NVIDIA

REFRAMING LIGHT TRANSPORT FOR REAL-TIME

Chris Wyman

Principal Research Scientist

High Performance Graphics 2020

START WITH SOME THANKS

Real-time capture, 3.4 million area lights

START WITH SOME THANKS

A lot of the thoughts you'll hear were informed & helped along by numerous others

Benedikt Bitterli,

Wojciech Jarosz, Matt Pharr, Peter Shirley, Aaron Lefohn,

Kate Anderson,

Alexey Panteleev, Tim Cheblokov, Pawel Kozlowski, Jacob Munkberg, Jon Hasselgren, Mike Songy,

Petrik Clarberg, Simon Kallweit, Marco Salvi, William Newhall, Bob Alfieri, Jacopo Pantaleoni,

John Burgess, Apollo Ellis, Kai-Hwa Yao, Lucy Chen

EXCITING TIME IN RENDERING

Real-time capture, 83,000 area lights

EXCITING TIME IN RENDERING

Real-time ray tracing is here!

EXCITING TIME IN RENDERING

Real-time ray tracing is here!

Big question… What's it good for? Besides the obvious:

- Offline: Accelerate computation
- Real-time: Better shadows, AO, reflections

EXCITING TIME IN RENDERING

Real-time ray tracing is here!

Big question… What's it good for? Besides the obvious:

- Offline: Accelerate computation
- Real-time: Better shadows, AO, reflections

Move to path tracing was good for film

- But we can't rely on Moore's Law for real-time performance

EXCITING TIME IN RENDERING

Real-time ray tracing is here!

Big question… What's it good for? Besides the obvious:

- Offline: Accelerate computation
- Real-time: Better shadows, AO, reflections

Move to path tracing was good for film

- But we can't rely on Moore's Law for real-time performance

Lots of space between "faster offline" & "better shadows"

- Few have applied ***constraints of real-time rendering*** to general problems in ray and path tracing

WHAT ARE THESE CONSTRAINTS?

Outlined in SIGGRAPH 2019 “Open Problems”; will discuss some specific thoughts today

Real-time capture, 13,000 area lights + emissive environment map

WHAT ARE THESE CONSTRAINTS?

Outlined in SIGGRAPH 2019 “Open Problems”; will discuss some specific thoughts today

- Compute until time constraint reached (not a quality target)
  - Time-to-image is a key metric; include per-frame data structure build costs
- Desire tunable quality knobs

WHAT ARE THESE CONSTRAINTS?

Outlined in SIGGRAPH 2019 “Open Problems”; will discuss some specific thoughts today

- Compute until time constraint reached (not a quality target)
  - Time-to-image is a key metric; include per-frame data structure build costs
- Desire tunable quality knobs
- Assume scenes are dynamic
  - And we want robustness even under user control

WHAT ARE THESE CONSTRAINTS?

Outlined in SIGGRAPH 2019 “Open Problems”; will discuss some specific thoughts today

Compute until time constraint reached (not a quality target)

- Time-to-image is a key metric; include per-frame data structure build costs

Desire tunable quality knobs

Assume scenes are dynamic

- And we want robustness even under user control

Need temporally stable image quality

Spatial and temporal reuse assumed

- Either to reduce cost or increase quality at a given cost

WHAT ARE THESE CONSTRAINTS?

Outlined in SIGGRAPH 2019 “Open Problems”; will discuss some specific thoughts today

- Compute until time constraint reached (not a quality target)
  - Time-to-image is a key metric; include per-frame data structure build costs
- Desire tunable quality knobs
- Assume scenes are dynamic
  - And we want robustness even under user control
- Need temporally stable image quality
- Spatial and temporal reuse assumed
  - Either to reduce cost or increase quality at a given cost
- Managing data and code divergence and incoherence

THERE'S BEAUTY & ELEGANCE IN BRUTE FORCE

But care is required for real-time

Real-time capture, 7,500 area lights + emissive environment map

THERE'S BEAUTY & ELEGANCE IN BRUTE FORCE

But care is required for real-time

- Simply not possible for many problems
  - E.g., can't expect to touch millions of lights, for all pixels, for all frames
  - E.g., can't trace all subsurface scattering paths

THERE'S BEAUTY & ELEGANCE IN BRUTE FORCE

But care is required for real-time

Simply not possible for many problems

- E.g., can't expect to touch millions of lights, for all pixels, for all frames
- E.g., can't trace all subsurface scattering paths

Suggests we need for stochastic techniques

- But must work in a streaming fashion, to fully leverage GPU resources

WHAT IF I TOLD YOU…

Or “speaking of probability and statistics…”

WHAT IF I TOLD YOU…

Or "speaking of probability and statistics…"

A mathematical method exists that:

- Integrates a non-analytic (and unknown) function

WHAT IF I TOLD YOU…

Or “speaking of probability and statistics…”

A mathematical method exists that:

- Integrates a non-analytic (and unknown) function
- Function changes ***during sampling***, in unknown and unknowable ways

WHAT IF I TOLD YOU…

Or “speaking of probability and statistics…”

A mathematical method exists that:

- Integrates a non-analytic (and unknown) function
- Function changes ***during sampling***, in unknown and unknowable ways
- The sensor properties for ground-truth measurement are unknown

WHAT IF I TOLD YOU…

Or "speaking of probability and statistics…"

A mathematical method exists that:

- Integrates a non-analytic (and unknown) function
- Function changes ***during sampling***, in unknown and unknowable ways
- The sensor properties for ground-truth measurement are unknown
- Can use sampling rates as low as 1 in 10^6 to 10^7
- Often only a poor-quality, uncontrollable PDF for sampling

WHAT IF I TOLD YOU…

Or "speaking of probability and statistics…"

A mathematical method exists that:

- Integrates a non-analytic (and unknown) function
- Function changes ***during sampling***, in unknown and unknowable ways
- The sensor properties for ground-truth measurement are unknown
- Can use sampling rates as low as 1 in 10^6 to 10^7
- Often only a poor-quality, uncontrollable PDF for sampling
- Gives answers with a 3-5% margin of error

WHAT IF I TOLD YOU…

Or “speaking of probability and statistics…”

A mathematical method exists that:

- Integrates a non-analytic (and unknown) function
- Function changes ***during sampling***, in unknown and unknowable ways
- The sensor properties for ground-truth measurement are unknown
- Can use sampling rates as low as 1 in 10^6 to 10^7
- Often only a poor-quality, uncontrollable PDF for sampling
- Gives answers with a 3-5% margin of error

Would that interest you?

WHAT IF I TOLD YOU…

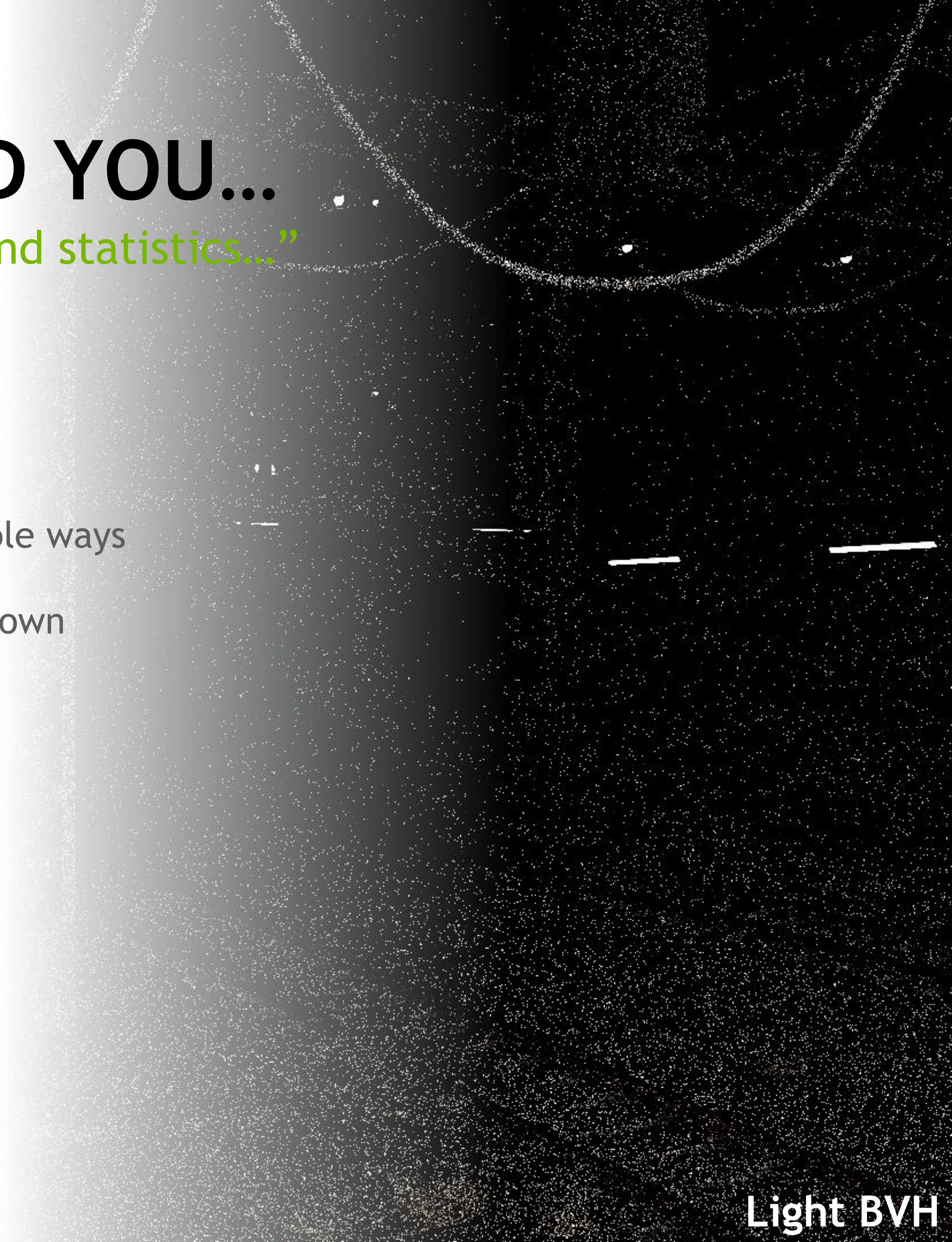
Or “speaking of probability and statistics…”

A mathematical method exists that:

- Integrates a non-analytic (and unknown) function
- Function changes ***during sampling***, in unknown and unknowable ways
- The sensor properties for ground-truth measurement are unknown
- Can use sampling rates as low as 1 in 10^6 to 10^7
- Often only a poor-quality, uncontrollable PDF for sampling
- Gives answers with a 3-5% margin of error

Would that interest you?

- 3-5% error seems ***amazing*** after these results



WHAT IF I TOLD YOU…

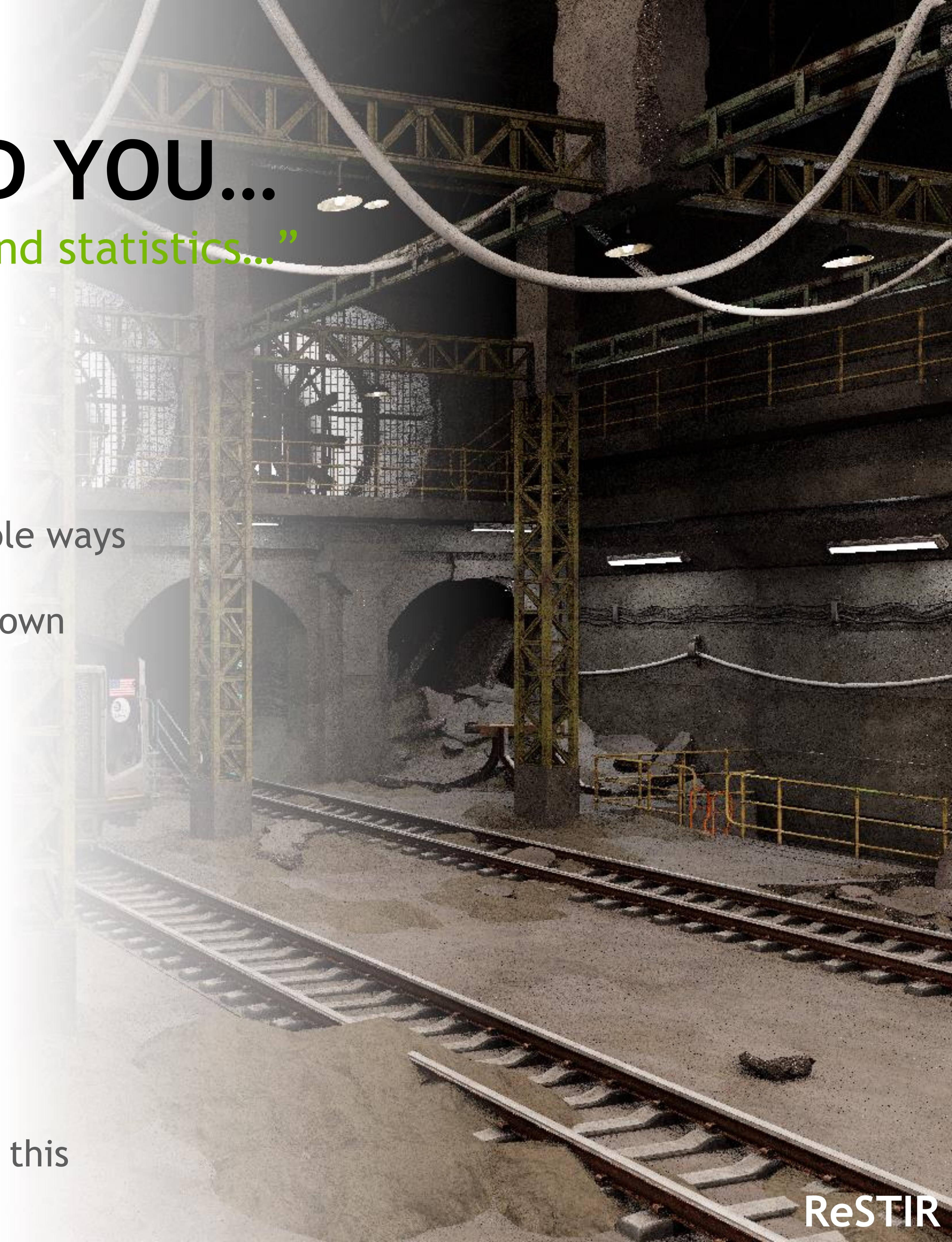
Or "speaking of probability and statistics…"

