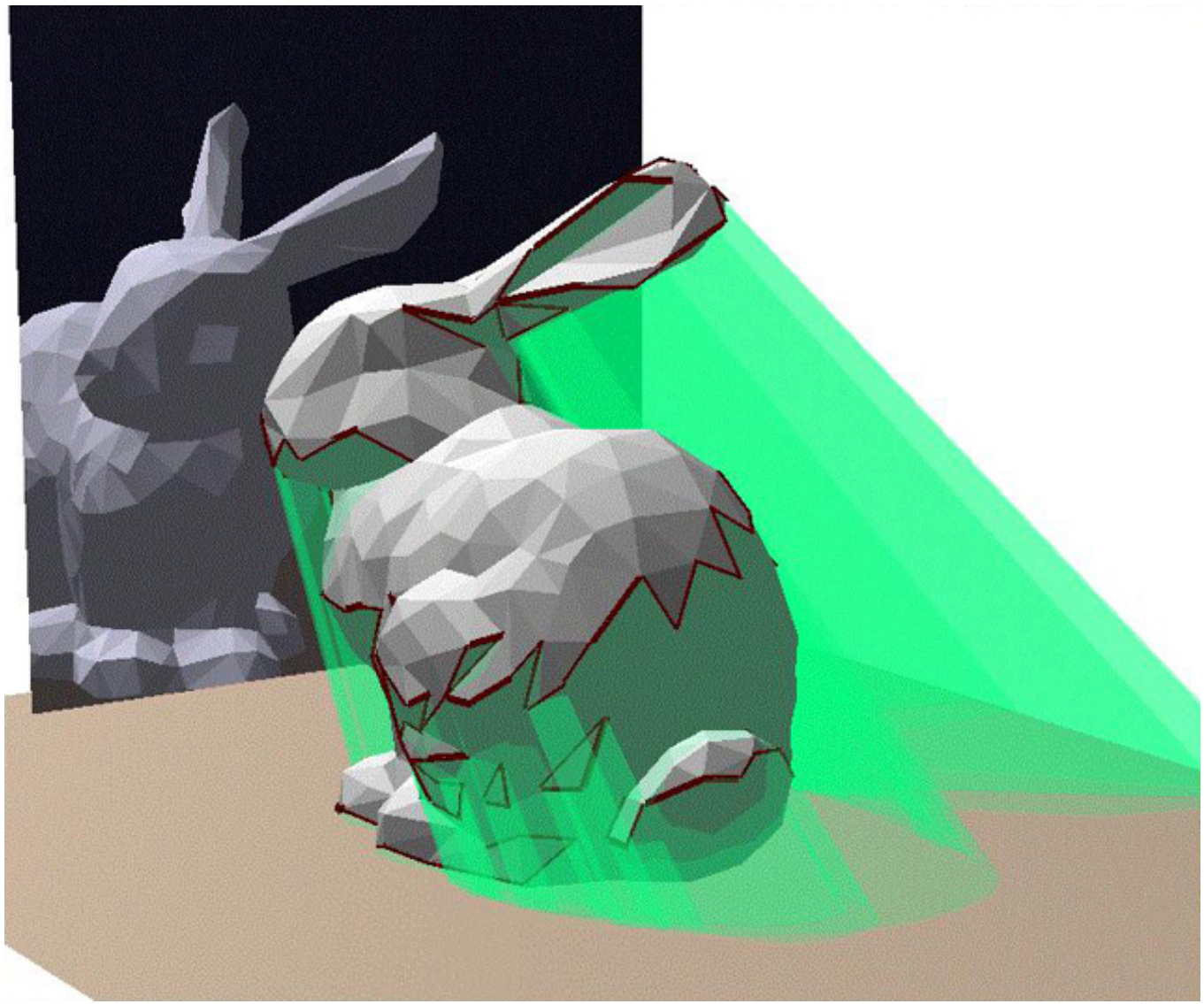


- $(L + t\vec{v} - P).\vec{n} = 0$
- $(L - P).\vec{n} + t\vec{v}.\vec{n} = 0$
- $t = -\frac{(L-P).\vec{n}}{\vec{v}.\vec{n}}$

