Shadow mapping

UCSD CSE 168
Rendering
Tzu-Mao Li
Today: shadows

https://en.wikipedia.org/wiki/Shadow
Rasterization-based shadows vs Raytraced shadows

foreach object
  foreach pixel sample
    ...

foreach pixel sample
  foreach object
    ...

?
Why do we want rasterization shadow while we have raytracing shadow?

- raytracing shadow is considered the “gold standard” esp. for hard shadow

- modern AAA games still use shadow mapping

- for deformable objects, ray tracing is much slower than rasterization (at least 3x-5x)

- for soft shadow, there are tricks in shadow mapping to soften the shadow ray tracing usually has to rely on denoising

- querying shadow map is just a few texture fetches: amortize cost over time
Shadow mapping: rasterization-based shadow

idea: record the closest surfaces for the light sources
Shadow mapping: rasterization-based shadow

1. render a depth map from the light source
Shadow mapping: rasterization-based shadow

1. render a depth map from the light source
2. project the shading point to the depth map
   if distance_to_light > depth:
     // in shadow
   else:
     // not in shadow
Shadow mapping: rasterization-based shadow

1. render a depth map from the light source
2. project the shading point to the depth map
   if distance_to_light > depth:
     // in shadow
   else:
     // not in shadow

quiz: what can go wrong?
Shadow mapping: rasterization-based shadow

1. render a depth map from the light source
2. project the shading point to the depth map
   \[
   \text{if distance\_to\_light} + \text{epsilon} > \text{depth}:
   \]
   // in shadow
   else:
   // not in shadow

“shadow map bias”

quiz: what can go wrong?
Slope-scaled shadow map bias

Shadow map for different lights

point light
use a cube map
directional/spot light
store a shadow map along the light direction
Shadow map artifacts

- aliasing
- shadow acne

Peter Panning

Shadow map artifacts

resolve by tuning shadow map bias (still painful)

aliasing

shadow acne

Peter Panning

Shadow map artifacts

a lot trickier

PETER PANNING

Why do we see shadow map aliasing? due to limited depth map resolution.
Why do we see shadow map aliasing?

due to limited depth map resolution

solution:
- increase shadow map resolution
- filter the shadow map
Why do we see shadow map aliasing?

due to limited depth map resolution

solution:
- increase shadow map resolution
- filter the shadow map
Cascaded Shadow Map

idea: render different shadow maps for different distance w.r.t. camera

quiz 1: which shadow map should have higher resolution?

https://ogldev.org/www/tutorial49/tutorial49.html
Cascaded Shadow Map

idea: render different shadow maps for different distance w.r.t. camera

quiz 1: which shadow map should have higher resolution?
quiz 2: why is this better than just using one very high-res shadow map?

https://ogldev.org/www/tutorial49/tutorial49.html
Cascaded Shadow Map

https://forum.babylonjs.com/t/cascaded-shadow-maps-csm-are-now-in/7970
Clipmaps in Unreal Engine 5

similar to Cascaded Shadow Map: store different shadow maps for different regions in the scene

Modern game engines use crazy resolutions for shadow mapping

Unreal Engine 5 uses 16k x 16k shadow maps (one for each cubemap/clipmap!!)

quiz: how do we store these crazy shadow map in memory?
Why do we see shadow map aliasing?

due to limited depth map resolution

solution:
- increase shadow map resolution
- filter the shadow map
Recall: sampling and reconstruction

\[ f(u, v) = \sum w_i F(u_i, v_i) \]

interpolating from nearby discrete values
Does standard texture filtering work for shadow mapping?

Percentage closer filtering

idea: filter the visibility function, not the depth

a) Ordinary texture map filtering. Does not work for depth maps.

b) Percentage closer filtering.
Without percentage closer filtering

from Christoph Peters’ fantastic slides: http://momentsingraphics.de/I3D2015.html
Without percentage closer filtering

from Christoph Peters' fantastic slides: http://momentsingraphics.de/I3D2015.html
Percentage closer filter
filters visibility instead of depth

from Christoph Peters' fantastic slides: http://momentsingraphics.de/I3D2015.html
With percentage closer filter

from Christoph Peters’ fantastic slides: http://momentsingraphics.de/I3D2015.html
Percentage closer filter can be used for simulating soft shadows

adjust the filter size based on the distance to the blocker

\[ w_{\text{Penumbra}} = \frac{(d_{\text{Receiver}} - d_{\text{Blocker}})}{d_{\text{Blocker}}} \cdot w_{\text{Light}} \]