A mathematical method exists that:

- Integrates a non-analytic (and unknown) function
- Function changes ***during sampling***, in unknown and unknowable ways
- The sensor properties for ground-truth measurement are unknown
- Can use sampling rates as low as 1 in 10^6 to 10^7
- Often only a poor-quality, uncontrollable PDF for sampling
- Gives answers with a 3-5% margin of error

Would that interest you?

- 3-5% error seems ***amazing*** after these results … and more like this



WHERE CAN YOU FIND OUT MORE?

WHERE CAN YOU FIND OUT MORE?

Look for help from… political polling

WHERE CAN YOU FIND OUT MORE?

Look for help from… political polling

Never know the collective ground truth, and it can change day-to-day

Actual voting (i.e., the ground-truth sensor) depends on who shows up

Uses only a few hundreds to thousands of samples; these take days to weeks to collect

Frequently cannot choose samples; must correct for the distribution you get

Quite accurate, considering they prefilter ***prior*** to a binary decision

WHERE CAN YOU FIND OUT MORE?

Look for help from… political polling

- Never know the collective ground truth, and it can change day-to-day
- Actual voting (i.e., the ground-truth sensor) depends on who shows up
- Uses only a few hundreds to thousands of samples; these take days to weeks to collect
- Frequently cannot choose samples; must correct for the distribution you get
- Quite accurate, considering they prefilter ***prior*** to a binary decision

Particularly interesting aspects include:

- Spatial and temporal aggregation often improves predictive quality

DON'T TRUST POLITICAL POLLING?

Numerous other applications:

- Over 100 US agencies collate statistics
- Census, employment, economic indicators, resources, infrastructure, etc., etc.
- Most with error **much** lower than 3-5%
- With more expensive and principled sampling, regularly validated with measurements

DON’T TRUST POLITICAL POLLING?

Numerous other applications:

- Over 100 US agencies collate statistics
- Census, employment, economic indicators, resources, infrastructure, etc., etc.
- Most with error **much** lower than 3-5%
- With more expensive and principled sampling, regularly validated with measurements

Particular call out to Statistics Sweden:

- Open-access, peer-reviewed “Journal of Official Statistics”
- Many of their statisticians write supremely clear articles that I found invaluable

PROMOTES PROMISE OF PATH TRACING IN REAL-TIME

Statistical techniques remove many assumptions; assets become easier to author

PROMOTES PROMISE OF PATH TRACING IN REAL-TIME

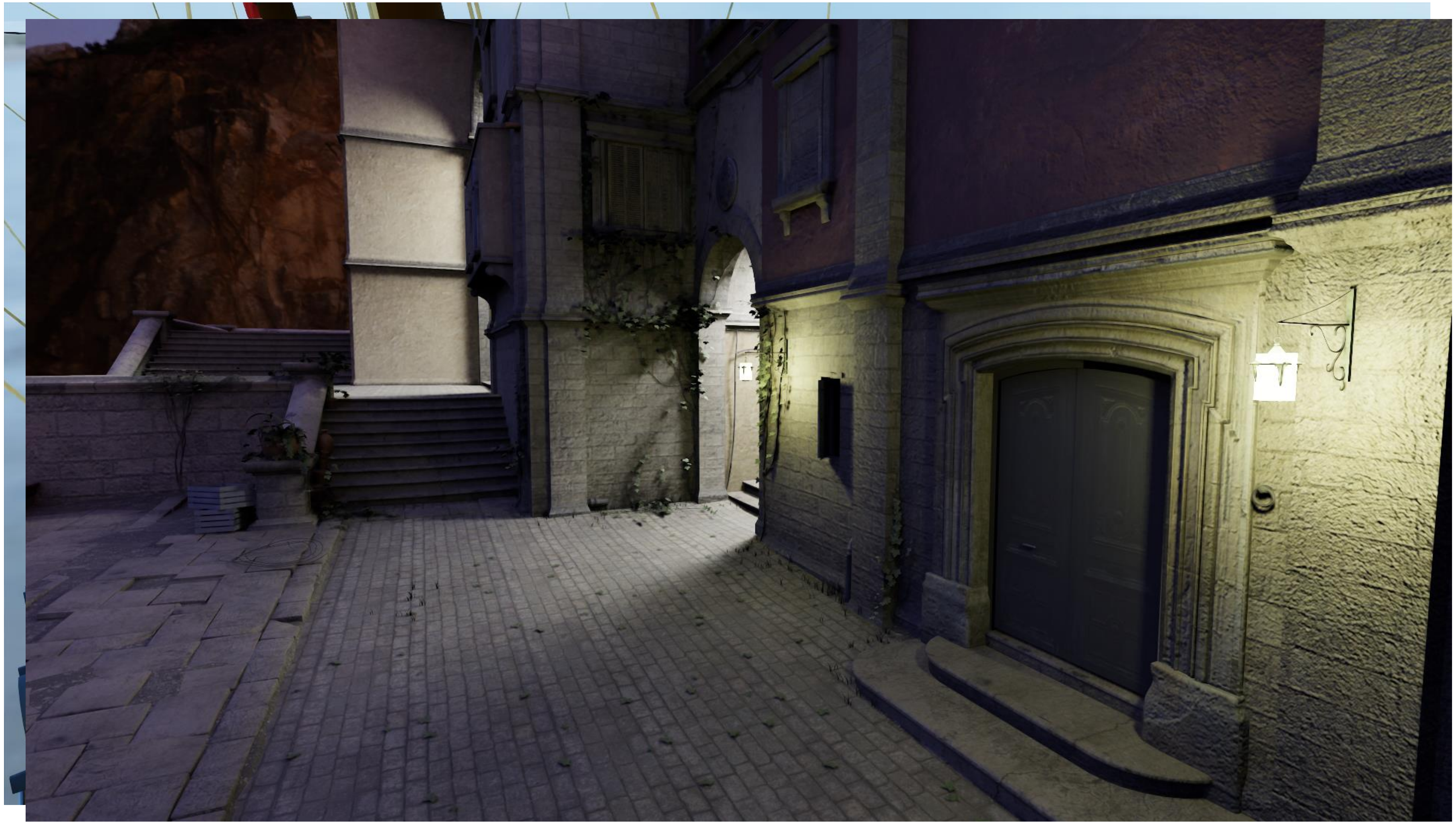
Statistical techniques remove many assumptions; assets become easier to author

5 minutes from unlit model to this

Suzanne’s Revenge from Blend Swap (user gregzaal)