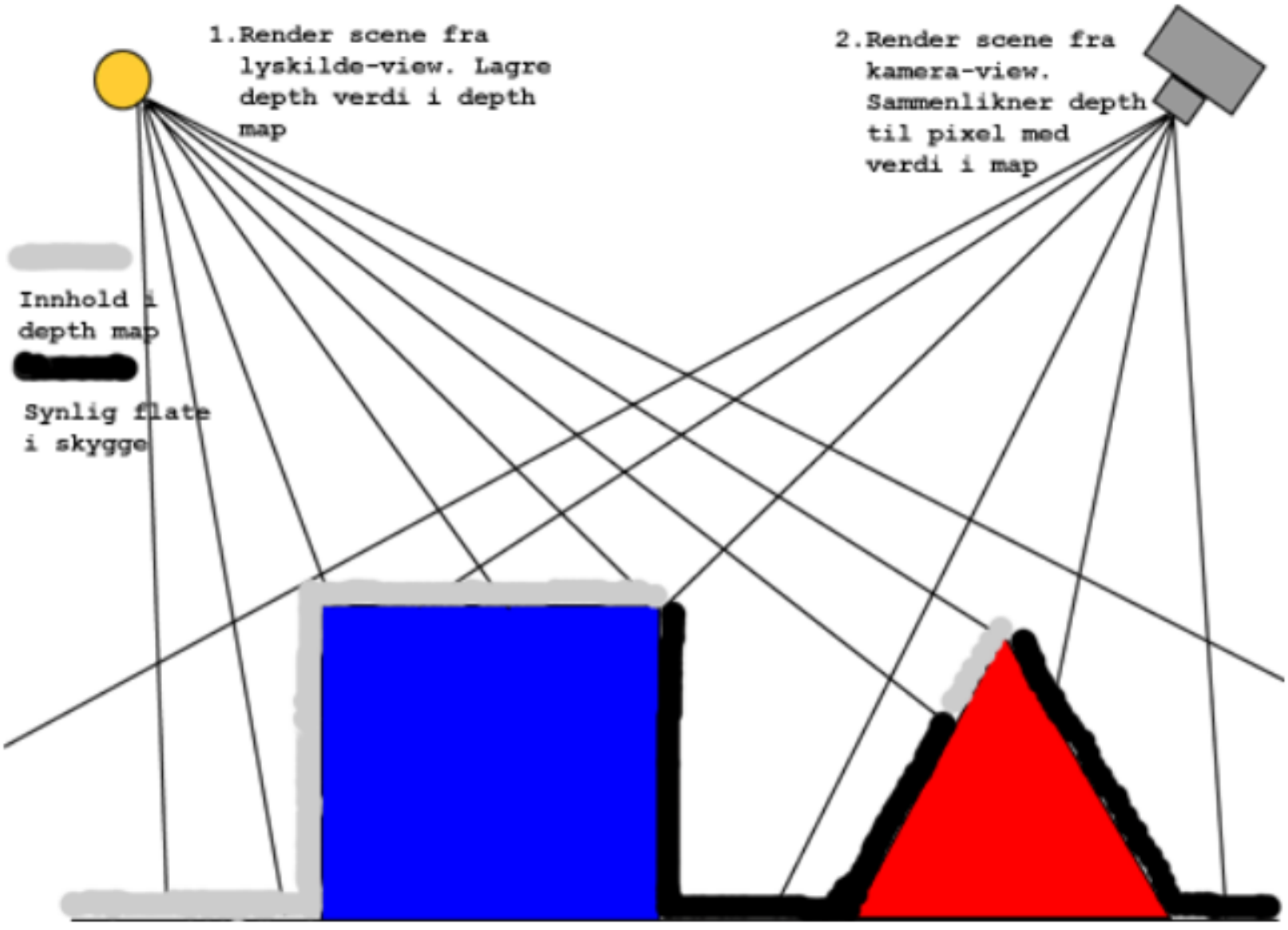
Shadow Volumes



Shadow Volumes



Depth buffer shadows



Depth buffer shadows - Algorithm

1. Render the scene using the light as the camera and perform z-buffering
2. Generate a light z buffer (called shadow map)
3. Render the scene using the regular camera, perform z-buffering, and run the following steps: (next slide)

