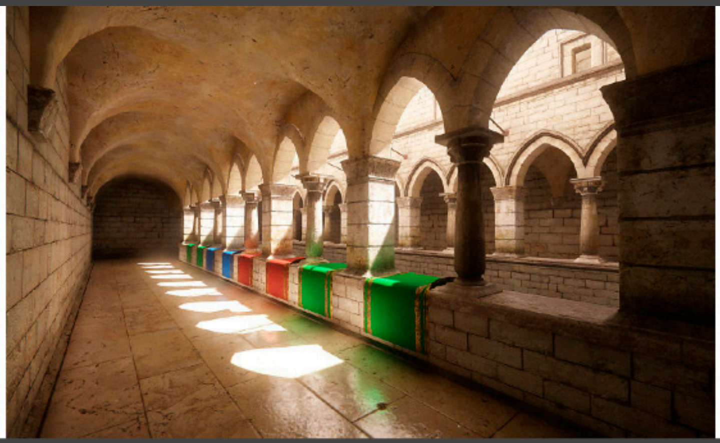
Real-Time High Quality Rendering

GAMES202, Lingqi Yan, UC Santa Barbara

Lecture 4:
Real-Time Shadows 2

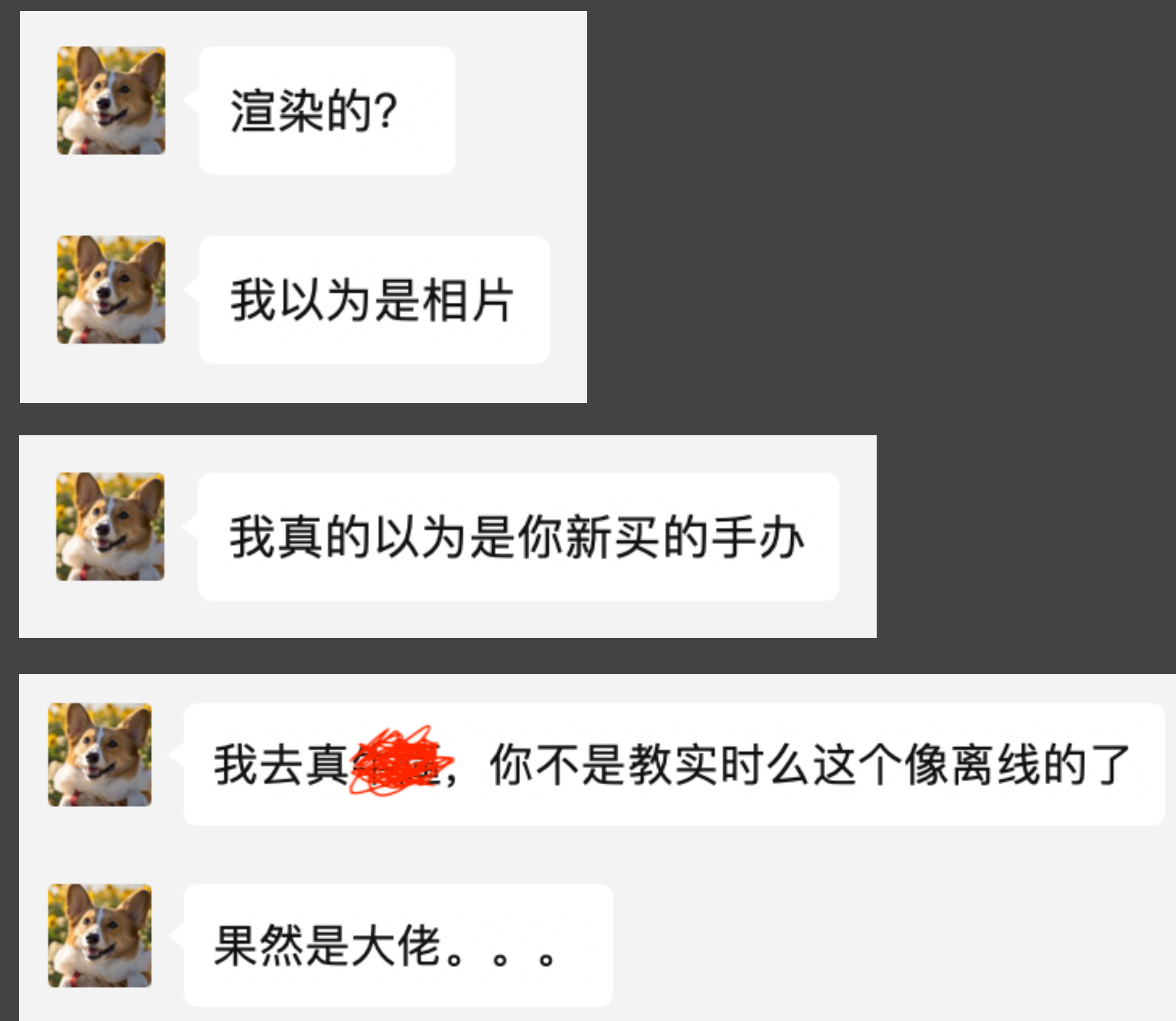



Announcement

- Assignment 1 will be released today or tomorrow
  - Due in 1.5 weeks

Before Assignment 1 is Released

- @Maxwell has already implemented a stunning version



— Edward (Shiqiu) Liu, NVIDIA

Announcement

- Assignment 1 will be released today or tomorrow
  - Due in 1.5 weeks
- Quite a lot to discuss today
  - WHY?

★★ Mysterious Power ★★

[Maru vs. Solar, IEM 2021]

★★ Great News ★★

- **4 papers accepted to SIGGRAPH 2021!**
  - Thanks to the all-star students and collaborators
  - Might be the winner of most #papers in this SIGGRAPH in the rendering domain
  - Career #SIGGRAPH (including SIGGRAPH Asia and ToG) papers has reached 20
- Some deeper thoughts

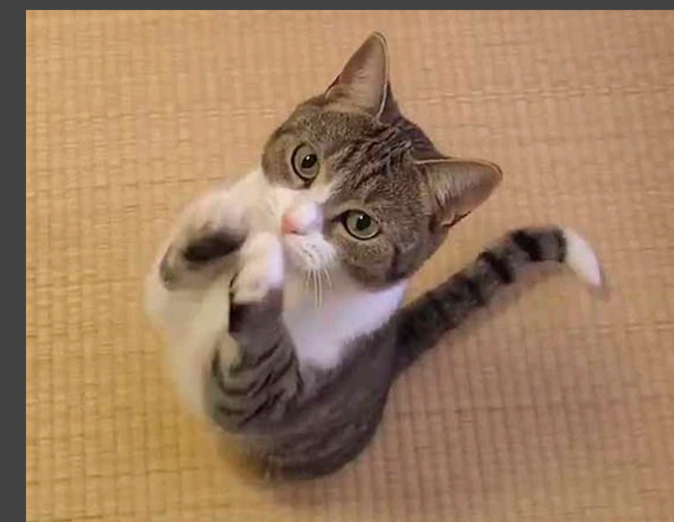
★★ Greater News ★★

- **I got my PlayStation 4 BACK!!!!!!**
  - My story
- Next Goals
  - 30 to unlock PlayStation 5!
  - La Campanella to unlock Steinway & Sons!