PROMOTES PROMISE OF PATH TRACING IN REAL-TIME

Statistical techniques remove many assumptions; assets become easier to author

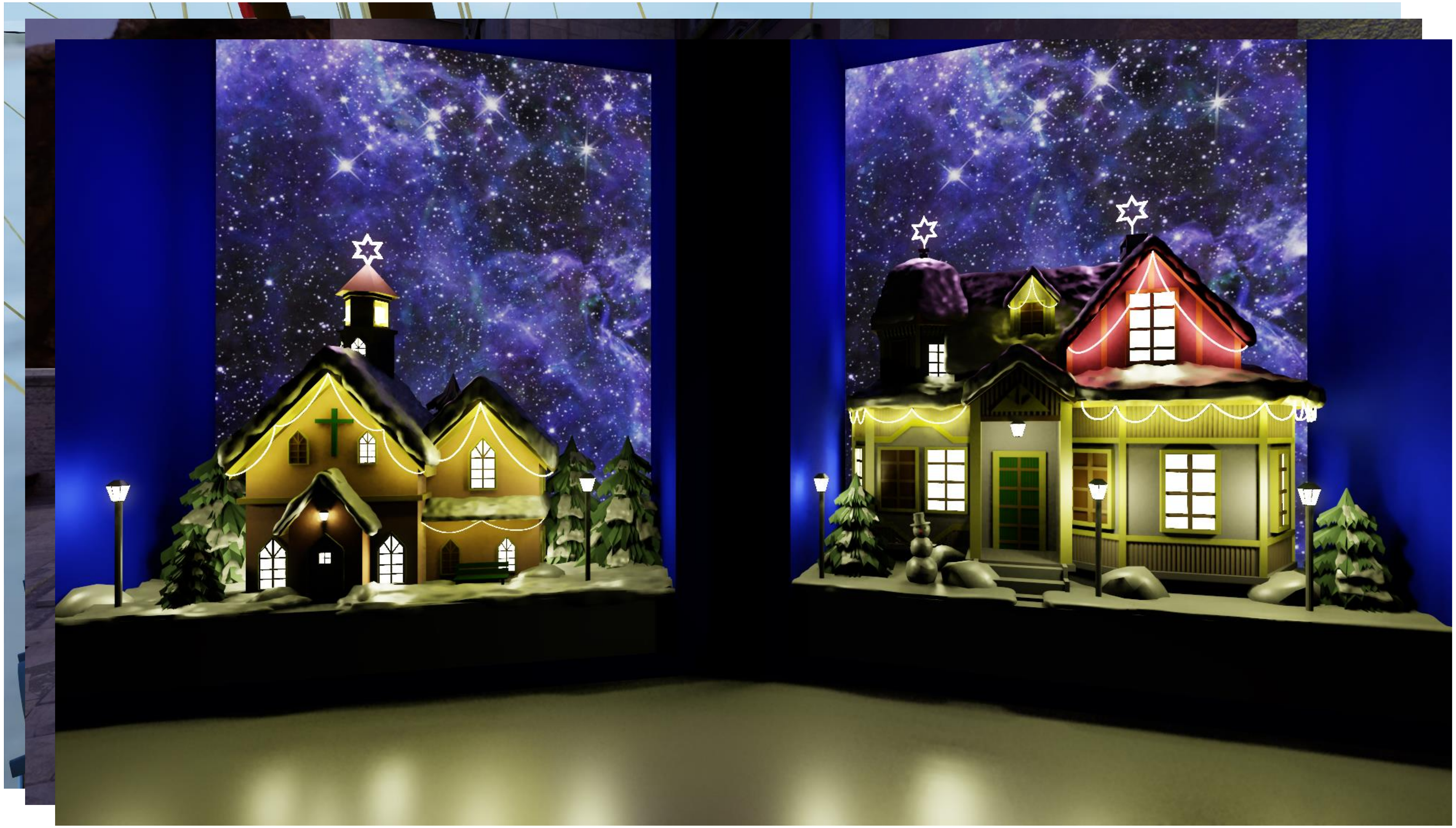


A tired 8+ year old asset gains new life

Tagged two matl's as emissive & added environment map

PROMOTES PROMISE OF PATH TRACING IN REAL-TIME

Statistical techniques remove many assumptions; assets become easier to author



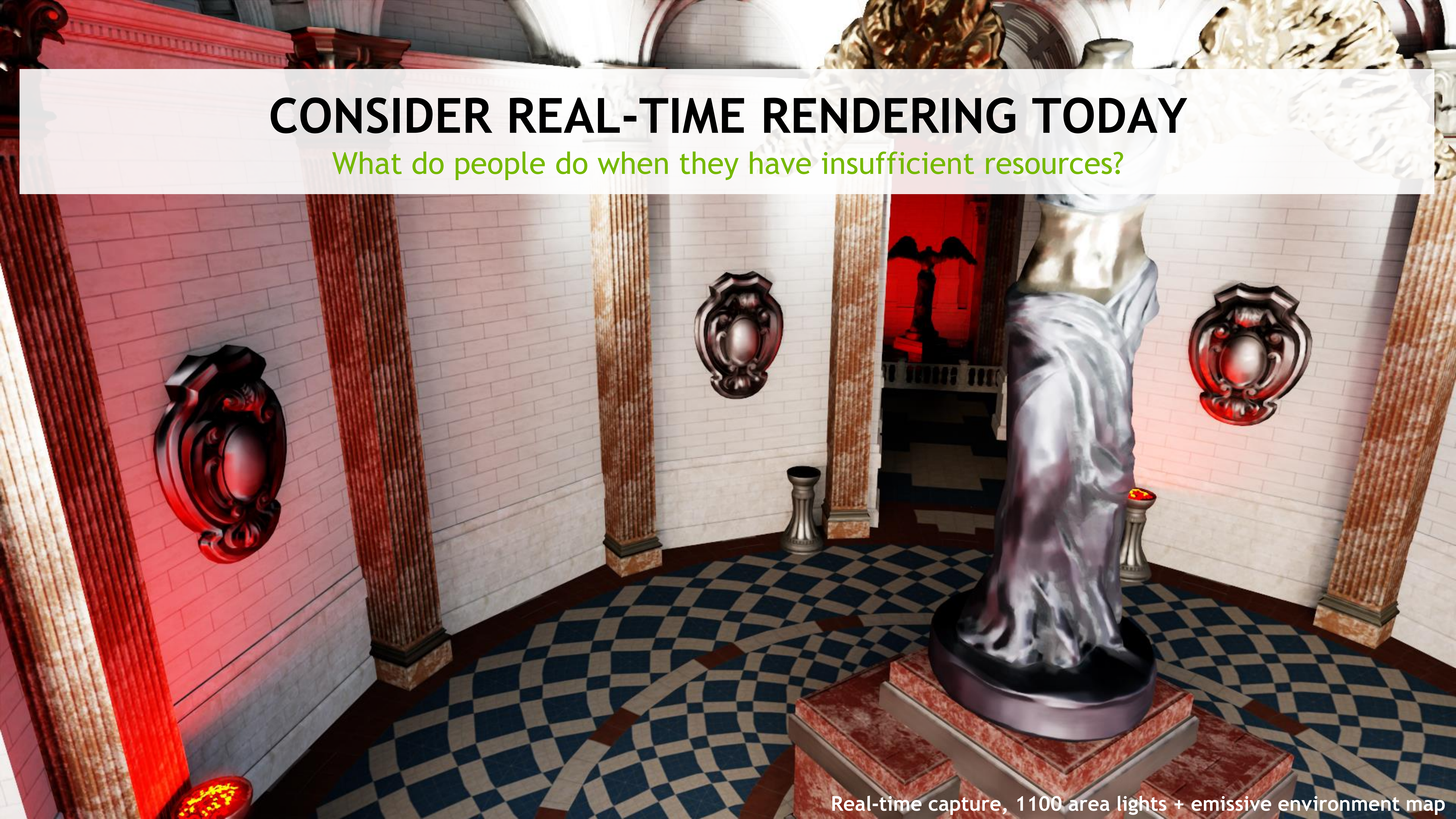
Asset I pulled from BlendSwap

(from user Mikel007)

Hardest part: converting model format to load

CONSIDER REAL-TIME RENDERING TODAY

What do people do when they have insufficient resources?



Real-time capture, 1100 area lights + emissive environment map

CONSIDER REAL-TIME RENDERING TODAY

What do people do when they have insufficient resources?

Spatial and temporal reuse of work

CONSIDER REAL-TIME RENDERING TODAY

What do people do when they have insufficient resources?

Spatial and temporal reuse of work

- Accumulation buffering / progressive rendering
- (Ir)radiance caching / light probes
- Antialiasing techniques
- Denoising and reconstruction filters
- Adaptive sampling
- … and probably many more

CONSIDER REAL-TIME RENDERING TODAY

What do people do when they have insufficient resources?

- Spatial and temporal reuse of work
  - Accumulation buffering / progressive rendering
  - (Ir)radiance caching / light probes
  - Antialiasing techniques
  - Denoising and reconstruction filters
  - Adaptive sampling
  - … and probably many more
- Also common in more offline contexts
  - E.g., path guiding, radiosity, etc.

WHAT SORTS OF THINGS GET REUSED?

WHAT SORTS OF THINGS GET REUSED?

Pixel colors	Antialiasing, post-process filtering, (typically) denoising filters
Texture colors	Light maps, environment maps
Colors (in other spaces)	Irradiance volumes

WHAT SORTS OF THINGS GET REUSED?

Pixel colors	Antialiasing, post-process filtering, (typically) denoising filters
Texture colors	Light maps, environment maps
Colors (in other spaces)	Irradiance volumes

During reuse, lost most non-color information

- **Easy to blur visibility**
- **Easy to blur specular highlights**
- **Easy to add bias**

WHAT SORTS OF THINGS GET REUSED?

Pixel colors	Antialiasing, post-process filtering, (typically) denoising filters
Texture colors	Light maps, environment maps
Colors (in other spaces)	Irradiance volumes
Projected radiance	Spherical harmonic coefficients

WHAT SORTS OF THINGS GET REUSED?

Pixel colors	Antialiasing, post-process filtering, (typically) denoising filters
Texture colors	Light maps, environment maps
Colors (in other spaces)	Irradiance volumes
Projected radiance	Spherical harmonic coefficients

More expensive to (pre-)compute

- **Reduces problems reusing color**
- **But not fully eliminated**

WHAT SORTS OF THINGS GET REUSED?

Pixel colors	Antialiasing, post-process filtering, (typically) denoising filters
Texture colors	Light maps, environment maps
Colors (in other spaces)	Irradiance volumes
Projected radiance	Spherical harmonic coefficients
Visibility	Shadow maps, form factors, ambient occlusion

WHAT SORTS OF THINGS GET REUSED?

Pixel colors	Antialiasing, post-process filtering, (typically) denoising filters
Texture colors	Light maps, environment maps
Colors (in other spaces)	Irradiance volumes
Projected radiance	Spherical harmonic coefficients
Visibility	Shadow maps, form factors, ambient occlusion

Visibility (mostly) always binary

- **Reuse = filtering**
- **Hard *not* to blur or alias**

WHAT SORTS OF THINGS GET REUSED?

Pixel colors	Antialiasing, post-process filtering, (typically) denoising filters
Texture colors	Light maps, environment maps
Colors (in other spaces)	Irradiance volumes
Projected radiance	Spherical harmonic coefficients
Visibility	Shadow maps, form factors, ambient occlusion
Ray segments	Bidirectional path tracing, light field rendering

WHAT SORTS OF THINGS GET REUSED?

Pixel colors	Antialiasing, post-process filtering, (typically) denoising filters
Texture colors	Light maps, environment maps
Colors (in other spaces)	Irradiance volumes
Projected radiance	Spherical harmonic coefficients
Visibility	Shadow maps, form factors, ambient occlusion
Ray segments	Bidirectional path tracing, light field rendering
Probabilities	Adaptive ray tracing, path guiding
Random variates	Primary sample space techniques

WHAT SORTS OF THINGS GET REUSED?

Pixel colors	Antialiasing, post-process filtering, (typically) denoising filters
Texture colors	Light maps, environment maps
Colors (in other spaces)	Irradiance volumes
Projected radiance	Spherical harmonic coefficients
Visibility	Shadow maps, form factors, ambient occlusion
Ray segments	Bidirectional path tracing, light field rendering
Probabilities	Adaptive ray tracing, path guiding
Random variates	Primary sample space techniques

Less well explored in real-time

- **Handles issues from above reuse**
- **Big question: Efficient algorithms?**

WHAT IS REUSE DOING?

Why is reusing computation reasonable?

WHAT IS REUSE DOING?

Why is reusing computation reasonable?

It assumes: "given small change in X, change in Y is __________"

Usually for "smooth", "small", or just "understandable" changes

WHAT IS REUSE DOING?

Why is reusing computation reasonable?

It assumes: "given small change in X, change in Y is __________"

Usually for "smooth", "small", or just "understandable" changes

When this assumption breaks, you get artifacts

Blurring, aliasing, lag, ghosting, halos, ringing

Generally, bias and noise

WHAT IS REUSE DOING?

Why is reusing computation reasonable?

It assumes: "given small change in X, change in Y is __________"

- Usually for "smooth", "small", or just "understandable" changes

When this assumption breaks, you get artifacts

- Blurring, aliasing, lag, ghosting, halos, ringing
- Generally, bias and noise

Assertion: Artifacts less correctable later in the rendering process

- Postprocessing final colors? Lost most data needed to correct for broken assumptions

OK, SO REUSE EARLY IN RENDERER

My suggestion: As part of importance sampling

OK, SO REUSE EARLY IN RENDERER

My suggestion: As part of importance sampling

Why?

Importance sampling function can be nearly arbitrary

If well chosen, convergence improves significantly (!/$ increase)

If poorly chosen, worst case added noise (!/$ decrease)

Rationale:

Good reuse, where assumptions hold, improves in quality significantly

A few places pixels become noisier, when reuse assumptions fail

OK, SO REUSE EARLY IN RENDERER

My suggestion: As part of importance sampling

Consider this reuse assumption:

Nearby pixels have similar probability to select a given light sample (or path vertex)

(Can probably replace “pixel” with “texel” or “voxel”)

OK, SO REUSE EARLY IN RENDERER

My suggestion: As part of importance sampling

Consider this reuse assumption:

Nearby pixels have similar probability to select a given light sample (or path vertex)

(Can probably replace “pixel” with “texel” or “voxel”)

This suggests reusing neighbors’ importance sampling functions

OK, SO REUSE EARLY IN RENDERER

My suggestion: As part of importance sampling

Consider this reuse assumption:

- Nearby pixels have similar probability to select a given light sample (or path vertex)
  - (Can probably replace "pixel" with "texel" or "voxel")

This suggests reusing neighbors' importance sampling functions

What does that even mean?

- If you construct a per-pixel sampling PDF, "aggregate" them to improve quality?

REMINDER: IMPORTANCE SAMPLING VITAL

Especially for real-time rendering

REMINDER: IMPORTANCE SAMPLING VITAL

Especially for real-time rendering

Importance sampled Monte Carlo integration:

$$\int f(x)\,dx \approx \frac{1}{N}\sum \frac{f(x_i)}{p(x_i)}$$

REMINDER: IMPORTANCE SAMPLING VITAL

Especially for real-time rendering

Importance sampled Monte Carlo integration:

$$\int f(x)\,dx \approx \frac{1}{N}\sum \frac{f(x_i)}{p(x_i)}$$

Better $p(x)$ means fewer samples N needed for convergence

- Truly vital to minimize sample count for real-time

REMINDER: IMPORTANCE SAMPLING VITAL

Especially for real-time rendering

Importance sampled Monte Carlo integration:

$$\int f(x)\,dx \approx \frac{1}{N}\sum \frac{f(x_i)}{p(x_i)}$$

Better $p(x)$ means fewer samples N needed for convergence

- Truly vital to minimize sample count for real-time

Idea: Target perfect importance sampling, $p(x) \propto f(x)$? Reduces samples needed to one

- But when importance sampling is taught, the idea is handwaved away

WHAT DOES PERFECT IMPORTANCE SAMPLING MEAN?

And should we really ignore it out of hand?

If $p(x) \propto f(x)$, this means $p(x) = cf(x)$ for some normalization constant

- Since for any PDF, $\int p(x)\,dx \equiv 1$, it's easy to show $c = {}^{1}/_{\int f(x)\,dx}$

WHAT DOES PERFECT IMPORTANCE SAMPLING MEAN?

And should we really ignore it out of hand?

If $p(x) \propto f(x)$, this means $p(x) = cf(x)$ for some normalization constant

Since for any PDF, $\int p(x)\,dx \equiv 1$, it's easy to show $c = {}^{1}/_{\int f(x)\,dx}$

This is where the handwaving starts, e.g., from the 3rd edition of PBRT:

Of course, this is ludicrous since we wouldn't bother using Monte Carlo if we could integrate f directly. However, if a density $p(x)$ can be found that is similar

WHAT DOES PERFECT IMPORTANCE SAMPLING MEAN?

