

5CM507 Graphics

Lecture 09 Shadow Maps

Dr Youbing Zhao

November 24, 2025

Last Week



- ▶ Normal mapping
- ▶ Cube Maps
- ▶ PBR Maps
- ▶ ...

Pre-session Readings

Shadows in painting

Light is connected with shadows

Shadows can

- ▶ Add depth and perspective
- ▶ Enhance realism
- ▶ Create contrast
(chiaroscuro) in artworks
and photography

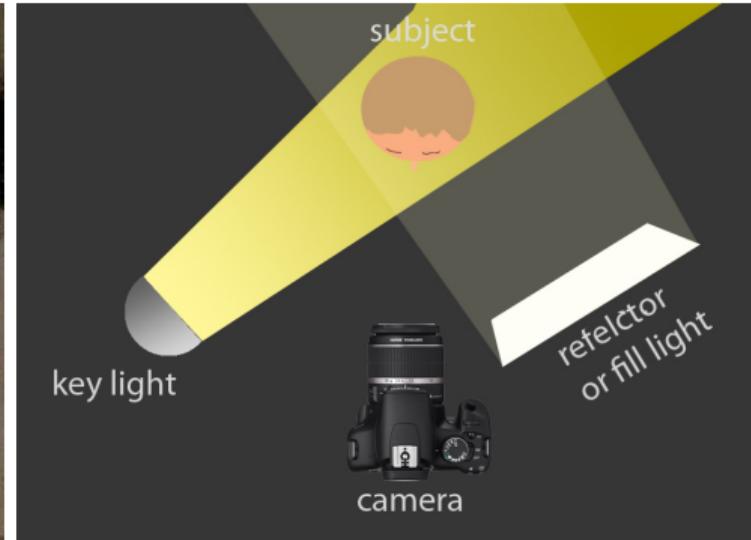


Supper at Emmaus, Caravaggio, 1601, National Gallery, London

Rembrandt Lighting

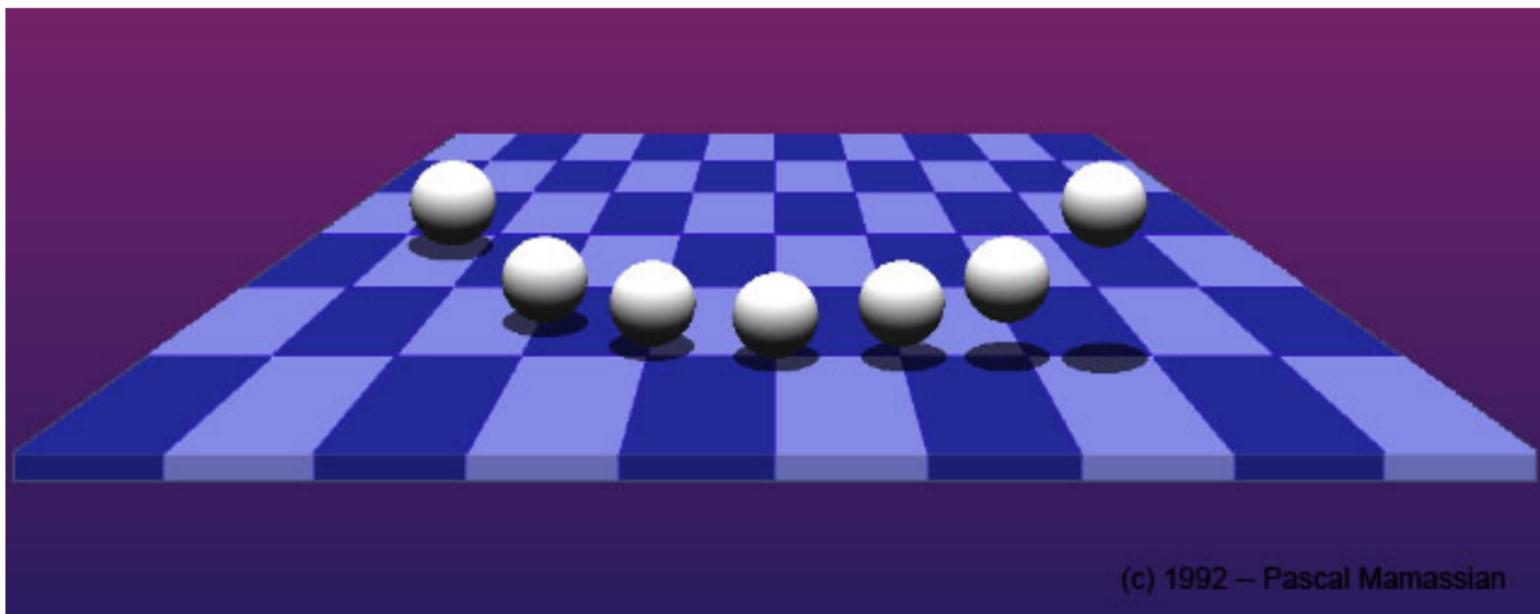


The Rembrandt patch in Portrait of Johannes Wtenbogaert, 1633



Rembrandt light setup

Shadows Provide Position and Depth Cues



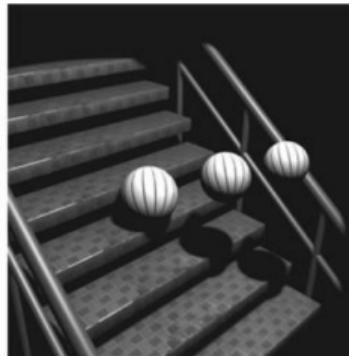
(c) 1992 – Pascal Mamassian

Shadows provides depth cues

Shadows Provide Position and Depth Cues



(a)