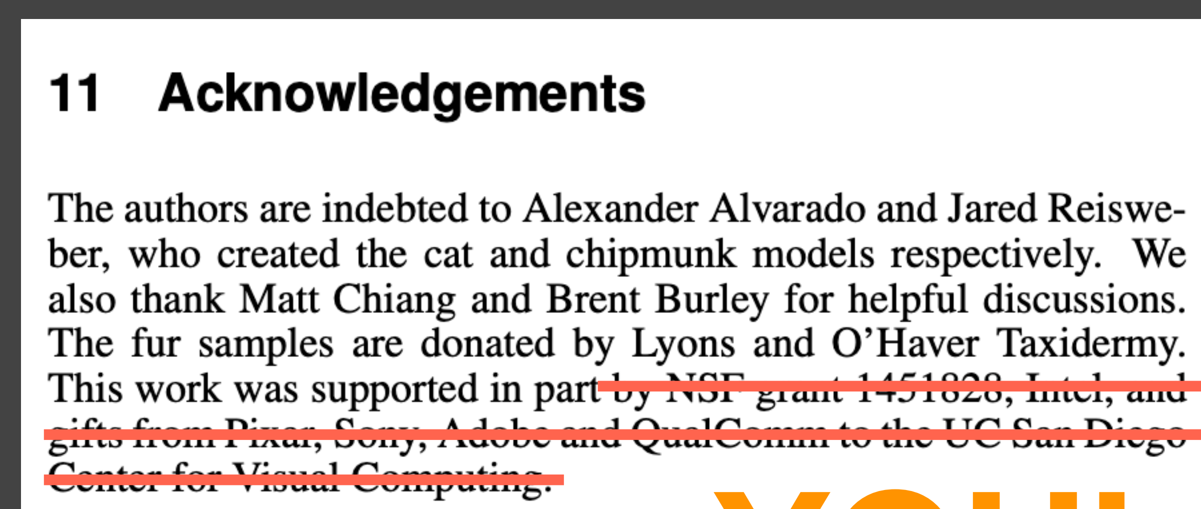
[Sony PlayStation 4]

★★ Food for Thought ★★

- My industrial friends
  - I need your help **immediately**
  - $50,000 can support a Ph.D student for 1 year
- Various ways
  - **Fellowships** and **gift funding** are certainly welcomed
  - I am also happy to be your **consultant** except you-know-who
  - …



https://modkat.com/blogs/modkat-purrr/how-to-prevent-or-stop-your-cat-from-begging-for-food



11 Acknowledgements

The authors are indebted to Alexander Alvarado and Jared Reiswe-ber, who created the cat and chipmunk models respectively. We also thank Matt Chiang and Brent Burley for helpful discussions. The fur samples are donated by Lyons and O'Haver Taxidermy. This work was supported in part ~~by NSF grant 1451828, Intel, and gifts from Pixar, Sony, Adobe and QualComm to the UC San Diego Center for Visual Computing.~~

Last Lecture

- Shadow mapping
  - Issues from shadow mapping and solutions
- The math behind shadow mapping
- Percentage closer soft shadows

Today

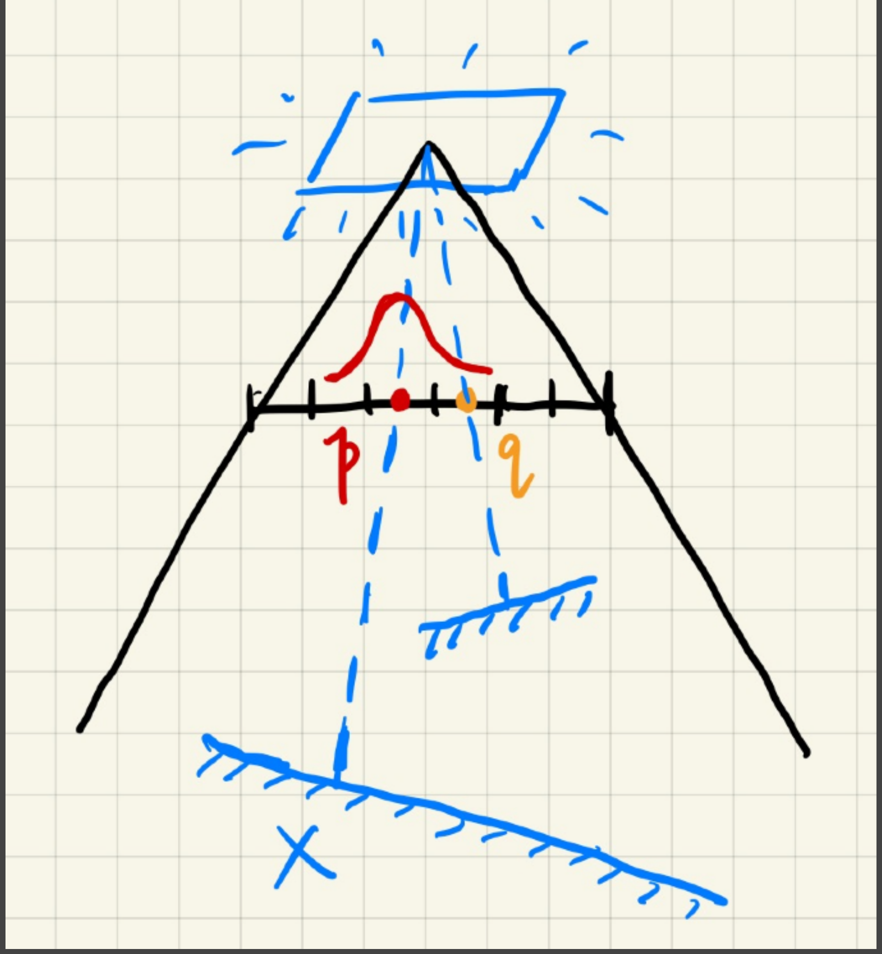
- More on PCF and PCSS
- Variance soft shadow mapping
- MIPMAP and Summed-Area Variance Shadow Maps
- Moment shadow mapping

A Deeper Look at PCF

- The math behind PCF

- Filter / convolution:

$$[w * f](p) = \sum_{q \in \mathcal{N}(p)} w(p,q) f(q)$$



- In PCSS

$$V(x) = \sum_{q \in \mathcal{N}(p)} w(p,q) \cdot \chi^{+}[D_{\mathrm{SM}}(q) - D_{\mathrm{scene}}(x)]$$

A Deeper Look at PCF

$$V(x) = \sum_{q \in \mathcal{N}(p)} w(p,q) \cdot \chi^{+}[D_{\mathrm{SM}}(q) - D_{\mathrm{scene}}(x)]$$