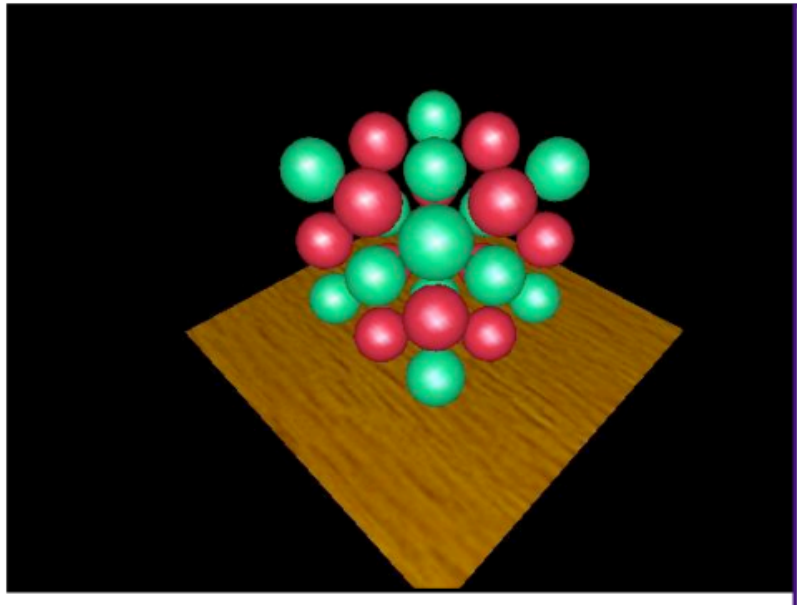
Depth buffer shadows - Algorithm

3.1 For each visible fragment with [x,y,z] in loccal space, perform a transformation to the light's clip space (light as the eye) [x1,y1,z1]

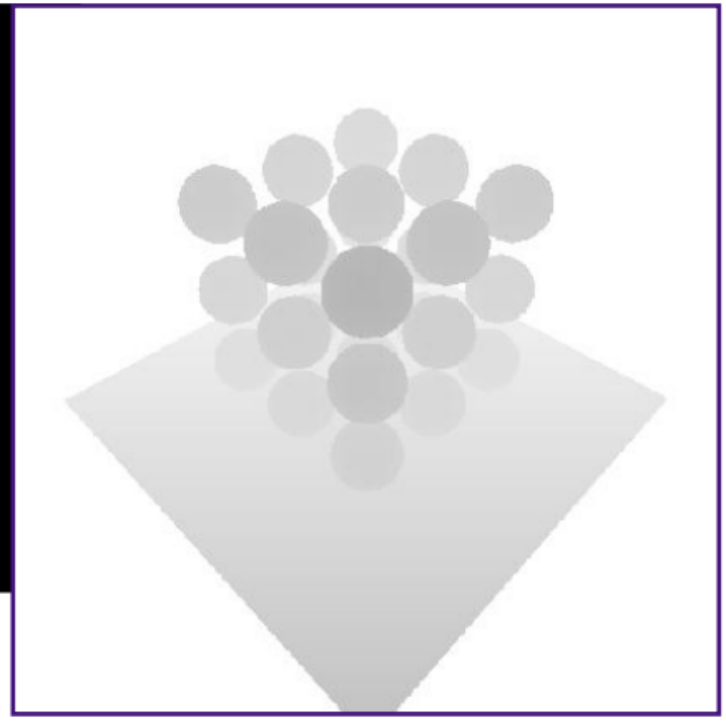
3.2 Compare z1 with z = shadow_map[x1,y1]

- If z1 <=z (closer to light), then the pixel in question is not in shadow; otherwise the fragment is shadowed