(b)



(c)



(d)

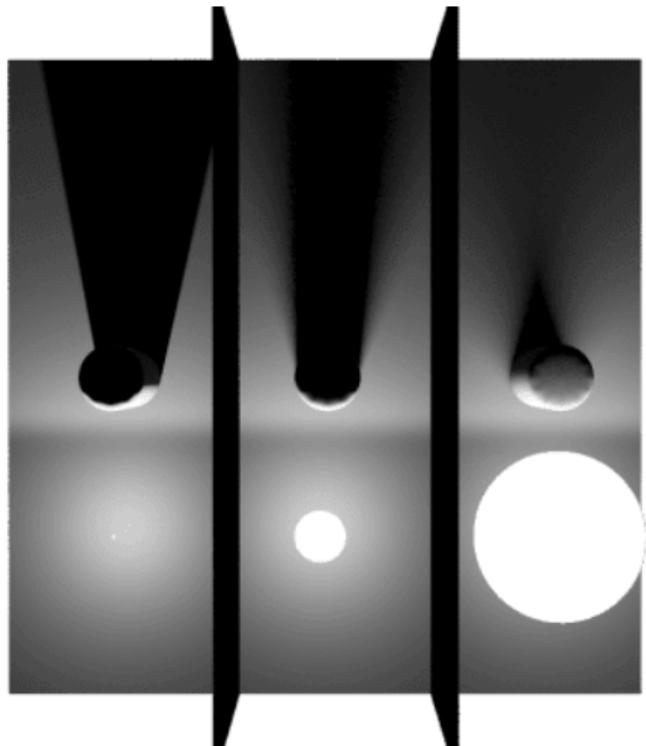
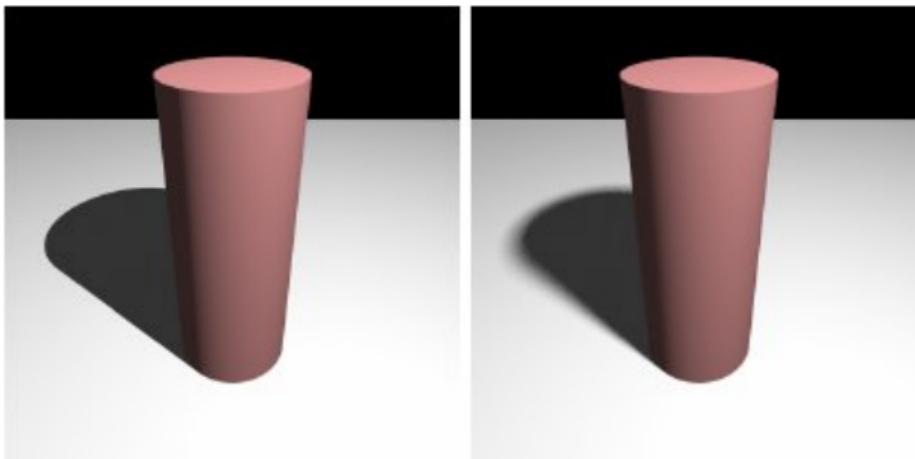


(e)

Shadows provides depth cues

Shadow Types

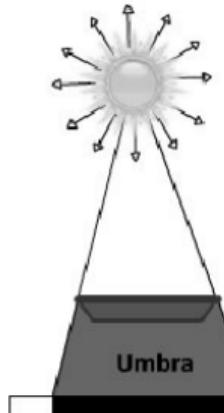
- ▶ Hard light example: direct sun light
- ▶ Soft light example: lamp with shade