- Therefore, PCF is not filtering the shadow map then compare

$$V(x) \neq \chi^{+}\{[w * D_{\mathrm{SM}}](q) - D_{\mathrm{scene}}(x)\}$$

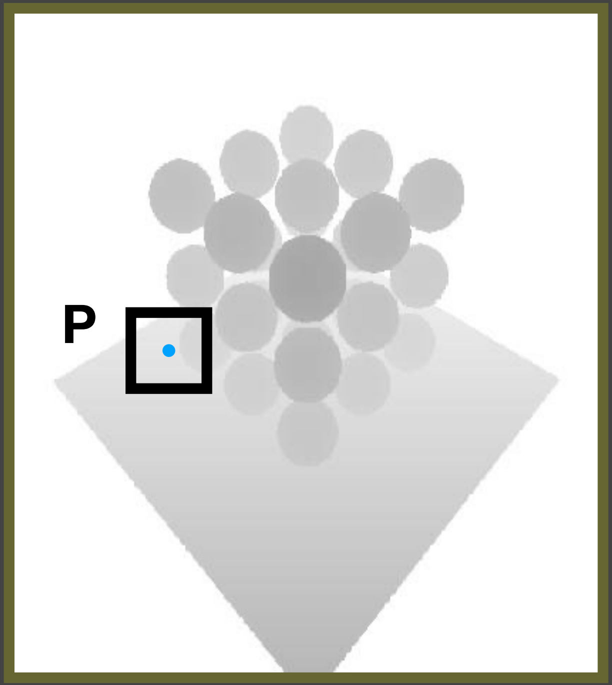
- And PCF is not filtering the resulting image with binary visibilities

$$V(x) \neq \sum_{q \in \mathcal{N}(p)} w(p,q) V(q)$$

Revisiting PCSS

- 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

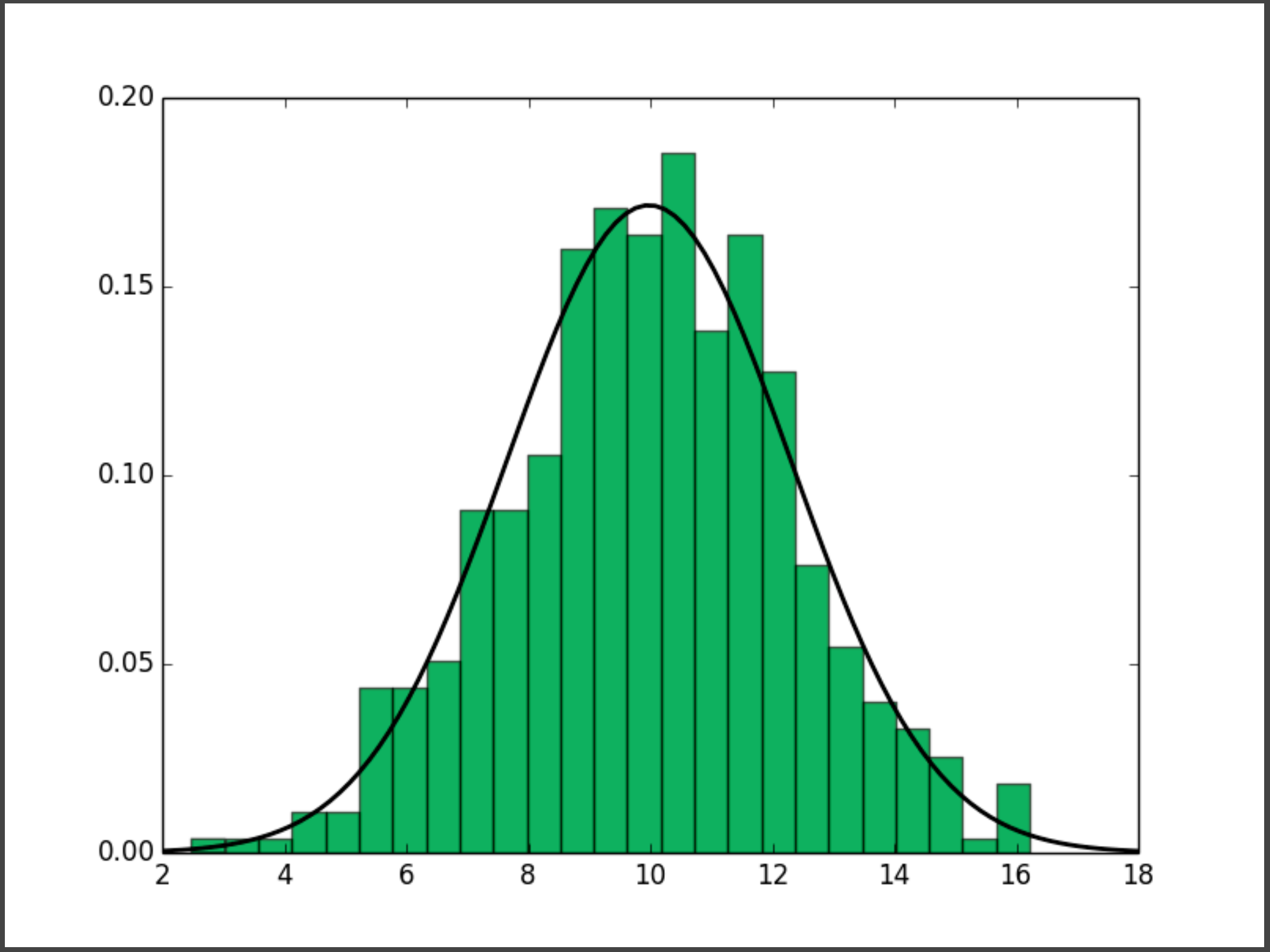


Variance Soft Shadow Mapping

- 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?