And should we really ignore it out of hand?

If $p(x) \propto f(x)$, this means $p(x) = cf(x)$ for some normalization constant

Since for any PDF, $\int p(x)\,dx \equiv 1$, it's easy to show $c = {}^{1}/_{\int f(x)\,dx}$

This is where the handwaving starts, e.g., from the 3rd edition of PBRT:

> Of course, this is ludicrous since we wouldn't bother using Monte Carlo if we could integrate f directly. However, if a density $p(x)$ can be found that is similar

But… If we're using Monte Carlo integration, why would an integral bother us?

We're already approximating, so can we just apply more Monte Carlo?

RESAMPLED IMPORTANCE SAMPLING (RIS)

Or my informal name: "Approximately perfect importance sampling"

RESAMPLED IMPORTANCE SAMPLING (RIS)

Or my informal name: "Approximately perfect importance sampling"

Imagine you have unnormalized $\hat{p}(x)$ you want to use for importance sampling

- This would be $\hat{p}(x) = f(x)$ for "perfect" importance sampling

RESAMPLED IMPORTANCE SAMPLING (RIS)

Or my informal name: "Approximately perfect importance sampling"

Imagine you have unnormalized $\hat{p}(x)$ you want to use for importance sampling

This would be $\hat{p}(x) = f(x)$ for "perfect" importance sampling

To normalize, you have $p(x) = \frac{\hat{p}(x)}{\int \hat{p}(x)\, dx}$

RESAMPLED IMPORTANCE SAMPLING (RIS)

Or my informal name: "Approximately perfect importance sampling"

Imagine you have unnormalized $\hat{p}(x)$ you want to use for importance sampling

This would be $\hat{p}(x) = f(x)$ for "perfect" importance sampling

To normalize, you have $p(x) = \frac{\hat{p}(x)}{\int \hat{p}(x)\,dx} \approx \frac{\hat{p}(x)}{\frac{1}{M}\sum\frac{\hat{p}(x_j)}{q(x_j)}}$ using a Monte Carlo estimate of $\int \hat{p}(x)\,dx$

RESAMPLED IMPORTANCE SAMPLING (RIS)

Or my informal name: "Approximately perfect importance sampling"

Imagine you have unnormalized $\hat{p}(x)$ you want to use for importance sampling

This would be $\hat{p}(x) = f(x)$ for "perfect" importance sampling

To normalize, you have $p(x) = \frac{\hat{p}(x)}{\int \hat{p}(x)\, dx} \approx \frac{\hat{p}(x)}{\frac{1}{M}\sum \frac{\hat{p}(x_j)}{q(x_j)}}$ using a Monte Carlo estimate of $\int \hat{p}(x)\, dx$

Plug $p(x)$ back into the original Monte Carlo estimator and you get:

$$\int f(x)\, dx \approx \frac{1}{N}\sum \left[\frac{f(x_i)}{\hat{p}(x_i)} \frac{1}{M} \sum \frac{\hat{p}(x_j)}{q(x_j)} \right]$$

RESAMPLED IMPORTANCE SAMPLING (RIS)

Or my informal name: "Approximately perfect importance sampling"

Imagine you have unnormalized $\hat{p}(x)$ you want to use for importance sampling

This would be $\hat{p}(x) = f(x)$ for "perfect" importance sampling

To normalize, you have $p(x) = \frac{\hat{p}(x)}{\int \hat{p}(x)\, dx} \approx \frac{\hat{p}(x)}{\frac{1}{M}\sum \frac{\hat{p}(x_j)}{q(x_j)}}$ using a Monte Carlo estimate of $\int \hat{p}(x)\, dx$

Plug $p(x)$ back into the original Monte Carlo estimator and you get:

$$\int f(x)\, dx \approx \frac{1}{N} \sum \left[\frac{f(x_i)}{\hat{p}(x_i)} \frac{1}{M} \sum \frac{\hat{p}(x_j)}{q(x_j)} \right]$$

This is exactly the RIS estimator

See Talbot et al., EGSR 2005

WHAT IS RIS?

Not well covered in books and courses; I recommend Talbot's paper and dissertation

WHAT IS RIS?

Not well covered in books and courses; I recommend Talbot's paper and dissertation

$$\int f(x)\,dx \approx \frac{1}{N}\sum\left[\frac{f(x_i)}{\hat{p}(x_i)}\,\frac{1}{M}\sum\frac{\hat{p}(x_j)}{q(x_j)}\right]$$

Take M samples from distribution $q(x)$, turn into N samples from (unnormalized) distribution $\hat{p}(x)$

(cheap, low-quality, or simply bad) *(complex, high-quality, or hard-to-sample)*

WHAT IS RIS?

Not well covered in books and courses; I recommend Talbot's paper and dissertation

$$\int f(x)\,dx \approx \frac{1}{N}\sum\left[\frac{f(x_i)}{\hat{p}(x_i)}\,\frac{1}{M}\sum\frac{\hat{p}(x_j)}{q(x_j)}\right]$$

Take M samples from distribution $q(x)$, turn into N samples from (unnormalized) distribution $\hat{p}(x)$

Typically $N \ll M$, and the N samples become higher quality (i.e., closer to $\hat{p}(x)$) as $M \to \infty$

WHAT IS RIS?

Not well covered in books and courses; I recommend Talbot's paper and dissertation

$$\int f(x)\,dx \approx \frac{1}{N}\sum\left[\frac{f(x_i)}{\hat{p}(x_i)}\,\frac{1}{M}\sum\frac{\hat{p}(x_j)}{q(x_j)}\right]$$

Take M samples from distribution $q(x)$, turn into N samples from (unnormalized) distribution $\hat{p}(x)$

Typically $N \ll M$, and the N samples become higher quality (i.e., closer to $\hat{p}(x)$) as $M \to \infty$

If $M = 1$, equivalent to sampling $q(x)$; if $M = \infty$, equivalent to sampling $\hat{p}(x)$

BENEFITS FOR REAL-TIME RENDERING

(I.e., why drag you through this math?)

Imagine a more complex function:

$$\int S(x)\, V(x)\, dx$$

BENEFITS FOR REAL-TIME RENDERING

(I.e., why drag you through this math?)

Imagine a more complex function:

$$\int S(x)\, V(x)\, dx \approx \frac{1}{N} \sum \left[\frac{S(x_i) V(x_i)}{S(x_i)} \frac{1}{M} \sum \frac{S(x_j)}{p(x_j)} \right]$$

Might pick target function $\hat{p}(x)$ using only part of the integrand, e.g., $\hat{p}(x) = S(x)$

BENEFITS FOR REAL-TIME RENDERING

(I.e., why drag you through this math?)

Imagine a more complex function:

$$\int S(x)\, V(x)\, dx \approx \frac{1}{N}\sum\left[\frac{S(x_i)V(x_i)}{S(x_i)}\ \frac{1}{M}\sum\frac{S(x_j)}{p(x_j)}\right] = \frac{1}{N}\sum\left[V(x_i)\ \frac{1}{M}\sum\frac{S(x_j)}{p(x_j)}\right]$$

Might pick target function $\hat{p}(x)$ using only part of the integrand, e.g., $\hat{p}(x) = S(x)$

BENEFITS FOR REAL-TIME RENDERING

(I.e., why drag you through this math?)

Imagine a more complex function:

$$\int S(x)\,V(x)\,dx \approx \frac{1}{N}\sum\left[\frac{S(x_i)V(x_i)}{S(x_i)}\,\frac{1}{M}\sum\frac{S(x_j)}{p(x_j)}\right] = \frac{1}{N}\sum\left[V(x_i)\,\frac{1}{M}\sum\frac{S(x_j)}{p(x_j)}\right]$$

Might pick target function $\hat{p}(x)$ using only part of the integrand, e.g., $\hat{p}(x) = S(x)$

Cleanly decomposes integrand into pieces that get ***evaluated at different frequencies***!

- Very valuable if $S(x)$ and $V(x)$ have very different costs
- This allows computing N rays per pixel using $\mathrm{M} \gg N$ (cheaper) shades
- The larger M, the better quality your N samples

BENEFITS FOR REAL-TIME RENDERING

(I.e., why drag you through this math?)

Can apply RIS multiple times; imagine picking $\hat{p}(x) = f_r(x)G(x)L(x)$:

$$\int f_r(x)\, G(x)\, L(x)\, V(x)\, dx \approx \frac{1}{N}\sum\left[V(x_i)\ \frac{1}{M}\sum\frac{f_r(x_j)\, G(x_j)\, L(x_j)}{p(x_j)}\right]$$

BENEFITS FOR REAL-TIME RENDERING

(I.e., why drag you through this math?)

Can apply RIS multiple times; imagine picking $\hat{p}(x) = f_r(x)G(x)L(x)$:

$$\int f_r(x)\, G(x)\, L(x)\, V(x)\, dx \approx \frac{1}{N}\sum \left[V(x_i)\ \underbrace{\frac{1}{M}\sum \frac{f_r(x_j)\, G(x_j)\, L(x_j)}{p(x_j)}} \right]$$

Still a legitimately hard integral, evaluated with Monte Carlo

Perhaps apply RIS again, to get a good $p(x)$?

BENEFITS FOR REAL-TIME RENDERING

(I.e., why drag you through this math?)

Can apply RIS multiple times; imagine picking $\hat{p}(x) = f_r(x)G(x)L(x)$:

$$\int f_r(x)\, G(x)\, L(x)\, V(x)\, dx \approx \frac{1}{N}\sum\left[V(x_i)\ \frac{1}{M}\sum\frac{f_r(x_j)\, G(x_j)\, L(x_j)}{p(x_j)}\right]$$

Let's use a second target function $p(x) = G(x)L(x)$:

$$\int f_r(x)\, G(x)\, L(x)\, V(x)\, dx \approx \frac{1}{N}\sum\left[V(x_i)\ \frac{1}{M}\sum\left[\frac{f_r(x_j)\, G(x_j)\, L(x_j)}{G(x_j)\, L(x_j)}\ \frac{1}{K}\sum\frac{G(x_k)\, L(x_k)}{q(x_k)}\right]\right]$$

BENEFITS FOR REAL-TIME RENDERING

(I.e., why drag you through this math?)

Can apply RIS multiple times; imagine picking $\hat{p}(x) = f_r(x)G(x)L(x)$:

$$\int f_r(x)\, G(x)\, L(x)\, V(x)\, dx \approx \frac{1}{N}\sum\left[V(x_i)\ \frac{1}{M}\sum\frac{f_r(x_j)\, G(x_j)\, L(x_j)}{p(x_j)}\right]$$

Let's use a second target function $p(x) = G(x)L(x)$:

$$\int f_r(x)\, G(x)\, L(x)\, V(x)\, dx \approx \frac{1}{N}\sum\left[V(x_i)\ \frac{1}{M}\sum\left[\frac{f_r(x_j)\, \cancel{G(x_j)\, L(x_j)}}{\cancel{G(x_j)\, L(x_j)}}\ \frac{1}{K}\sum\frac{G(x_k)\, L(x_k)}{q(x_k)}\right]\right]$$

BENEFITS FOR REAL-TIME RENDERING

(I.e., why drag you through this math?)

Can apply RIS multiple times; imagine picking $\hat{p}(x) = f_r(x)G(x)L(x)$:

$$\int f_r(x)\, G(x)\, L(x)\, V(x)\, dx \approx \frac{1}{N}\sum\left[V(x_i)\ \frac{1}{M}\sum\frac{f_r(x_j)\, G(x_j)\, L(x_j)}{p(x_j)}\right]$$

Let's use a second target function $p(x) = G(x)L(x)$:

$$\int f_r(x)\, G(x)\, L(x)\, V(x)\, dx \approx \frac{1}{N}\sum\left[V(x_i)\ \frac{1}{M}\sum\left[f_r(x_j)\ \frac{1}{K}\sum\frac{G(x_k)\, L(x_k)}{q(x_k)}\right]\right]$$

BENEFITS FOR REAL-TIME RENDERING

(I.e., why drag you through this math?)

Can apply RIS multiple times; imagine picking $\hat{p}(x) = f_r(x)G(x)L(x)$:

$$\int f_r(x)\, G(x)\, L(x)\, V(x)\, dx \approx \frac{1}{N}\sum\left[V(x_i)\ \frac{1}{M}\sum\frac{f_r(x_j)\, G(x_j)\, L(x_j)}{p(x_j)}\right]$$

Let's use a second target function $p(x) = G(x)L(x)$:

$$\int f_r(x)\, G(x)\, L(x)\, V(x)\, dx \approx \frac{1}{N}\sum\left[V(x_i)\ \frac{1}{M}\sum\left[f_r(x_j)\ \frac{1}{K}\sum\frac{G(x_k)\, L(x_k)}{q(x_k)}\right]\right]$$

And now, it's pretty easy to sample $q(x) \sim L(x)$, which allows further cancellation

Able to split numerical integration so ***each term*** evaluated with ***different sampling rate***!