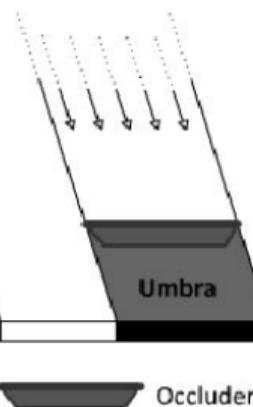
Hard and soft shadows

Shadow Types

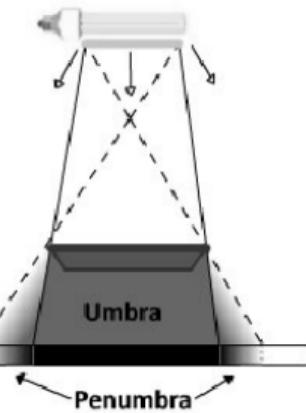
a) Point light



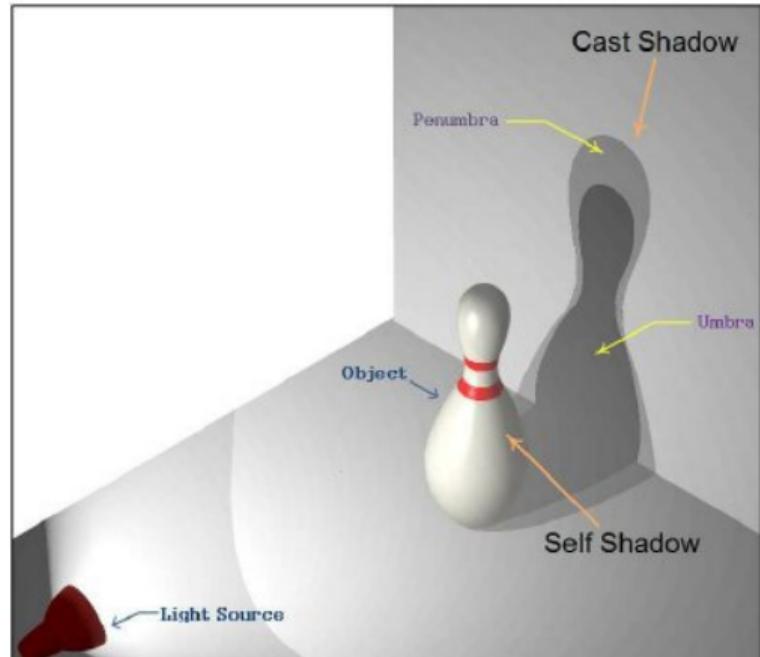
b) Parallel light



c) Area light



Umbra and Penumbra



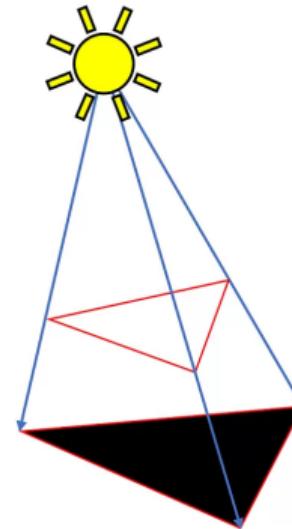
Shadow types

Projective Fake Shadows in Early Games

- ▶ Use a fake shadow polygon
- ▶ Shadows are totally opaque
- ▶ Project the polygon from the light source to the ground.



Projective shadows in Tomb Raider 1996



Lecture

Contents

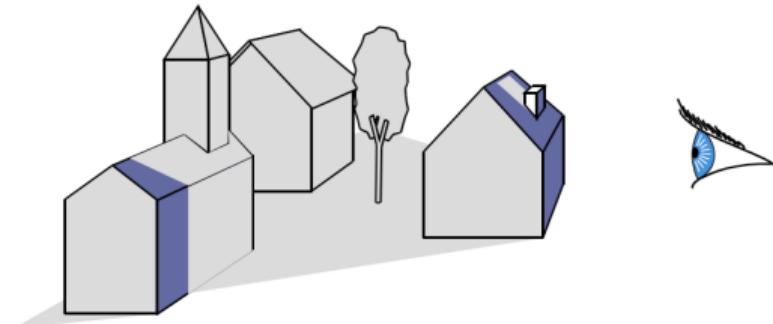
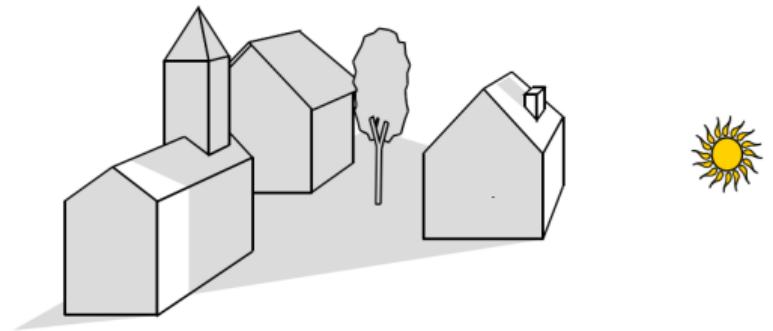


- ▶ Shadow Maps
- ▶ Shadow Map Artefacts
- ▶ Ambient Occlusion

Shadow Projection vs Perspective Projection

Analogy between lighting and viewing:

- ▶ A point light source : a camera
- ▶ Both cast “rays”
- ▶ Shadows: not “visible” from the light source
- ▶ Shadow computation: can make use of depth in perspective projection



