Lecture 4:
Shadow and Environment Mapping (cont.)
Paper Presentations

• Sign up for paper presentations in class
  - [https://docs.google.com/spreadsheets/d/1iNIhLKzSPwhmohXhCM6kvPgavvWoTnFjh2vE_ZWQKig/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1iNIhLKzSPwhmohXhCM6kvPgavvWoTnFjh2vE_ZWQKig/edit?usp=sharing)
  - I have made some edits to the topics
  - We need one more presenter for next lecture on: Real-Time, All-Frequency Shadows in Dynamic Scenes (Convolutional Shadow Maps, very smart idea)
Recap: Last Lecture

- Shading languages
- The rendering equation
- Basic shadow mapping techniques
- Environment mapping
Today’s Lecture
Outline

• Environment mapping
• Percentage closer soft shadows
• Variance soft shadow mapping
• Project 1
Reflection Maps

Blinn and Newell, 1976
Environment Maps

Miller and Hoffman, 1984
Environment Maps

*Interface, Chou and Williams (ca. 1985)*
Environment Maps

Environment Maps

Cylindrical Panoramas

Cubical Environment Map

180 degree fisheye
Photo by R. Packo
Reflectance Maps

- Reflectance Maps
- Horn, 1977
- Irradiance (N) and Phong (R) Reflection Maps
- Miller and Hoffman, 1984

Mirror Sphere  Chrome Sphere  Matte Sphere
Irradiance Environment Maps

Incident Radiance (Illumination Environment Map)
Analytic Irradiance Formula

Diffuse surface acts like a low-pass filter

\[ E_{lm} = A_l L_{lm} \]

Ramamoorthi and Hanrahan 01
Basri and Jacobs 01

\[ A_l = 2\pi \frac{(-1)^{l-1}}{(l+2)(l-1)} \left[ \frac{l!}{2^l \left( \frac{l}{2} \right)!} \right] \quad l \text{ even} \]
Fourier Transform

Represent a function as a weighted sum of sines and cosines

\[ f(x) = \frac{A}{2} + \frac{2A \cos(t\omega)}{\pi} - \frac{2A \cos(3t\omega)}{3\pi} + \frac{2A \cos(5t\omega)}{5\pi} - \frac{2A \cos(7t\omega)}{7\pi} + \ldots \]
9 Parameter Approximation

Exact image

Order 0
1 term

RMS error = 25 %

\[ Y_{lm}(\theta, \varphi) \]

\[ l \]

\[ m \]

0 1 2

-2 -1 0 1 2

\[ x^2 - y^2 \]
\[ xz \]
\[ 3z^2 - 1 \]
\[ yz \]
\[ xy \]
9 Parameter Approximation

Exact image

Order 1
4 terms

RMS Error = 8%

\[ Y_{lm}(\theta,\varphi) \]
9 Parameter Approximation

Exact image

RMS Error = 1%

For any illumination, average error < 3% [Basri Jacobs 01]
Environment Map Summary

- Very popular for interactive rendering
- Extensions handle complex materials
- But cannot directly combine with shadow maps
  - Env + Shadows? Precomputed transfer!
- Limited to distant lighting assumption
  - Not any more! (since around 2015)
Questions?
Outline

- Environment mapping
- Percentage closer soft shadows
- Variance soft shadow mapping
- Project 1
From Hard Shadows to Soft Shadows

[https://www.timeanddate.com/eclipse/umbra-shadow.html]
Percentage Closer Filtering (PCF)

- Provides **anti-aliasing** at shadows’ edges
  - Not for soft shadows (PCSS is, introducing later)
  - Filtering the results of shadow comparisons

- Why not filtering the shadow map?
  - Texture filtering just averages color components, i.e. you’ll get blurred shadow map first
  - Averaging depth values, then comparing, you still get a **binary** visibility
Percentage Closer Filtering (PCF)

- **Solution [Reeves, SIGGARPH 87]**
  - Perform multiple (e.g. 7x7) depth comparisons for each fragment
  - Then, averages **results of** comparisons
  - e.g. for point P on the floor,
    1. compare its depth with all pixels in the red box, e.g. 3x3
    2. get the compared results, e.g. 1, 0, 1, 1, 0, 1, 1, 1, 0,
    3. take avg. to get visibility, e.g. 0.667
Percentage Closer Filtering

Again, not soft shadows in the umbra/penumbra sense
Percentage Closer Filtering

[https://developer.nvidia.com/gpugems/GPU Gems3/gpugems3_ch08.html]
Percentage Closer Filtering

• Does filtering size matter?
  - Small -> sharper
  - Large -> softer

• Can we use PCF to achieve soft shadow effects?

• Key thoughts
  - From hard shadows to soft shadows
  - What’s the *correct size* to filter?
  - Is it uniform?
Percentage Closer Soft Shadows

• Key observation [Fernando et al.]
  - Where is sharper? Where is softer?

Percentage Closer Soft Shadows

- Key conclusion
  - Filter size <-> blocker distance
  - More accurately, relative average projected blocker depth!