Depth buffer shadows - Example

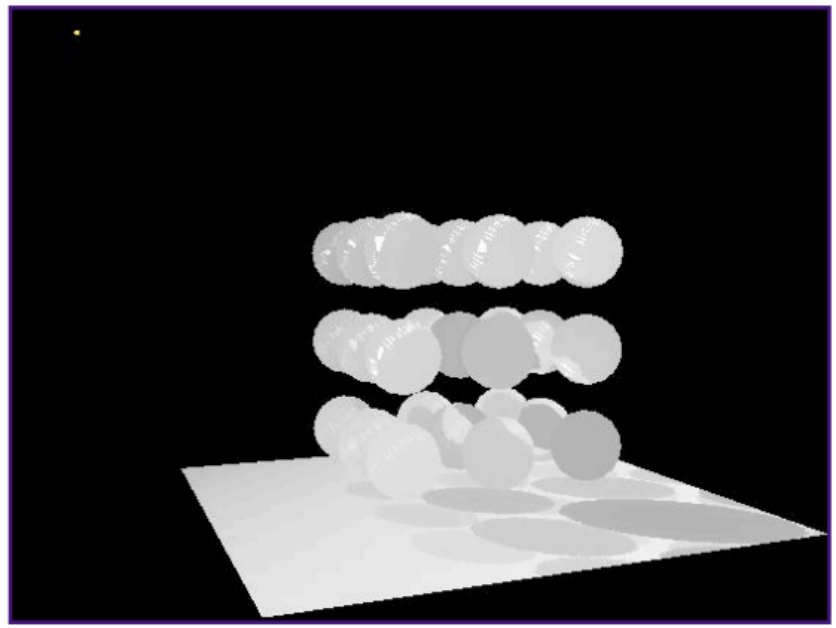


View from light



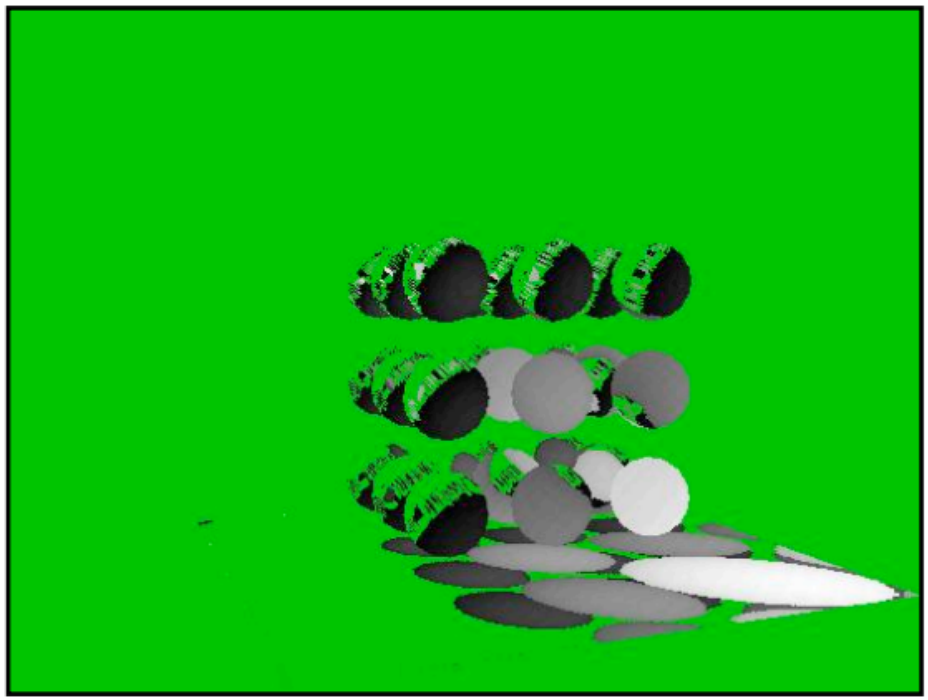
Depth Buffer (shadow map)

Depth buffer shadows - Example

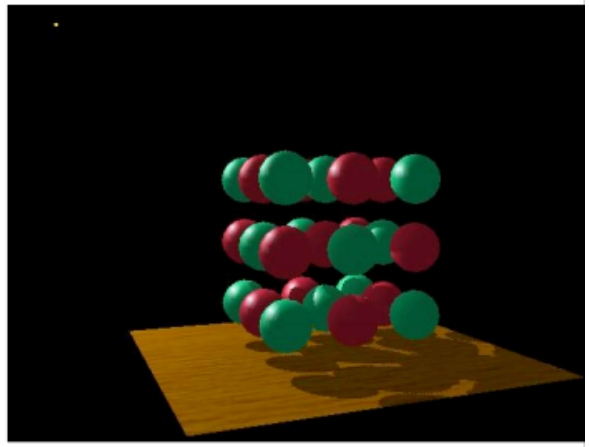


Visible surface depth

Depth buffer shadows - Example



Non-green in shadow



Final Image

Shadow - simplification

Fake shadows

Next class

- 22^{nd} October 9-10
- Topic: Texturing
- Reminder - Topics should be finalized one week before your seminar date