Projection from a point light source vs perspective projection

Perspective Projection Revisited

$$v_{clip} = \mathbf{PVM}v \quad \mathbf{M} = \mathbf{TRS}$$

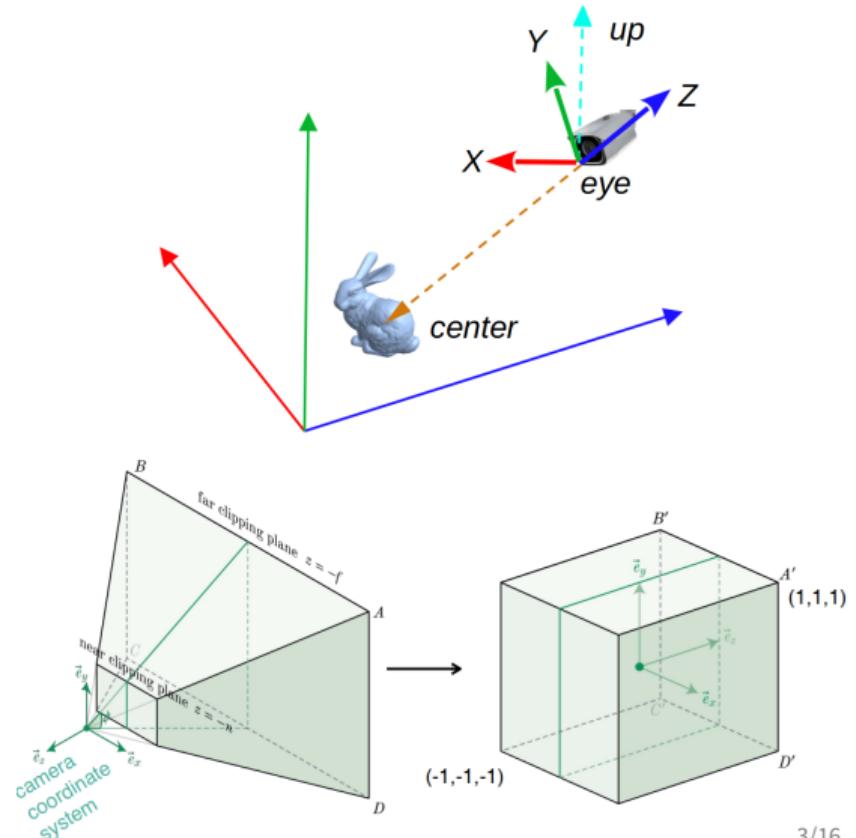
$$\mathbf{V} = \begin{bmatrix} {}^wX_c^x & {}^wX_c^y & {}^wX_c^z & -\vec{e} \cdot {}^wX_c \\ {}^wY_c^x & {}^wY_c^y & {}^wY_c^z & -\vec{e} \cdot {}^wY_c \\ {}^wZ_c^x & {}^wZ_c^y & {}^wZ_c^z & -\vec{e} \cdot {}^wZ_c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} \frac{2n}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2n}{t-b} & 0 & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

$$v_{clip} = [x_{clip} \quad y_{clip} \quad z_{clip} \quad w_{clip} = -z]^T$$

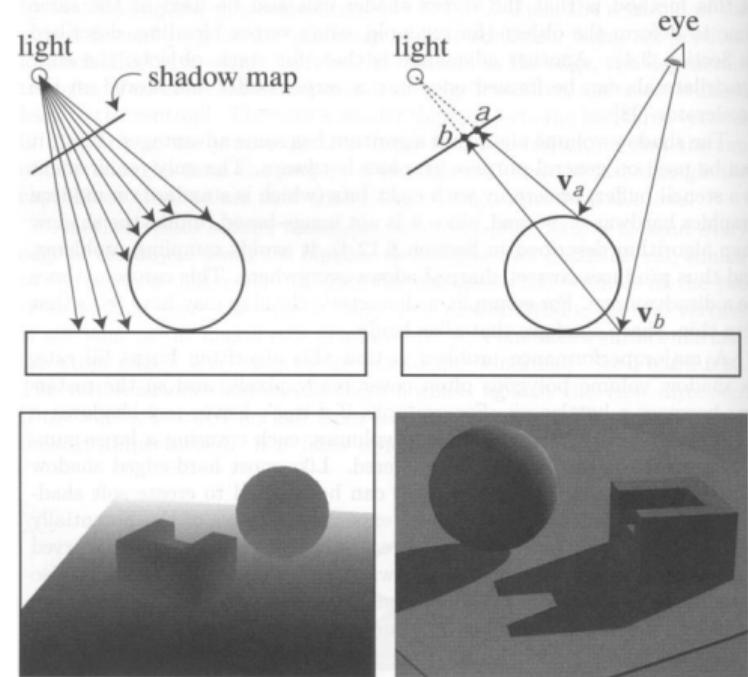
$$\text{OpenGL: } z_{norm} = -\frac{z'}{z} \in [-1, 1]$$

$$depth_{z\text{-buffer}} = \frac{z_{norm}+1}{2} \in [0, 1]$$



Shadow Maps

- ▶ Limitations
 - ▶ Single light source
 - ▶ Hard shadows
- ▶ Shadow Map: a depth map from a light source
 - ▶ closest scene depth from light source
 - ▶ like z-buffer, depth range : $[\underbrace{0}_{\text{near}}, \underbrace{1}_{\text{far}}]$
- ▶ Shadowing : if a point is “visible” by the light source
 - ▶ In shadow: $z > d_{map}$, behind lighted surfaces
 - ▶ Not in shadow: $z \leq d_{map}$

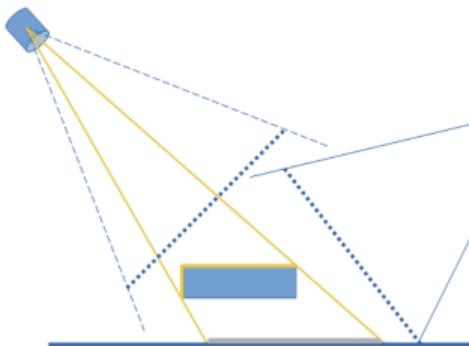


[Foley et al. “Computer Graphics Principles and Practice”]

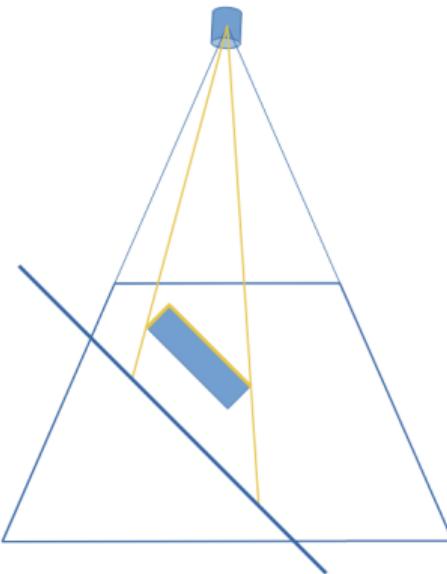
Shadow Map Generation: Pass I - Light-space Depth