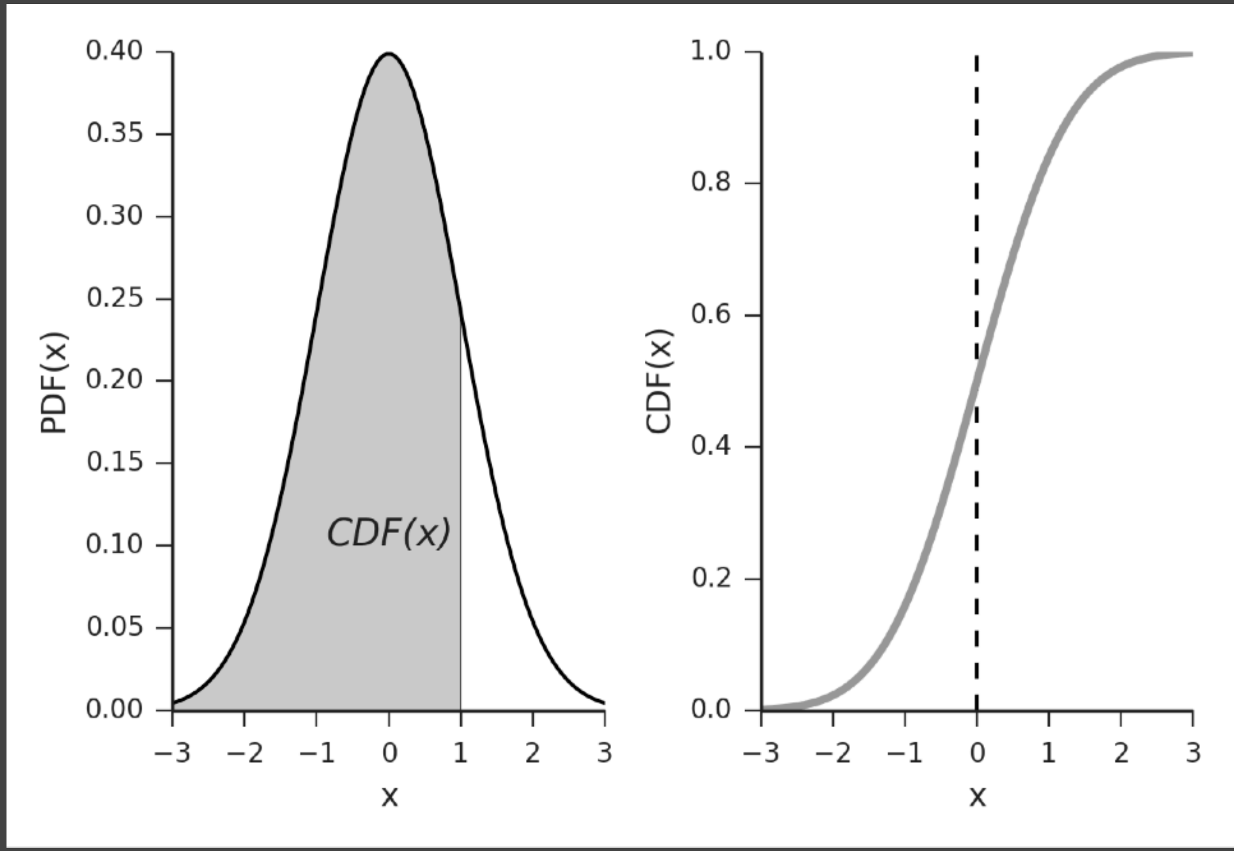


Variance Soft Shadow Mapping

- Key idea
  - Quickly compute the mean and variance of depths in an area
- Mean (average)
  - Hardware MIPMAPing
  - Summed Area Tables (SAT)
- Variance
  - $\mathrm{Var}(X) = \mathrm{E}(X^2) - \mathrm{E}^2(X)$
  - So you just need the mean of (depth^2)
  - 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

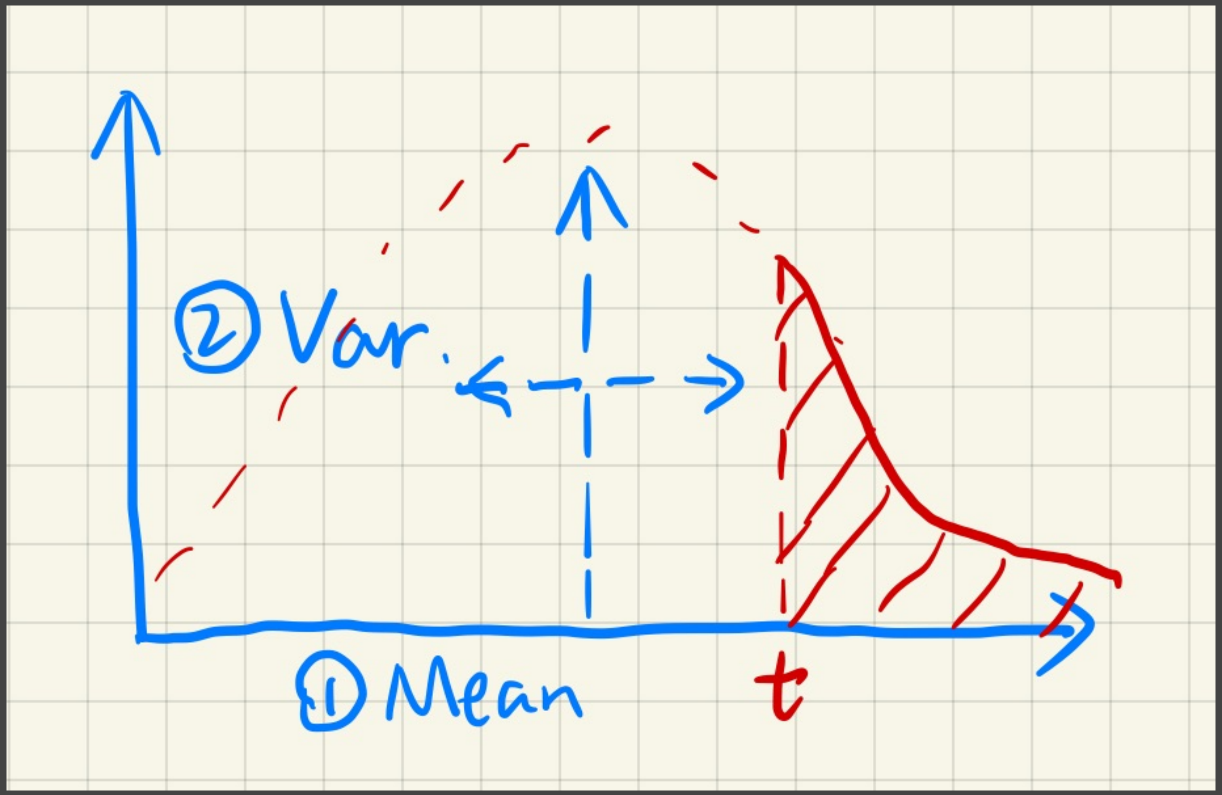
- It doesn't have to be too accurate!
  - Chebychev's inequality (one-tailed version, for t > mu)

$$P(x > t) \leq \frac{\sigma^2}{\sigma^2 + (t - \mu)^2}$$

μ : mean

σ^2 : variance

Doesn't even assume Gaussian distribution!



Variance Soft Shadow Mapping

- Performance
  - Shadow map generation:
    - “square depth map”: parallel, along with shadow map, #pixels
    - Anything else?
  - Run time
    - Mean of depth in a range: O(1)
    - Mean of depth square in a range: O(1)
    - Chebychev: O(1)
    - No samples / 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}

4	4	8	8	8
4	4	8	8	8
4	4	6	8	8
6	6	6	8	9
8	8	8	9	9

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: N1 / N = P(x > t), Chebychev!
  - Approximation: N2 / 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

4	4	8	8	8
4	4	8	8	8
4	4	6	8	8
6	6	6	8	9
8	8	8	9	9

Variance Soft Shadow Mapping

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

Questions?