IMPORTANCE RESAMPLING TAKEAWAYS

Why you should teach / learn RIS

Real-time capture , 4200 area lights

IMPORTANCE RESAMPLING TAKEAWAYS

Why you should teach / learn RIS

- Converts M cheap samples into N of better quality
  - Allows sampling from (nearly) arbitrary functions in an unbiased way

IMPORTANCE RESAMPLING TAKEAWAYS

Why you should teach / learn RIS

Converts M cheap samples into N of better quality

- Allows sampling from (nearly) arbitrary functions in an unbiased way

My mental model: “approximate” perfect importance sampling

- As $M \to \infty$, you approach ideal sampling of your target distribution

IMPORTANCE RESAMPLING TAKEAWAYS

Why you should teach / learn RIS

- Converts M cheap samples into N of better quality
  - Allows sampling from (nearly) arbitrary functions in an unbiased way
- My mental model: “approximate” perfect importance sampling
  - As $M \rightarrow \infty$, you approach ideal sampling of your target distribution
- Apply RIS in multiple stages
  - Lots of flexibility for designing algorithms

IMPORTANCE RESAMPLING TAKEAWAYS

Why you should teach / learn RIS

Converts M cheap samples into N of better quality

- Allows sampling from (nearly) arbitrary functions in an unbiased way

My mental model: “approximate” perfect importance sampling

- As $M \rightarrow \infty$, you approach ideal sampling of your target distribution

Apply RIS in multiple stages

- Lots of flexibility for designing algorithms

Decouple computational frequencies

- Use cheap-to-compute terms to improve placement of expensive operations

TAKING $M \to \infty$

For RIS, you want large pools of candidate samples

What if we target M = 10,000 candidates to pick N = 1 sample…

TAKING $M \rightarrow \infty$

For RIS, you want large pools of candidate samples

What if we target M = 10,000 candidates to pick N = 1 sample…

- Do you need to keep all 10,000 candidates resident in memory?
- If so… limits increases to M
- Most people today keep candidates resident
- Compute discrete CDF, invert to draw samples

TAKING $M \to \infty$

For RIS, you want large pools of candidate samples

What if we target M = 10,000 candidates to pick N = 1 sample…

- Do you need to keep all 10,000 candidates resident in memory?
- If so… limits increases to M
- Most people today keep candidates resident
- Compute discrete CDF, invert to draw samples

Why keep 10,000 in memory if returning only one?

- Why not discard incrementally?
- Keep just N candidates in memory at a given time

WEIGHTED RESERVOIR SAMPLING

Do yourself a favor; learn this algorithm and keep it near the top of your toolbox

WEIGHTED RESERVOIR SAMPLING

Do yourself a favor; learn this algorithm and keep it near the top of your toolbox

Well known algorithm from early 1980s

- Data often stored on reel-to-reel tape
- Big pain if rewind needed for algorithm's 2nd pass over data

WEIGHTED RESERVOIR SAMPLING

Do yourself a favor; learn this algorithm and keep it near the top of your toolbox

Well known algorithm from early 1980s

- Data often stored on reel-to-reel tape
- Big pain if rewind needed for algorithm's 2nd pass over data
- Streaming selection of arbitrary weighted samples
- Constant memory, one pass over data, size need not be known

WEIGHTED RESERVOIR SAMPLING

Do yourself a favor; learn this algorithm and keep it near the top of your toolbox

Well known algorithm from early 1980s

- Data often stored on reel-to-reel tape
- Big pain if rewind needed for algorithm's 2^{nd} pass over data
- Streaming selection of arbitrary weighted samples
- Constant memory, one pass over data, size need not be known

Easy proof by induction

- Multiple people have reinvented this wheel
- But go read up on 40 years of theory

WEIGHTED RESERVOIR SAMPLING

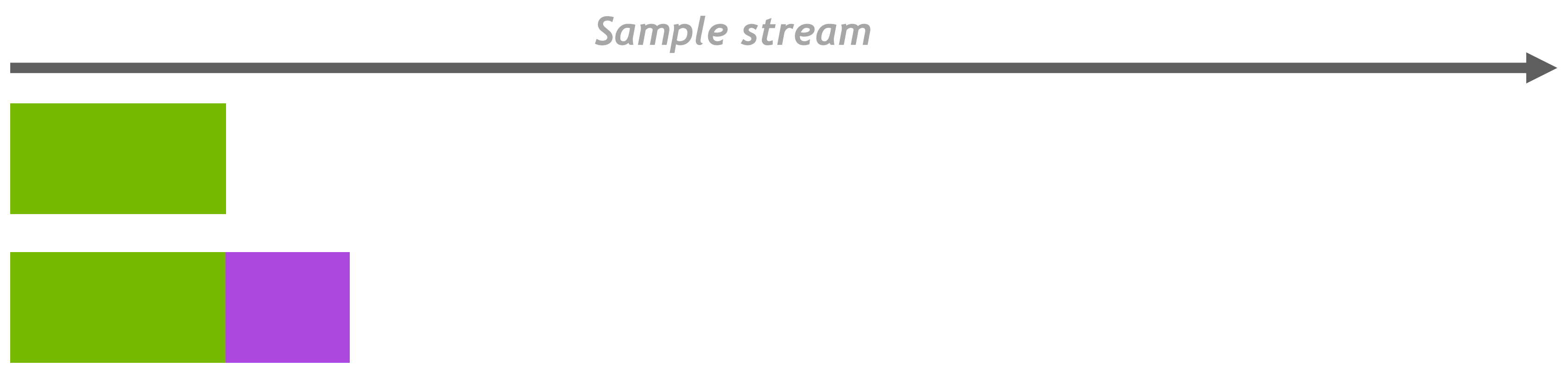
Do yourself a favor; learn this algorithm and keep it near the top of your toolbox

Always pick first sample encountered with weight > 0

WEIGHTED RESERVOIR SAMPLING

Do yourself a favor; learn this algorithm and keep it near the top of your toolbox

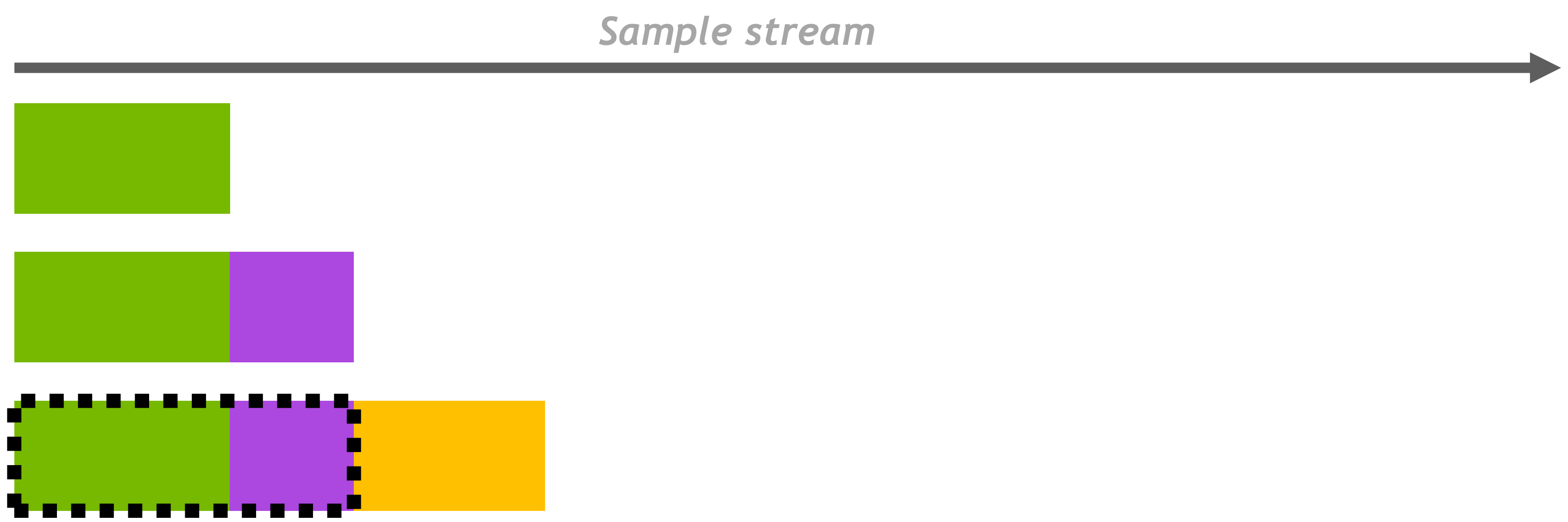
Discard old, choose new sample with probability $w_{purple} / (w_{purple} + w_{green})$



WEIGHTED RESERVOIR SAMPLING

Do yourself a favor; learn this algorithm and keep it near the top of your toolbox

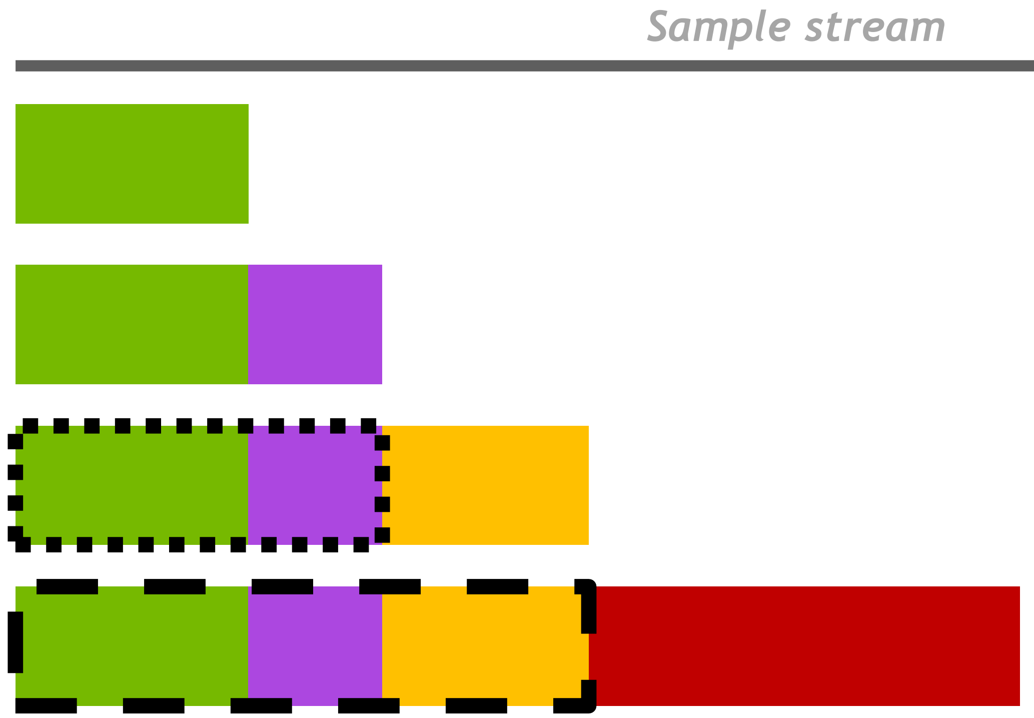
Choose new sample with probability $w_{orange} / (w_{orange} + w_{dotted})$



WEIGHTED RESERVOIR SAMPLING

Do yourself a favor; learn this algorithm and keep it near the top of your toolbox