- A mathematical “translation”

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

[Fernando et al.]
Percentage Closer Soft Shadows

• Now the only question:
  - What’s the blocker depth $d_{\text{Blocker}}$

• The complete algorithm of PCSS
  - Step 1: Blocker search
    (getting the average blocker depth in a certain region)
  - Step 2: Penumbra estimation
    (use the average blocker depth to determine filter size)
  - Step 3: Percentage Closer Filtering

• Which region to perform blocker search?
  - Can be set constant (e.g. 5x5), but can be better with heuristics
### Percentage Closer Soft Shadows

- Which region (on the shadow map) to perform blocker search?
  - depends on the light size
  - and receiver’s distance from the light

[Fernando et al.]
Percentage Closer Soft Shadows

Video game: Dying Light
Questions?
Outline

• Environment mapping
• Percentage closer soft shadows
• Variance soft shadow mapping
• Project 1
Percentage Closer Soft Shadows

- The complete algorithm of PCSS
  - Step 1: Blocker search
    (getting the average blocker depth in a certain region)
  - Step 2: Penumbra estimation
    (use the average blocker depth to determine filter size)
  - Step 3: Percentage Closer Filtering

- Which step(s) can be slow?
  - Looking at every texel inside a region (steps 1 and 3)
  - Softer -> larger filtering region -> slower
Variance Soft Shadow Mapping

- Fast blocker search (step 1) and filtering (step 3) [Yang et al.]

- Let’s think from “percentage closer” filtering
  - The percentage of texels that are in front of the shading point, i.e.,
  - how many texels are closer than $t$ in the search area, i.e.,
  - how many students did better than you in an exam
Variance Soft Shadow Mapping

• How many students did better than you in an exam?
  - Using a histogram -> accurate answer!
  - Using a **Normal distribution** -> approximate answer!
  - What do you need to define a normal distribution?
Variance Soft Shadow Mapping

- **Key idea**
  - Quickly compute the mean and variance of depths in an area

- **Mean (average)**
  - Hardware MIPMAPing
  - Summed Area Tables (SAT)

- **Variance**
  - \( \text{Var}(X) = E(X^2) - E(X)^2 \)
  - So you just need the mean of (depth square)
  - Just generate a “square-depth map” along with the shadow map!
Variance Soft Shadow Mapping

• Back to the question
  - Percentage of texels that are closer than the shading point
  - You want to calculate the shade’s area
  - Accurate answer exists (hint: What’s the CDF of a Gaussian PDF?)

[http://work.thaslwanter.at/Stats/html/statsDistributions.html]
Variance Soft Shadow Mapping

- It doesn’t have to be too accurate!
  - Chebychev’s inequality (one-tailed version, for $t > \mu$)

$$P(x \geq t) \leq p_{max}(t) \equiv \frac{\sigma^2}{\sigma^2 + (t - \mu)^2}$$

$\mu$ : mean
$\sigma^2$ : variance

Doesn’t even assume Gaussian distribution!
(Learn more: moments)

[http://work.thaslwanter.at/Stats/html/statsDistributions.html]
Variance Soft Shadow Mapping

- **Performance**
  - Shadow map generation:
    - “square depth map”: parallel, along with shadow map, #pixels
    - MIPMAP building: negligible
    - Summed area table building: #pixels, still parallel!
  - Run time
    - Mean of depth in a range: $O(1)$
    - Mean of depth square in a range: $O(1)$
    - Chebychev: $O(1)$
    - No loops needed!
- Step 3 (filtering) solved perfectly (?)
Variance Soft Shadow Mapping

• Back to Step 1: blocker search (within an area)
  - Also require sampling (loop) earlier, also inefficient
  - The average depth of blockers
  - Not the average depth $z_{avg}$
  - The average depth of those texels whose depth $z < t$

• Key idea
  - Blocker ($z < t$), avg. $z_{occ}$
  - Non-blocker ($z > t$), avg. $z_{unocc}$
Variance Soft Shadow Mapping

- **Key idea**
  - Blocker ($z < t$), avg. $z_{occ}$ (we want to compute)
  - Non-blocker ($z > t$), avg. $z_{unocc}$
  - \[
  \frac{N_1}{N} z_{unocc} + \frac{N_2}{N} z_{occ} = z_{Avg} \]
  - Approximation: $N_1 / N = P(x > t)$, Chebychev!
  - Approximation: $N_2 / N = 1 - P(x > t)$
  - $z_{unocc}$, we really don’t know
  - Approximation: $z_{unocc} = t$ (i.e. shadow receiver is a plane)

- Step 1 solved with negligible additional cost
Variance Soft Shadow Mapping

[https://developer.nvidia.com/gpugems/GPU Gems3/gpugems3_ch08.html]
Variance Soft Shadow Mapping

• Limitations?
  - Light bleeding
  - non-planarity artifact

• Chebychev is to blame?
  - Only valid when \( t > z_{\text{avg}} \)

[https://developer.nvidia.com/gpugems/GPUGems3/gpugems3_ch08.html]
Questions?
Outline

• Environment mapping
• Percentage closer soft shadows
• Variance soft shadow mapping
• Project 1