Today

- More on PCF and PCSS
- Variance soft shadow mapping
- MIPMAP and Summed-Area Variance Shadow Maps
- Moment shadow mapping

Variance Soft Shadow Mapping

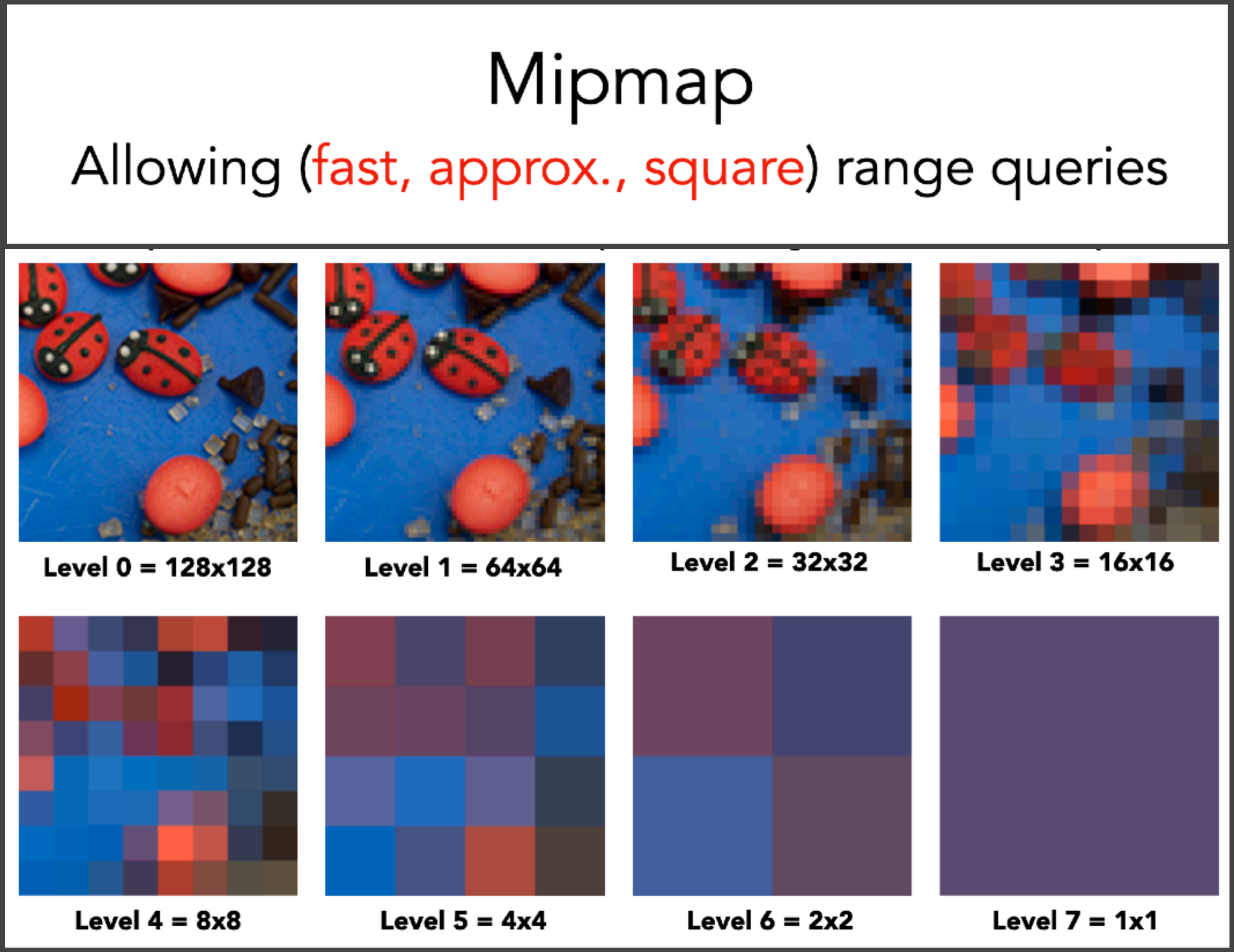
- Key observation: in order to accelerate
  - Need to quickly grab μ and σ from an arbitrary range (rectangular)

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

- For the average μ, this is rectangular range query
  - Can be handled by both MIPMAP and Summed Area Table (SAT)

MIPMAP for Range Query

- Recall: MIPMAP



- Note: still approximate even with trilinear interpolation

SAT for Range Query

- Classic data structure and algorithm (prefix sum)
  - In 1D:

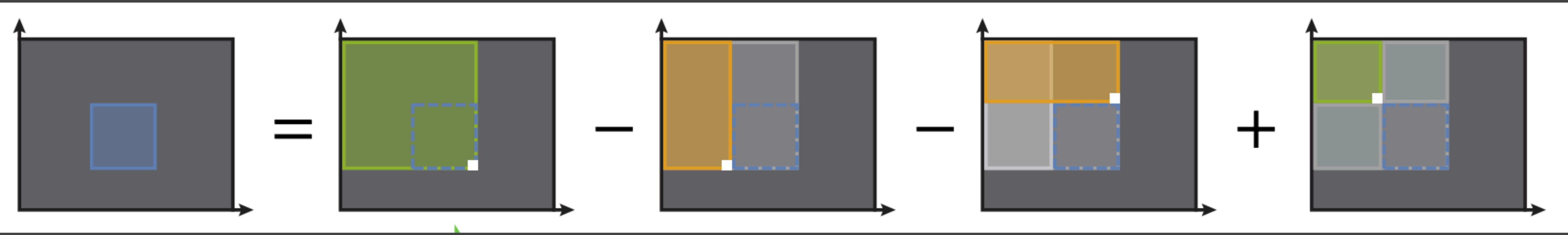
Input:	1	3	5	3	7	1	3	8	6	4

sum = ?

SAT:	1	4	9	12	19	20	23	31	37	41

SAT for Range Query

- Classic data structure and algorithm
  - In 2D:



[Gamboa et al.]

- Note: accurate, but need $O(n)$ time and storage to build
  - Storage might not be an issue
  - Can we speed up building SAT?

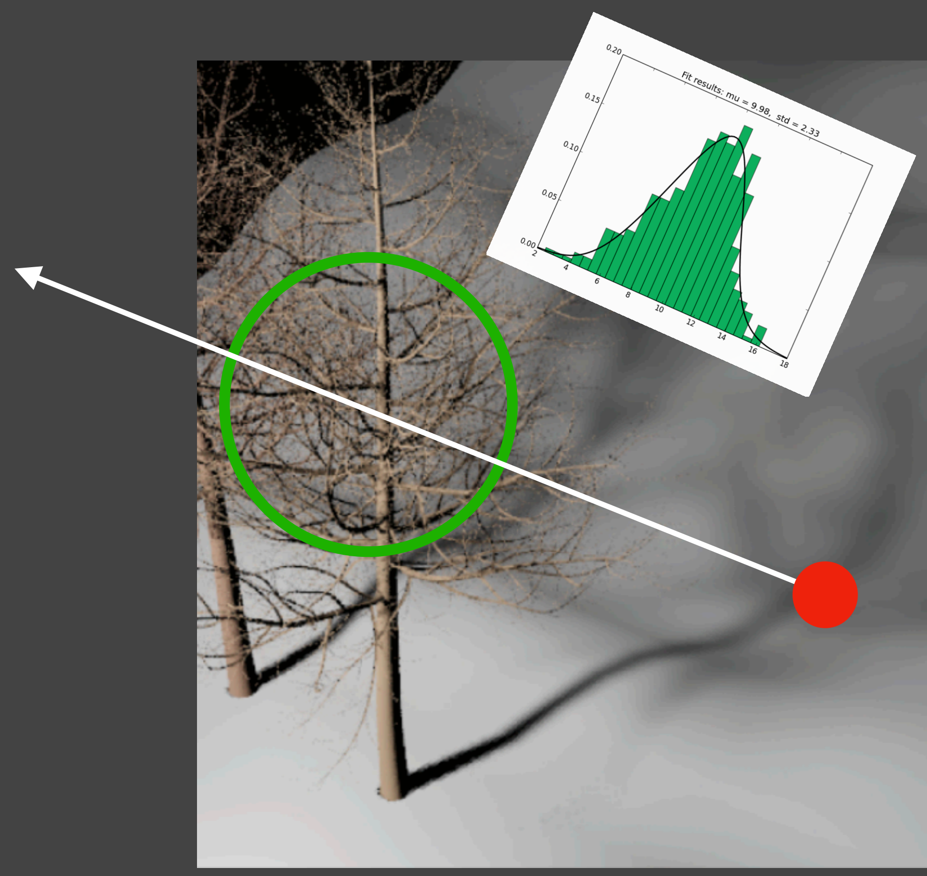
Questions?