Take the light source as the camera, perform offscreen rendering;

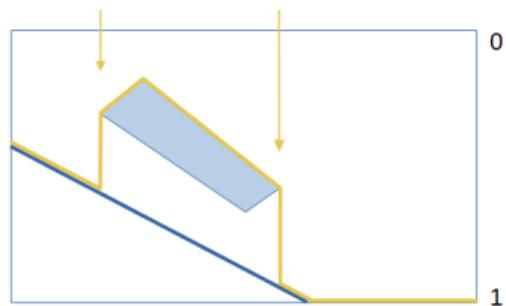
Save the depth buffer as a texture.



Top view of the scene



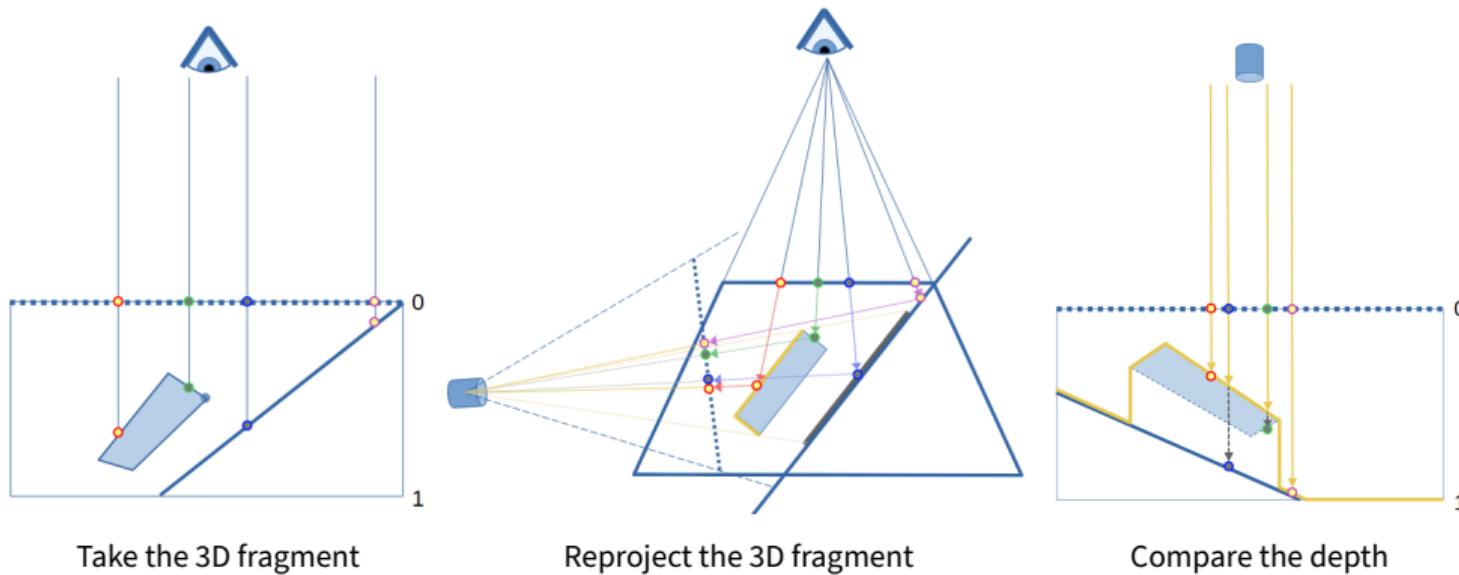
Projection from the light source



Depth map

Shadow Generation: Pass II - World Space

- ▶ Project the vertex in both the world space and the light space in the vertex shader
- ▶ Compare the lightspace fragment depth with the shadow map in the fragment shader



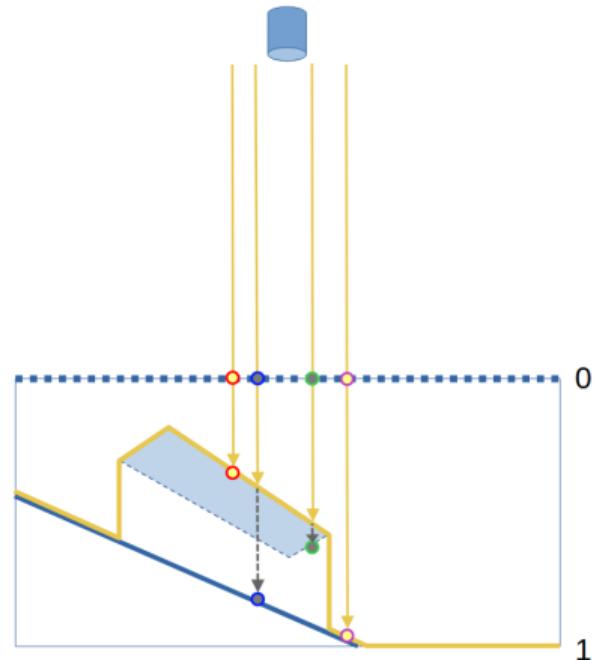
Shadow Generation: Pass II - Depth Comparison