Choose new sample with probability $w_{red} / (w_{red} + w_{dashed})$



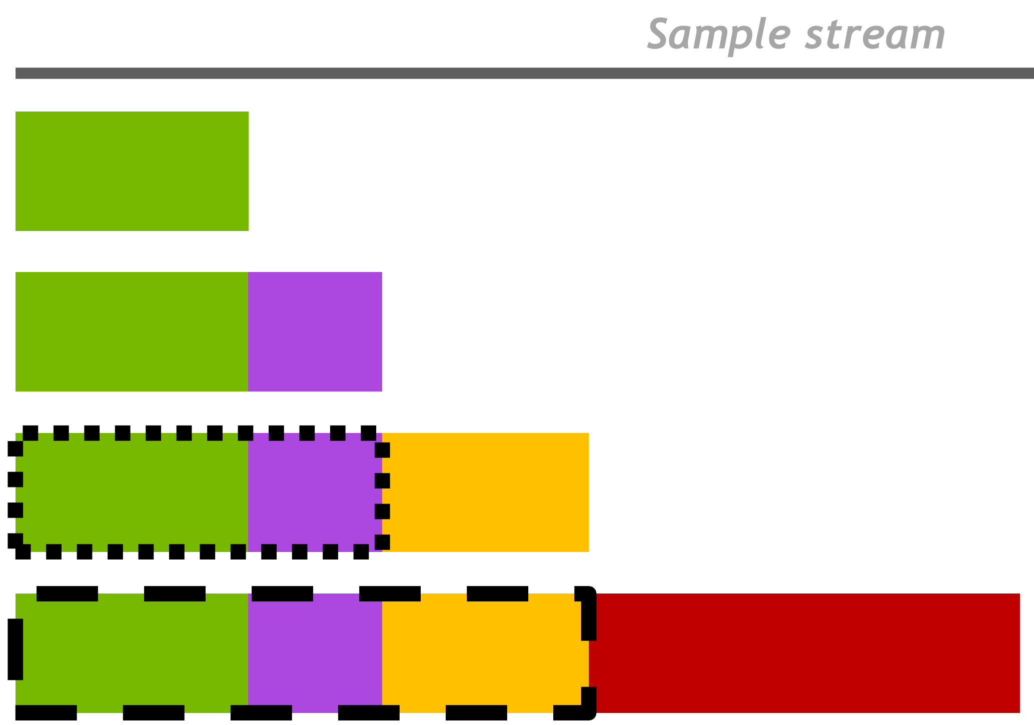
Just an argument about proportionality

w_{purple} / w_{green} same throughout stream

WEIGHTED RESERVOIR SAMPLING

Do yourself a favor; learn this algorithm and keep it near the top of your toolbox

Whenever stream stops, just output current selection as final answer



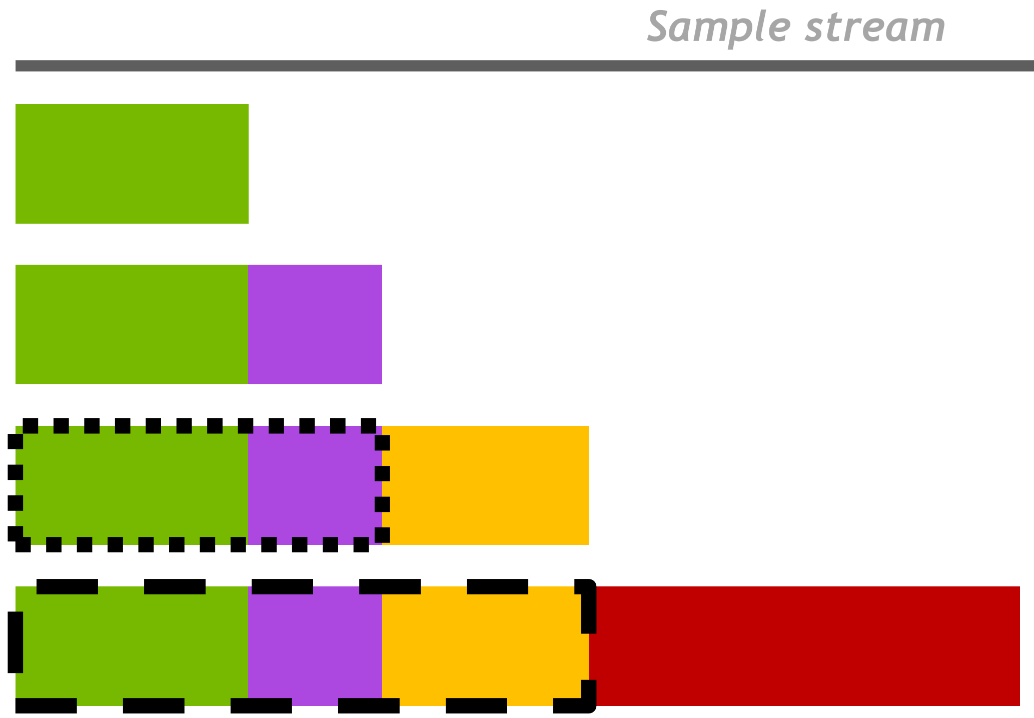
Decision at each step only needs:

Sum of prior weights, new element weight

WEIGHTED RESERVOIR SAMPLING

Do yourself a favor; learn this algorithm and keep it near the top of your toolbox

Whenever stream stops, just output current selection as final answer



Decision at each step only needs:

Sum of prior weights, new element weight

$$\int f(x)\,dx \approx \frac{1}{N}\sum\left[\frac{f(x_i)}{\hat{p}(x_i)}\frac{1}{M}\sum\frac{\hat{p}(x_j)}{q(x_j)}\right]$$

WEIGHTED RESERVOIR SAMPLING

Very simple, but also very powerful

- Allows using RIS without memory limiting sample count
  - Memory O(N), and for real-time this is generally a small constant

WEIGHTED RESERVOIR SAMPLING

Very simple, but also very powerful

Allows using RIS without memory limiting sample count

- Memory O(N), and for real-time this is generally a small constant

Unfortunately, RIS is still O(M) compute, and we want $M \to \infty$ …

WEIGHTED RESERVOIR SAMPLING

Very simple, but also very powerful

Allows using RIS without memory limiting sample count

Memory O(N), and for real-time this is generally a small constant

Unfortunately, RIS is still O(M) compute, and we want $M \to \infty$ …

But, weighted reservoir sampling provides another benefit

WEIGHTED RESERVOIR SAMPLING

Very simple, but also very powerful

Can combine two independent streams of samples

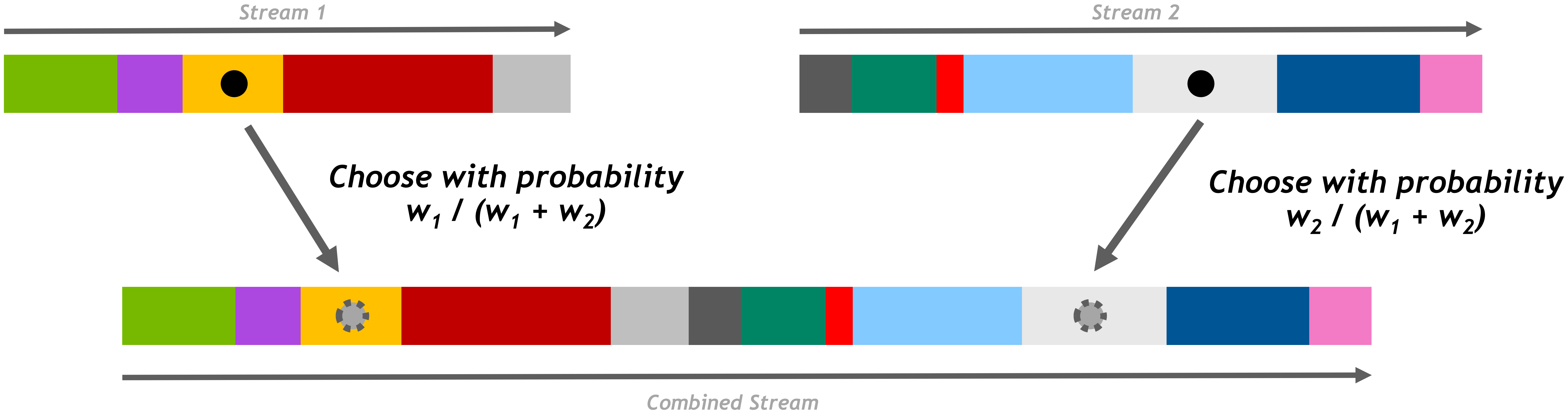
Without reprocessing individual samples!

WEIGHTED RESERVOIR SAMPLING

Very simple, but also very powerful

Can combine two independent streams of samples

Without reprocessing individual samples!

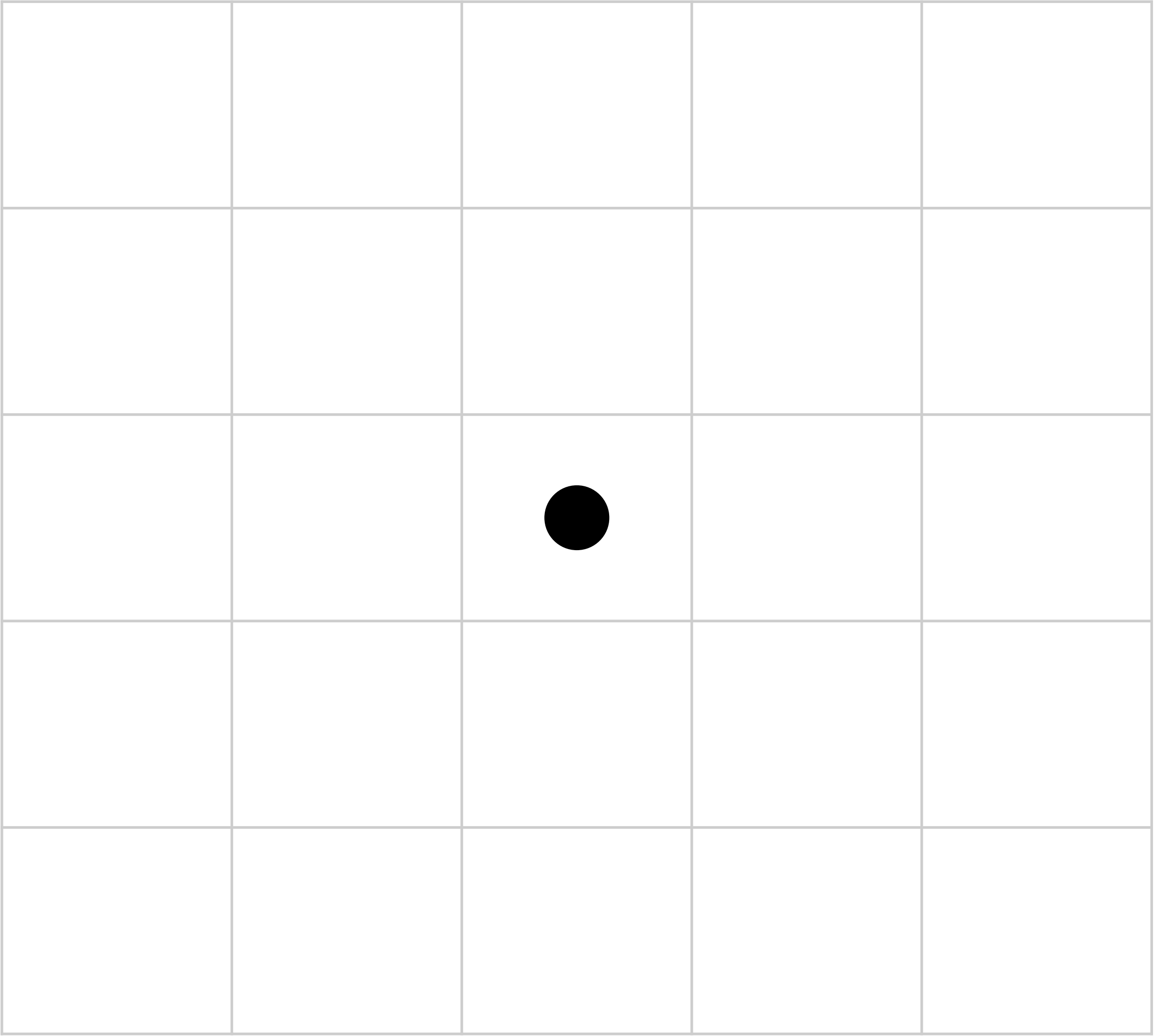


WHY IS THIS USEFUL?

Allows the spatiotemporal reuse we desire

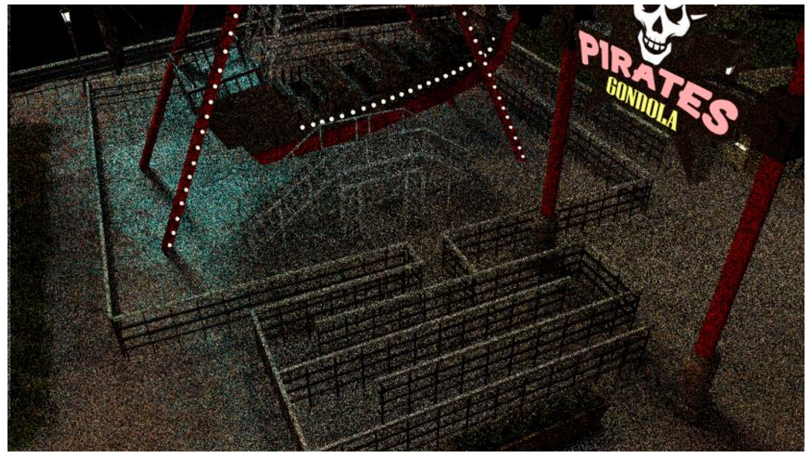
WHY IS THIS USEFUL?