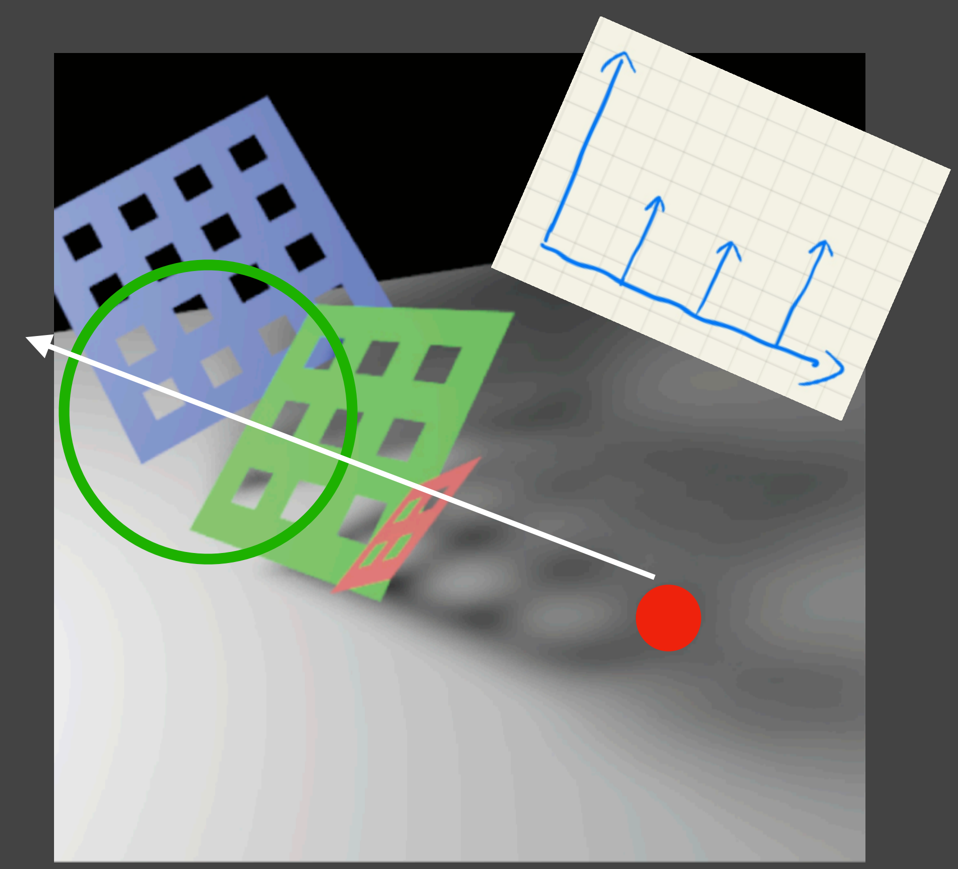
Today

- More on PCF and PCSS
- Variance soft shadow mapping
- MIPMAP and Summed-Area Variance Shadow Maps
- Moment shadow mapping

Revisit: VSSM

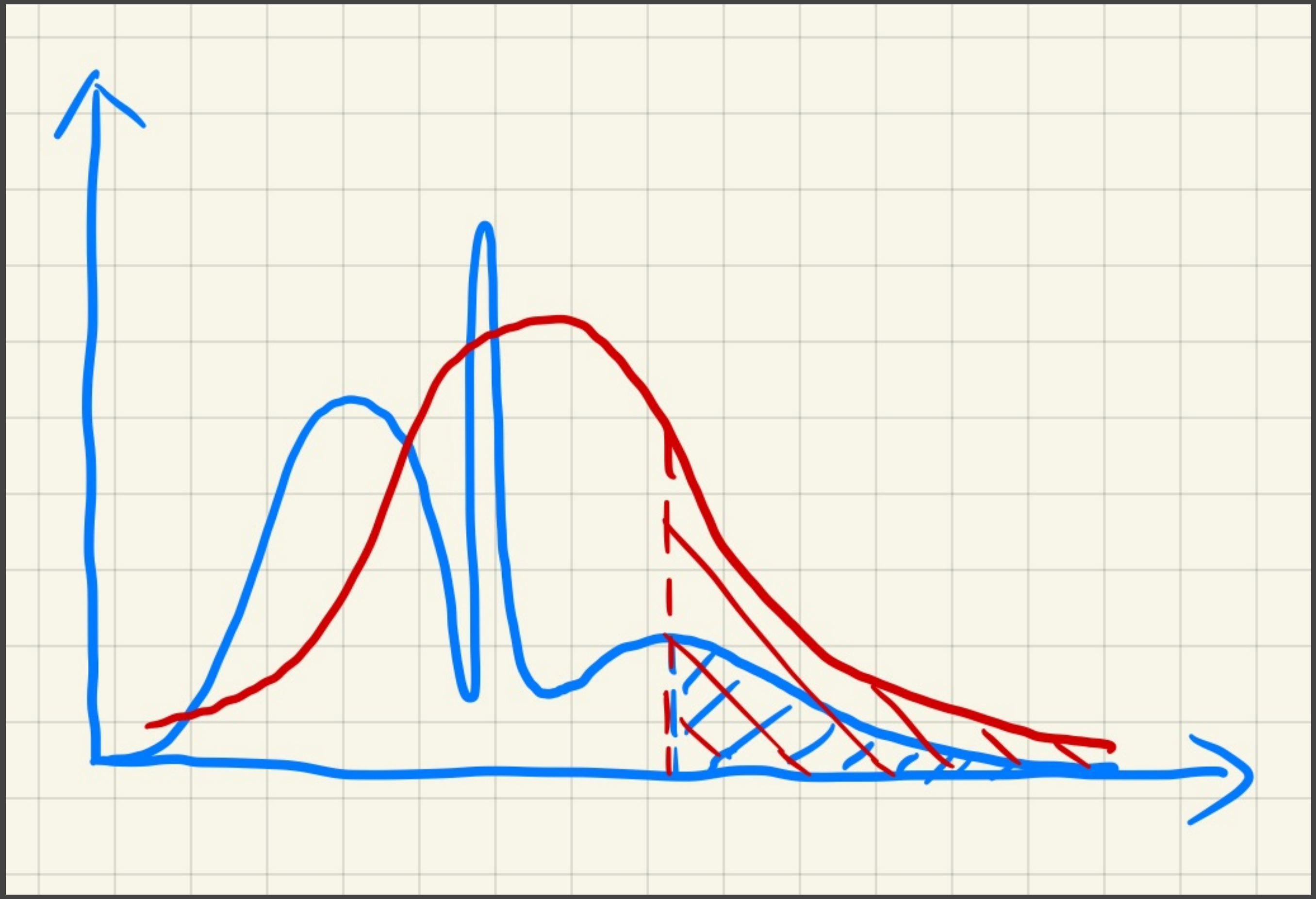
- Is a normal distribution always good enough to approximate the distribution of fragments' distances?