Lightspace fragment coordinate: (x', y', z')

Larger $z' \iff$ Further from the light source

Compare the depth z' of the fragment in light space with the depth stored in the shadow map at (x', y')

- ▶ $z' > \text{ShadowMap}(x', y')$
 - ▶ Further from lighted surfaces \Rightarrow In shadow
- ▶ $z' \leq \text{ShadowMap}(x', y')$
 - ▶ On the lighted surface \Rightarrow Not in shadow



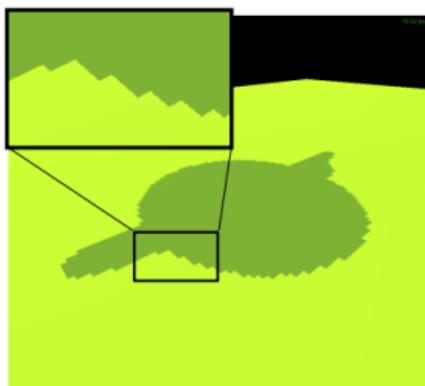
Shadow Map Artifacts



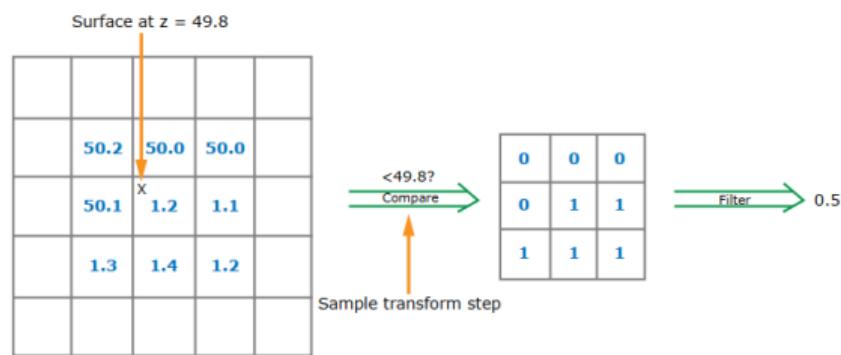
- ▶ Aliasing
- ▶ Surface Acne
- ▶ Peter Panning

Shadow Aliasing Solutions

1. Increasing shadow map resolution;
2. Percentage-Closer Filtering (PCF): Filter the result of the shadow test (using weighted average of comparison results for shadow blending, can also be used for generating soft shadows)



Jagged shadow edges

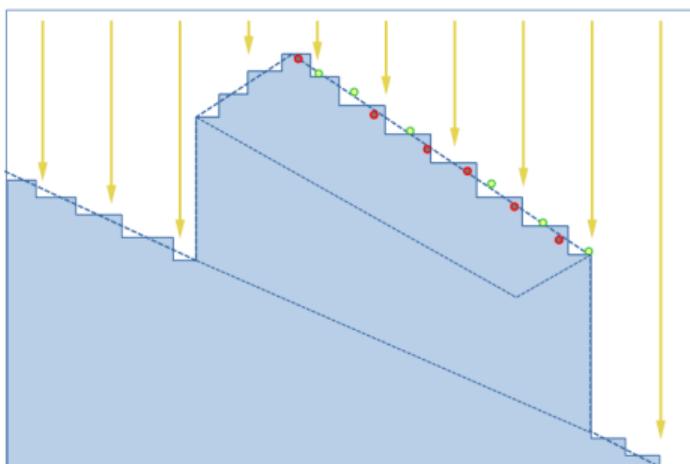


Smoothed Shadow Boundary

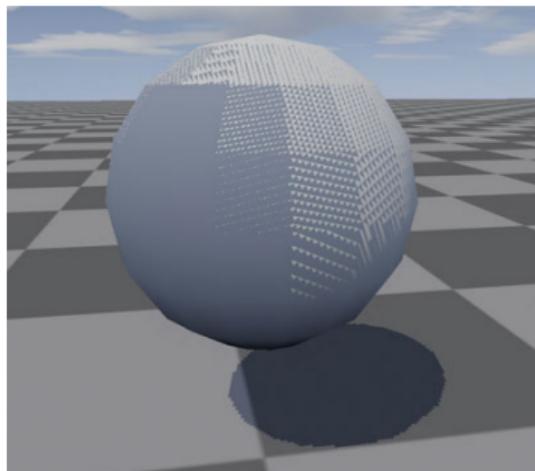
Shadow Map Precision Artefacts (Shadow Acne)

Visible points taken as shadows due to z-fighting, showing speckles and streaks.

Solutions: 1. Depth offsetting; 2. Front face culling.