Allows the spatiotemporal reuse we desire



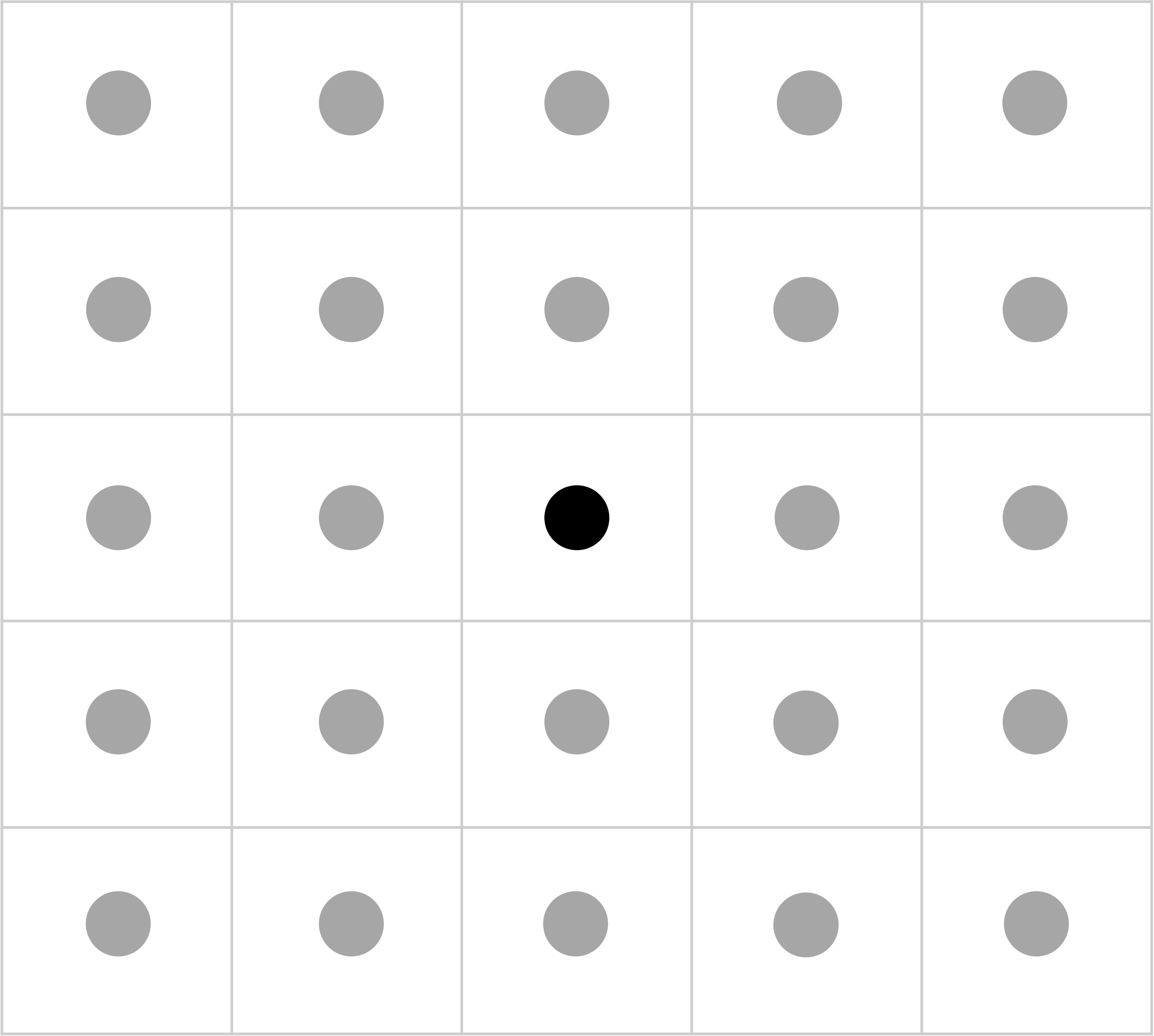
At each pixel, say we select 1 light from 32 random samples

Using RIS and weighted reservoir sampling



WHY IS THIS USEFUL?

Allows the spatiotemporal reuse we desire

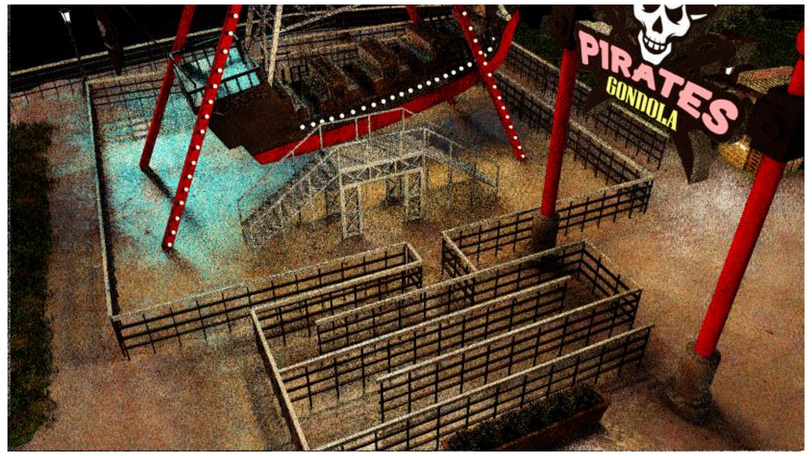


At each pixel, say we select 1 light from 32 random samples

- Using RIS and weighted reservoir sampling

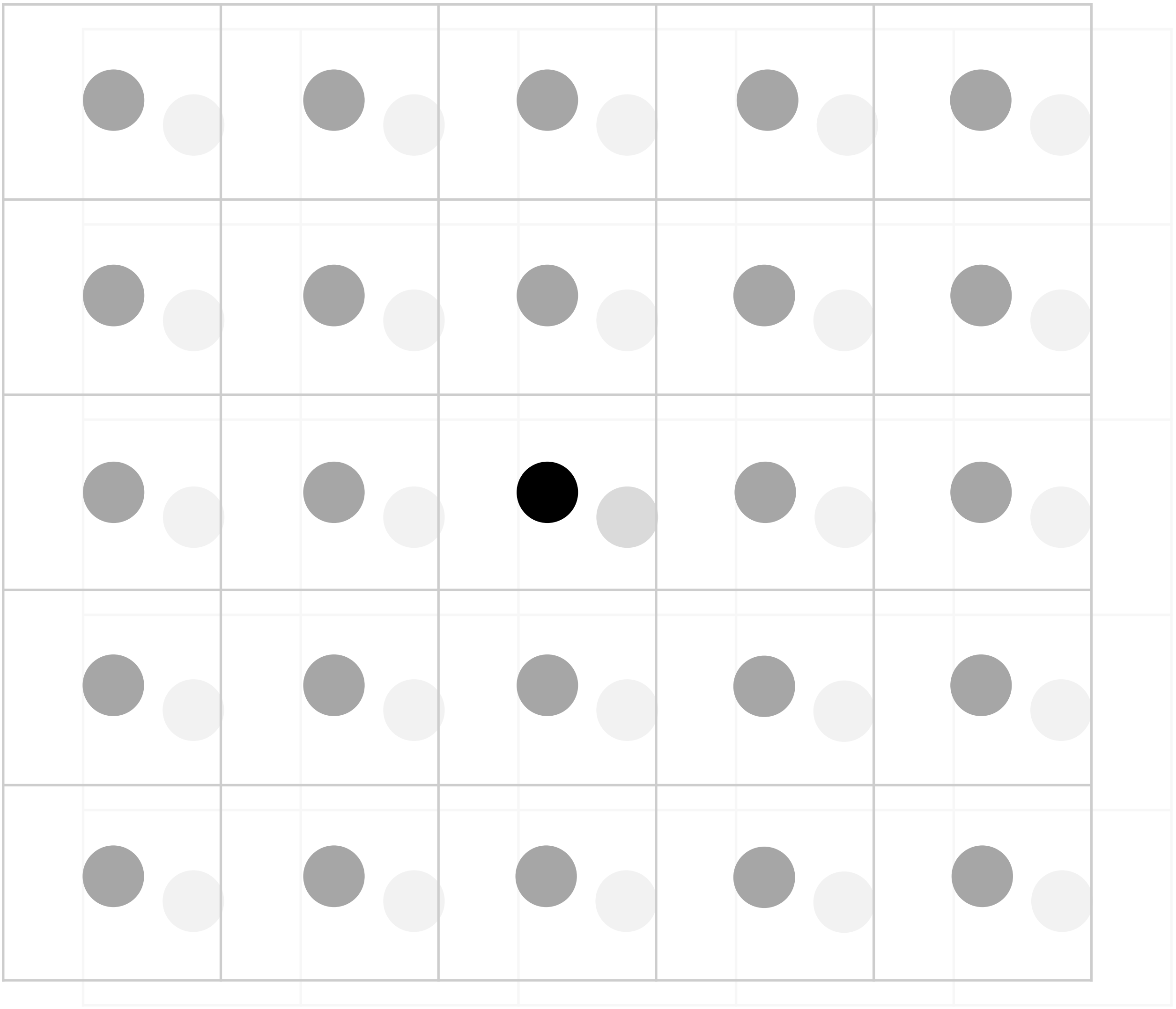
At a cost of merging these 25 reservoirs

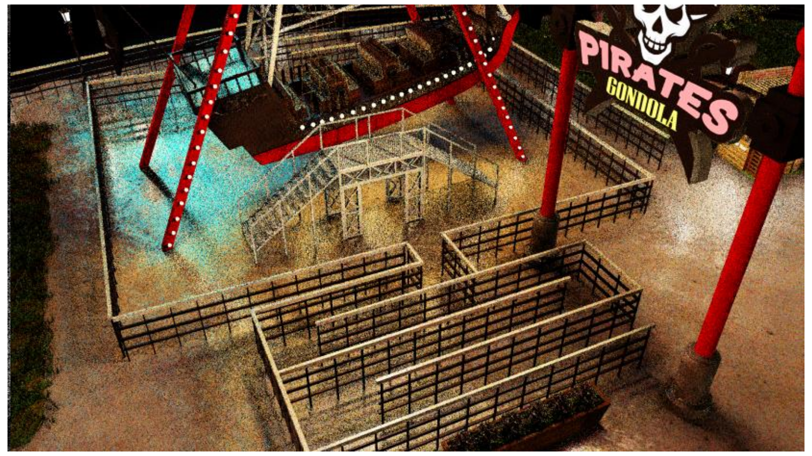
- We select 1 light from (25 x 32 =) 800 effective samples



WHY IS THIS USEFUL?