Revisit: VSSM

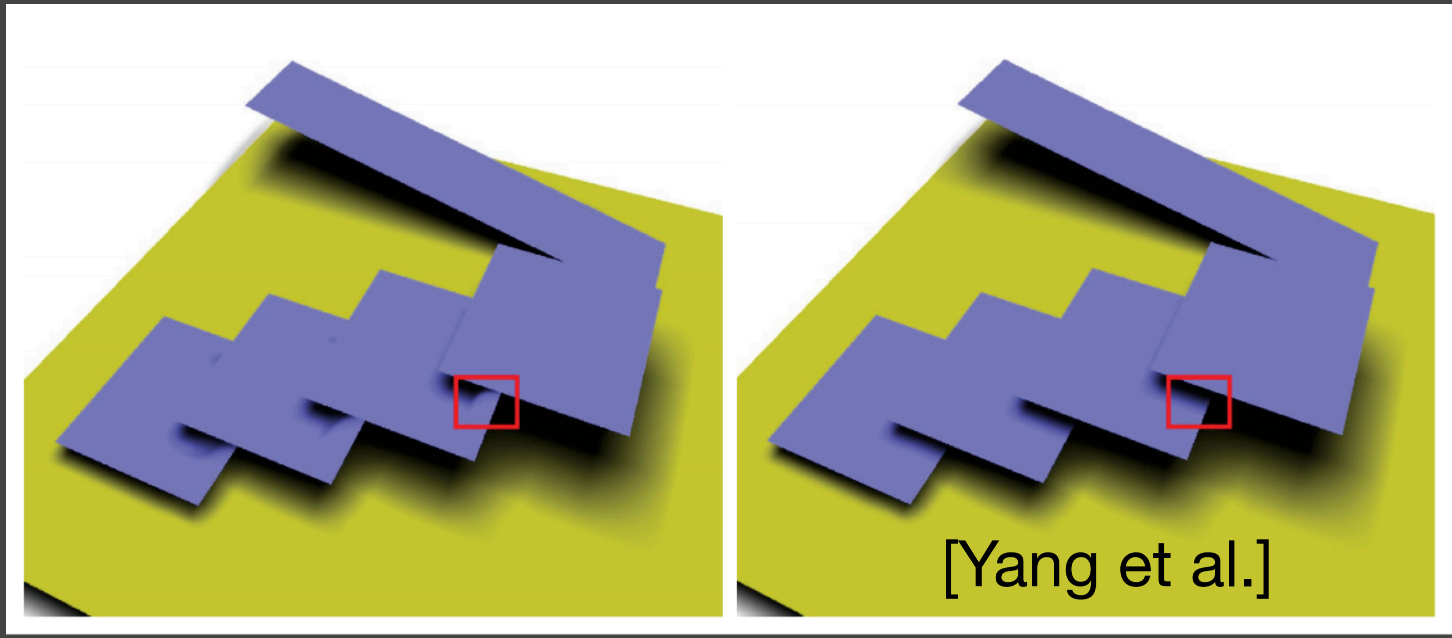
- Issues if the depth distribution is inaccurate
  - Overly dark: may be aceeptable
  - Overly bright: LIGHT LEAKING!



Revisit: VSSM

- Limitations?
  - Light leaking
  - non-planarity artifact
- Chebychev is to blame?
  - Only valid when $t > z_{avg}$
- Can we do better?

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



[Yang et al.]

Moment Shadow Mapping

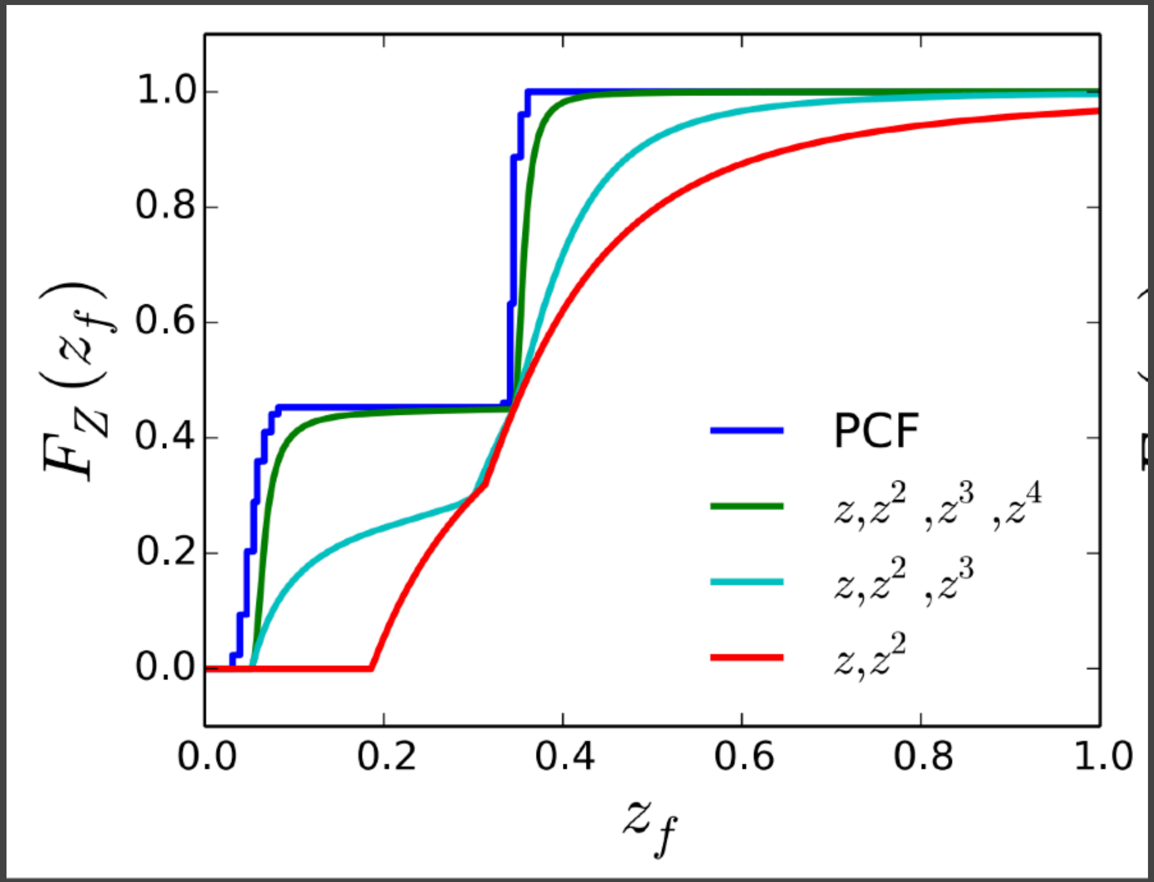
- Goal
  - Represent a distribution more accurately
    (but still not too costly to store)
- Idea
  - Use higher order moments to represent a distribution