Shadow map depth precision problem

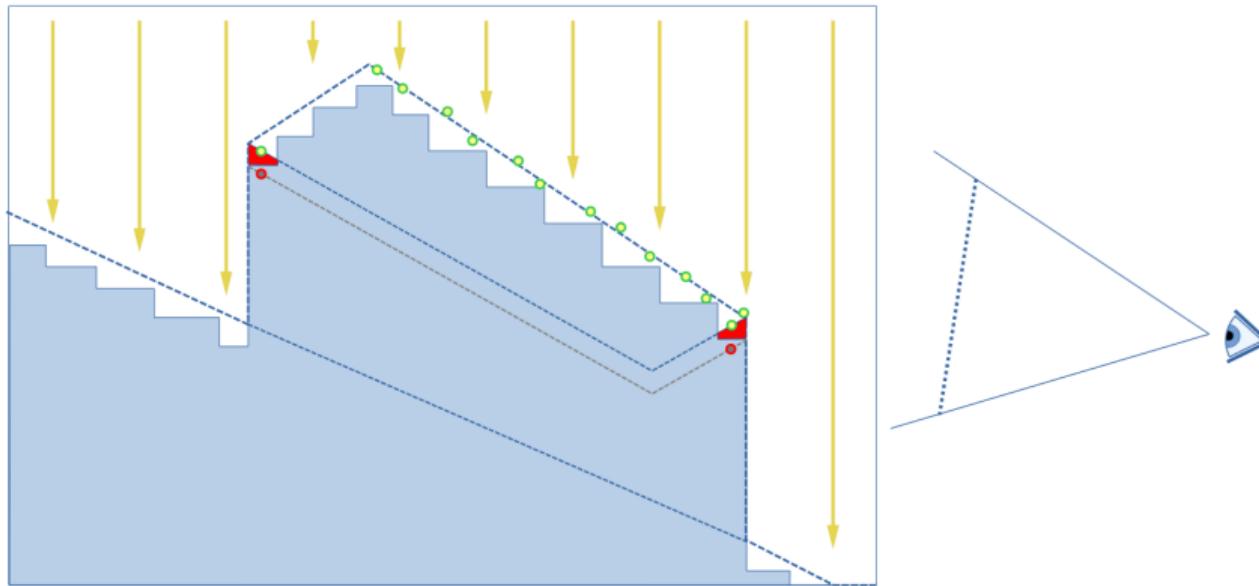


Shadow acne

Using Depth Bias

Subtract the depth of the point by a small bias.

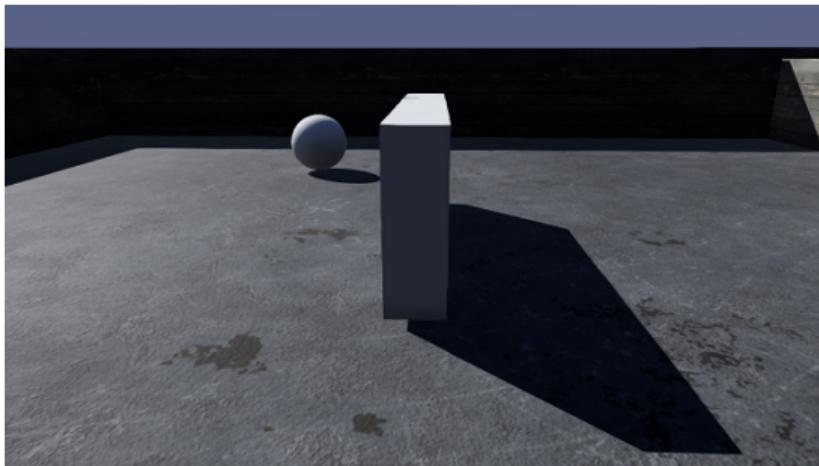
Equivalent to moving the geometry towards the light source.



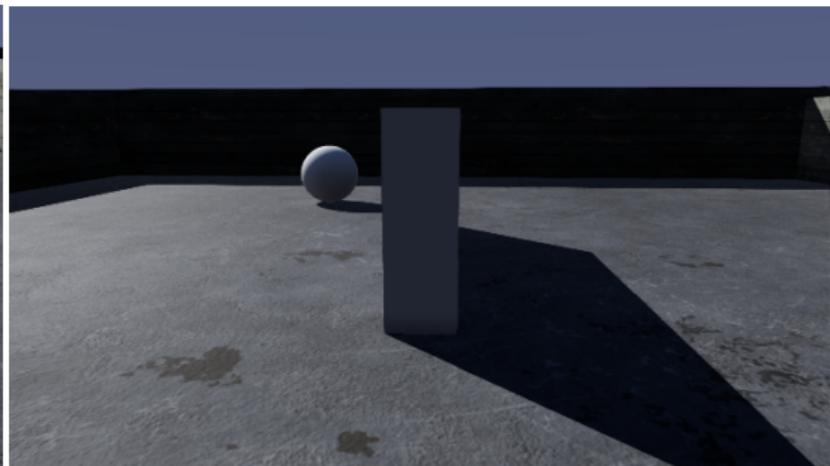
Depth bias (Incorrect shadow detachment shown in red)

Shadow Detachment (Peter Panning)

A large bias make the shadow detaches from the object.