- Assumes that blocker, receiver, and light are parallel
Issues with percentage closer filtering

1. needs lots of texture queries
2. still limited by the visibility resolution

from Christoph Peters’ fantastic slides: http://momentsingraphics.de/I3D2015.html
A probabilistic view of percentage closer filtering

A single sample in a normal shadow map represents a step function in shadow(depth)

from Christoph Peters’ fantastic slides: http://momentsingraphics.de/I3D2015.html
A probabilistic view of percentage closer filtering

A single sample in a normal shadow map represents a step function in shadow(depth)

from Christoph Peters’ fantastic slides: http://momentsingraphics.de/I3D2015.html
A probabilistic view of percentage closer filtering

A single sample in a normal shadow map represents a step function in shadow(depth)

from Christoph Peters’ fantastic slides: http://momentsingraphics.de/I3D2015.html
A probabilistic view of percentage closer filtering

A single sample in a normal shadow map represents a step function in shadow(depth)

from Christoph Peters’ fantastic slides: http://momentsingraphics.de/I3D2015.html
A probabilistic view of percentage closer filtering

weighted average over samples gives us a cumulative density function over depth

goal: precompute and compress the depth curve per shadow map pixel

from Christoph Peters’ fantastic slides: http://momentsingraphics.de/I3D2015.html
Variance shadow map

idea: use mean & variance of depth to approximate the large filter

2006

Variance Shadow Maps

William Donnelly* Andrew Lauritzen†

from Christoph Peters’ fantastic slides: http://momentsingraphics.de/I3D2015.html
Variance shadow map

idea: use mean & variance of depth to approximate the large filter

step one: render an anti-aliased shadow map and store mean & second moment

\[ R = z, \quad G = z^2 \]
Variance shadow map

idea: use mean & variance of depth to approximate the large filter

step one: render an anti-aliased shadow map and store mean & second moment
step two: blur the shadow map to get the
average mean & average second moment
(done as a postprocessing of the shadow map rendering)

\[ R = z, \quad G = z^2 \]

from Christoph Peters’ fantastic slides: http://momentsingraphics.de/I3D2015.html
Variance shadow map

idea: use mean & variance of depth to approximate the large filter

step one: render an anti-aliased shadow map and store mean & second moment
step two: blur the shadow map to get the
    average mean & average second moment
    (done as a postprocessing of the shadow map rendering)
step three: during depth testing, use mean/variance to approximate
    the visibility

\[
\mu = E[z] \\
\sigma^2 = E[z^2] - E[z]^2 \\
F_z(z) \geq 1 - \frac{\sigma^2}{\sigma^2 + (z - \mu)^2}
\]

if \( z > \mu \)
    use a lower bound so that we make sure

from Christoph Peters’ fantastic slides: http://momentsingraphics.de/I3D2015.html
Exponential shadow map

instead of $E[z] \& E[z^2]$, store $E[\exp(z)]$ and $E[\exp(2z)]$

(a) The used 16 · 16 kernel.  (b) $F_Z$ with VSM and ESM approximations.

Convolutional shadow map

instead of $E[z]$ & $E[z^2]$, store the expectation of Fourier basis coefficients
Moment shadow map

generalize over all previous methods with moments of functions
Moment shadow map

4 moments per pixel
PCF vs MSM
PCF vs MSM
Shadow mapping is messy

parameters for bias/filtering are usually highly scene dependent
Shadow mapping vs ray tracing

can we move to all ray-traced shadows?

Ray traced

Cascaded shadow maps

Ray tracing is often used for “local shadows”

https://twitter.com/Dachsjaeger/status/1493652010985562113/photo/1
Ray tracing is often used for “local shadows”
Distance field shadow

- use the distance to occluder to blur the shadow
- higher quality shadow and good antialiasing
- doesn’t quite work for deformable objects
Next: transparency

https://developer.nvidia.com/content/transparency-or-translucency-rendering