Moment Shadow Mapping

- Moments
  - Quite a few variations on the definition
  - We use the simplest:
    $x, x^2, x^3, x^4, \ldots$
  - So, VSSM is essentially using the first two orders of moments

Moment Shadow Mapping

- What can moments do?
  - Conclusion:
    first m orders of moments can represent a function with $m/2$ steps
  - Usually, 4 is good enough to approximate the actual CDF of depth dist.
  - ~~How to restore a CDF whose moments match the given moments~~



[Peters et al., Moment Shadow Mapping]

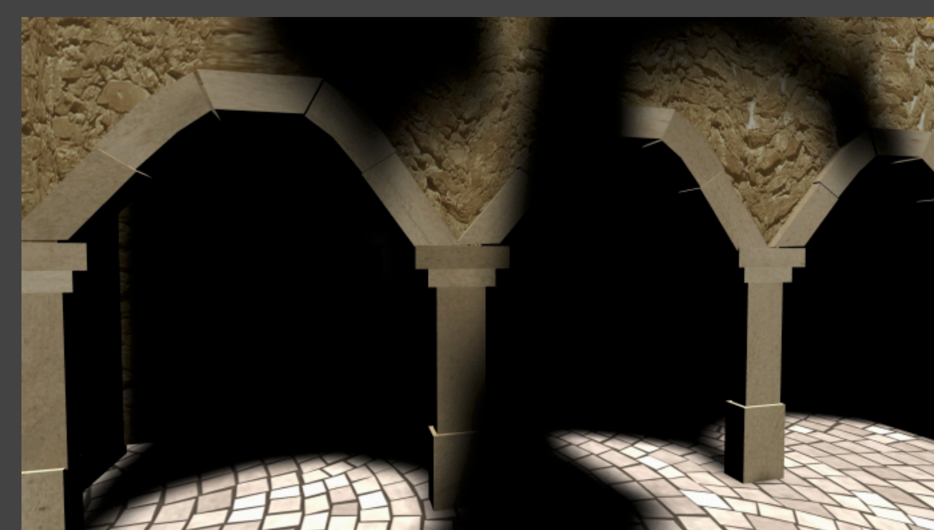
Moment Shadow Mapping

- Moment Shadow Mapping
  - Extremely similar to VSSM
  - When generating the shadow map, record z, z^2, z^3, z^4
  - Restore the CDF during blocker search & PCF

Moment Shadow Mapping

- Pro: very nice results
- Cons
  - Costly storage (might be fine)
  - Costly performance (in the reconstruction)

Variance shadow mapping, 1.15ms



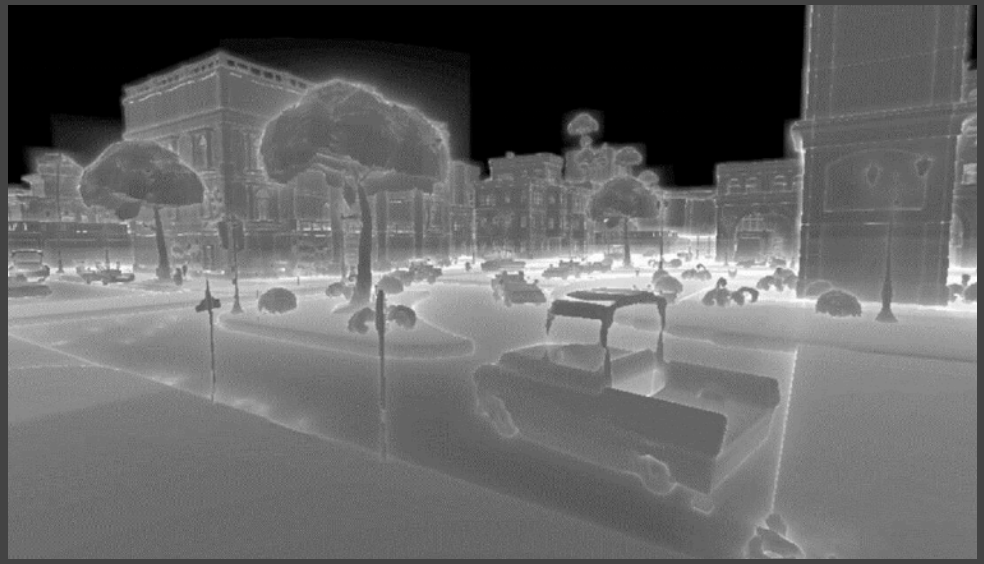
Moment shadow mapping, 1.46ms

[Peters et al., Moment Shadow Mapping]

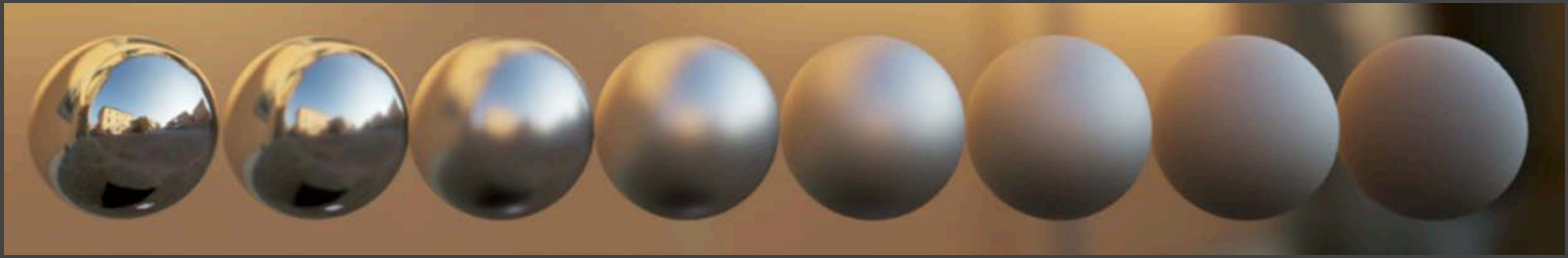
Questions?

Next Lecture

- Distance Field Soft Shadows
- Real-Time Environment Lighting
  - Prefiltering
  - The split sum method



https://docs.unrealengine.com/en-US/BuildingWorlds/LightingAndShadows/RayTracedDistanceFieldShadowing/index.html



https://cdn2.unrealengine.com/Resources/files/2013SiggraphPresentationsNotes-26915738.pdf

Thank you!