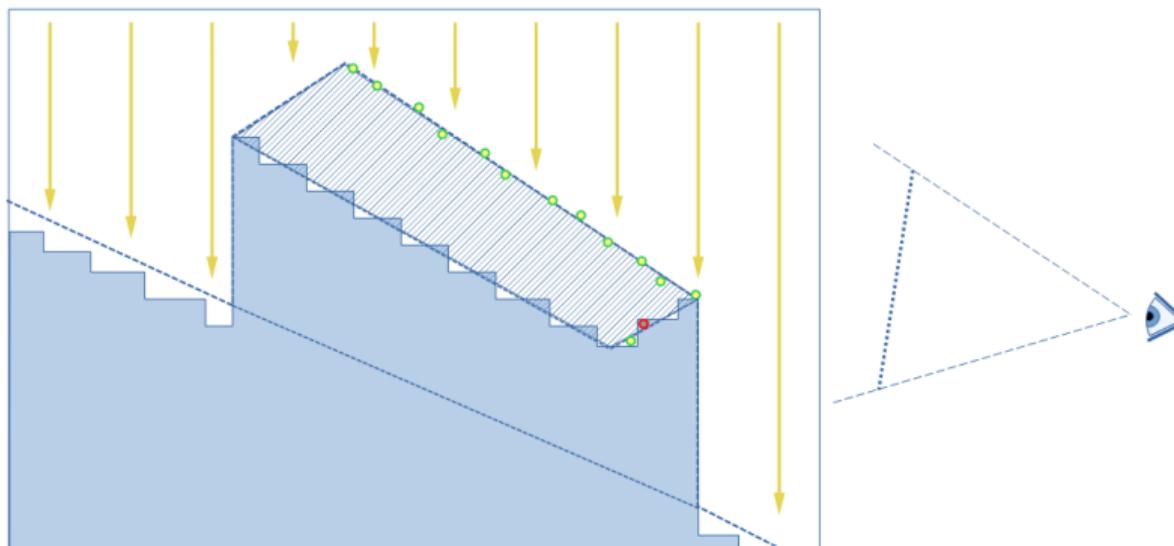
Peter Panning



Without Peter Panning

Using Front Face Culling

- ▶ Shadow acne happens on the visible/front surface of a solid object.
- ▶ Use the back faces (front face culling) of a solid object when rendering the depth map
- ▶ Can avoid using a bias.

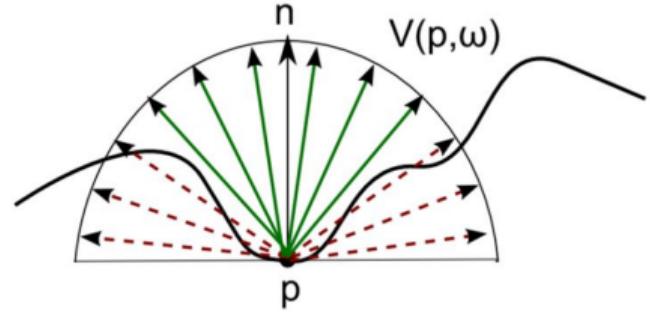


Using the depth of back faces in a shadow map

Ambient Occlusion

Ambient Occlusion

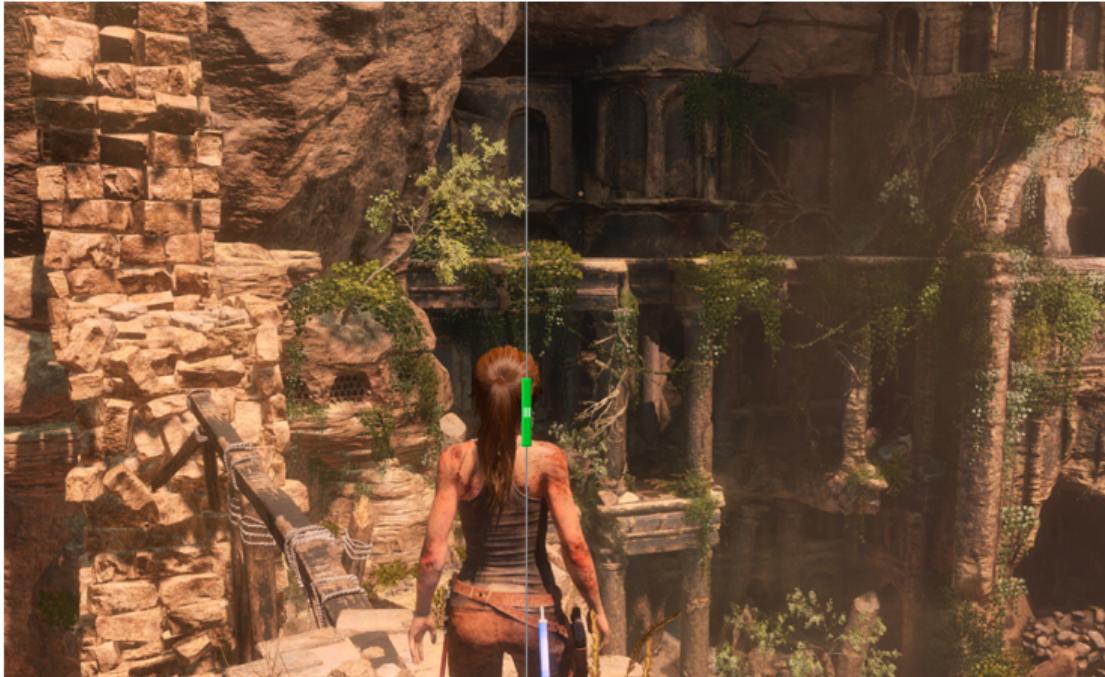
- ▶ Average amount of self-occlusion
- ▶ Darkening areas light hardly reaches
- ▶ Techniques
 - ▶ Ray tracing
 - ▶ Screen space ambient occlusion (SSAO)
 - ▶ Ambient occlusion map (static)



Ambient Occlusion Computing

Ambient Occlusion

Tomb Raider Example (Click to have interactive comparisons)



Ambient Occlusion in Tomb Raider

Summary

- ▶ Shadow Maps
- ▶ Shadow Map Artifacts
- ▶ Ambient Occlusion

Questions?



Chinese shadows