Allows the spatiotemporal reuse we desire





At each pixel, say we select 1 light from 32 random samples

- Using RIS and weighted reservoir sampling

At a cost of merging these 25 reservoirs

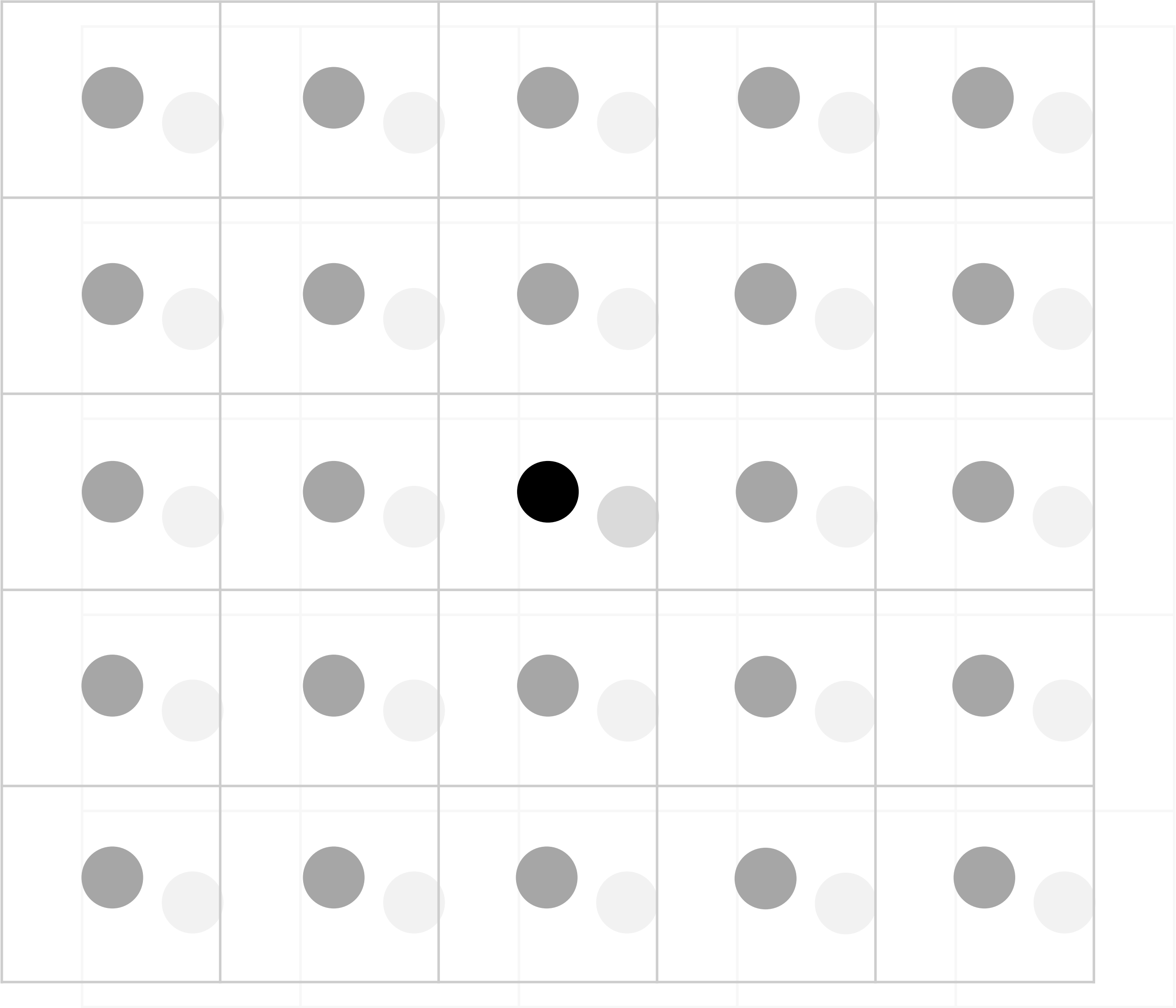
- We select 1 light from (25 x 32 =) 800 effective samples

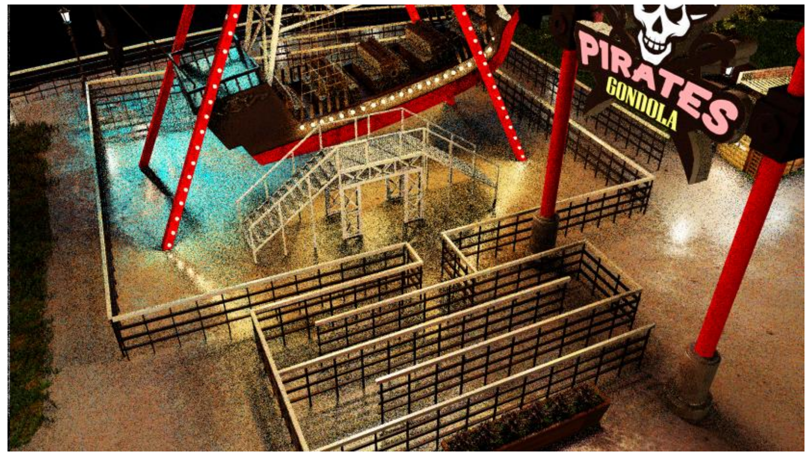
Merge with our reservoir from last frame

- We select 1 light from (2 x 800 =) 1600 effective samples

WHY IS THIS USEFUL?

Allows the spatiotemporal reuse we desire





At each pixel, say we select 1 light from 32 random samples

- Using RIS and weighted reservoir sampling

At a cost of merging these 25 reservoirs

- We select 1 light from (25 x 32 =) 800 effective samples

Merge with our reservoir from last frame

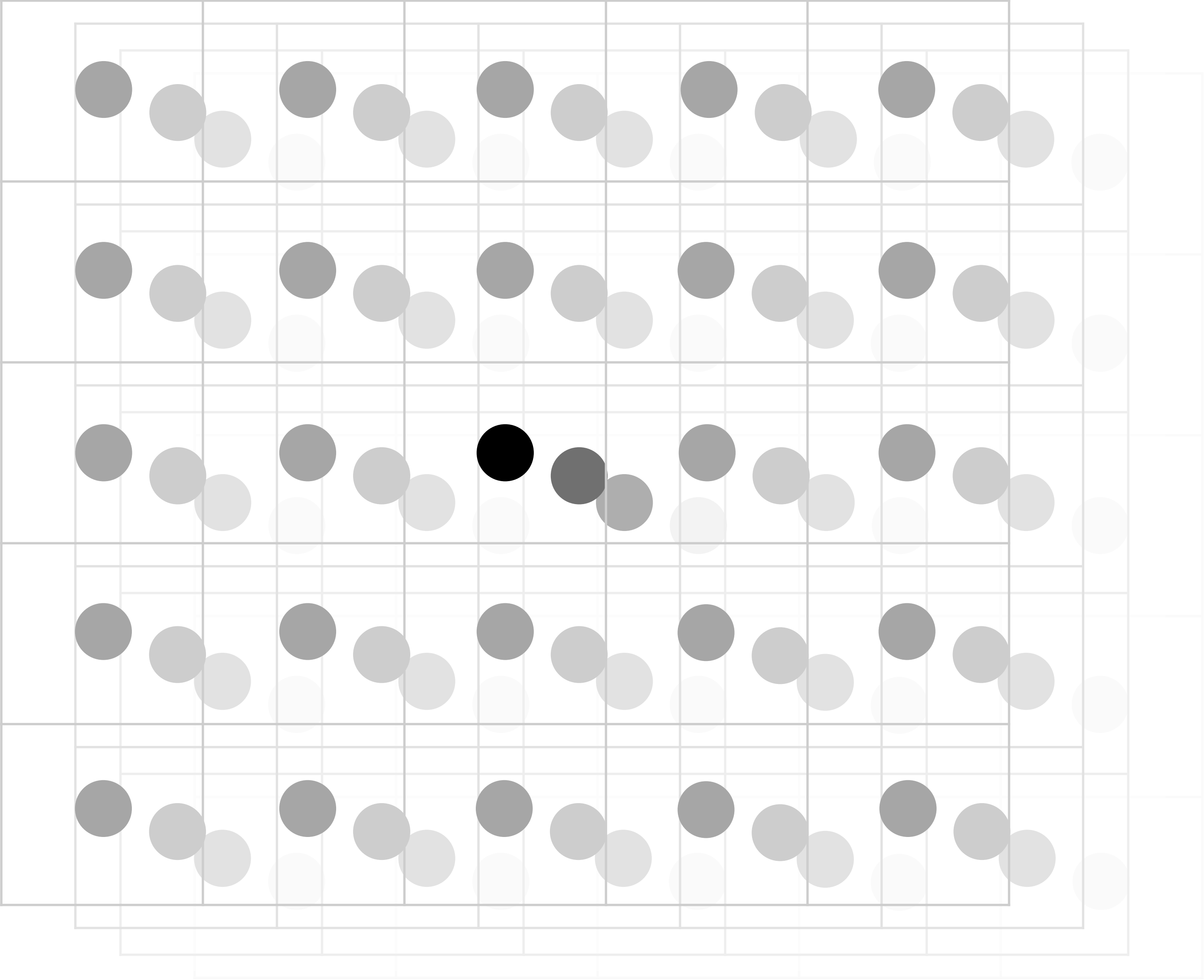
- We select 1 light from (2 x 800 =) 1600 effective samples

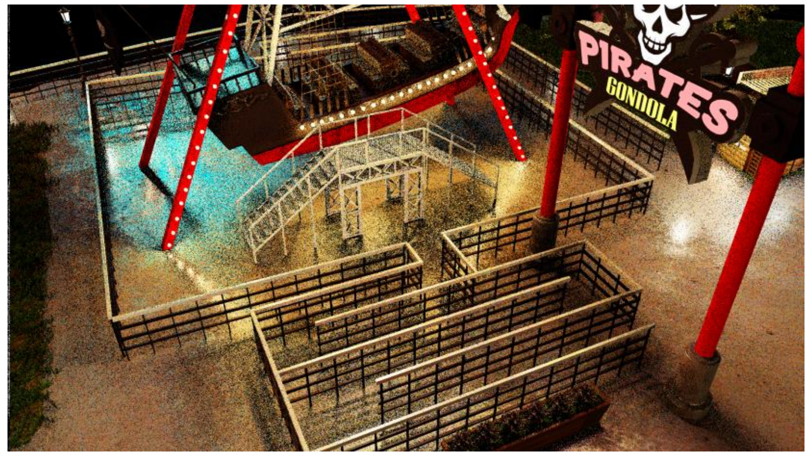
Do temporal reuse ***prior*** to spatial reuse

- Each spatial sample has 32 + 800 effective samples
- So we select 1 light from 25 x (32 + 800) ≈ 20,000 effective samples

WHY IS THIS USEFUL?

Allows the spatiotemporal reuse we desire





At each pixel, say we select 1 light from 32 random samples

- Using RIS and weighted reservoir sampling

At a cost of merging these 25 reservoirs

- We select 1 light from (25 x 32 =) 800 effective samples

Merge with our reservoir from last frame

- We select 1 light from (2 x 800 =) 1600 effective samples

Do temporal reuse ***prior*** to spatial reuse

- Each spatial sample has 32 + 800 effective samples
- So we select 1 light from 25 x (32 + 800) ≈ 20,000 effective samples

Reuse from a frame that reuses it’s prior frame…

IMPORTANT TO UNDERSTAND

Each reservoir merge is a *new application* of RIS

IMPORTANT TO UNDERSTAND

Each reservoir merge is a *new application* of RIS

Design space is huge:

What is your target function $\hat{p}(x)$ at each step?

IMPORTANT TO UNDERSTAND

Each reservoir merge is a *new application* of RIS

Design space is huge:

- What is your target function $\hat{p}(x)$ at each step?
- $\hat{p}(x)$ need not be physically correct; may include SSAO term to approx visibility

IMPORTANT TO UNDERSTAND

Each reservoir merge is a *new application* of RIS

Design space is huge:

- What is your target function $\hat{p}(x)$ at each step?
- $\hat{p}(x)$ need not be physically correct; may include SSAO term to approx visibility
- What coordinate space are samples stored in?

IMPORTANT TO UNDERSTAND

Each reservoir merge is a *new application* of RIS

Design space is huge:

- What is your target function $\hat{p}(x)$ at each step?
- $\hat{p}(x)$ need not be physically correct; may include SSAO term to approx visibility
- What coordinate space are samples stored in?
- What sampling PDF should you use for initial samples $q(x)$?

IMPORTANT TO UNDERSTAND

Each reservoir merge is a *new application* of RIS

Design space is huge:

- What is your target function $\hat{p}(x)$ at each step?
- $\hat{p}(x)$ need not be physically correct; may include SSAO term to approx visibility
- What coordinate space are samples stored in?
- What sampling PDF should you use for initial samples $q(x)$?
- Can you upsample and downsample reservoirs to change spatial compute frequency?

IMPORTANT TO UNDERSTAND

Each reservoir merge is a *new application* of RIS

Design space is huge:

- What is your target function $\hat{p}(x)$ at each step?
- $\hat{p}(x)$ need not be physically correct; may include SSAO term to approx visibility
- What coordinate space are samples stored in?
- What sampling PDF should you use for initial samples $q(x)$?
- Can you upsample and downsample reservoirs to change spatial compute frequency?
- Which neighbors to reuse (and which **not** to reuse)?

IMPORTANT TO UNDERSTAND

Each reservoir merge is a *new application* of RIS

Design space is huge:

- What is your target function $\hat{p}(x)$ at each step?
- $\hat{p}(x)$ need not be physically correct; may include SSAO term to approx visibility
- What coordinate space are samples stored in?
- What sampling PDF should you use for initial samples $q(x)$?
- Can you upsample and downsample reservoirs to change spatial compute frequency?
- Which neighbors to reuse (and which **not** to reuse)?
- How to scale to longer path lengths?

IMPORTANT TO UNDERSTAND

Each reservoir merge is a *new application* of RIS

Design space is huge:

- What is your target function $\hat{p}(x)$ at each step?
- $\hat{p}(x)$ need not be physically correct; may include SSAO term to approx visibility
- What coordinate space are samples stored in?
- What sampling PDF should you use for initial samples $q(x)$?
- Can you upsample and downsample reservoirs to change spatial compute frequency?
- Which neighbors to reuse (and which **not** to reuse)?
- How to scale to longer path lengths?
- PDFs from neighbor pixels can be quite different; how does this affect sampling quality?

EXPLORING THE DESIGN SPACE: VISIBILITY

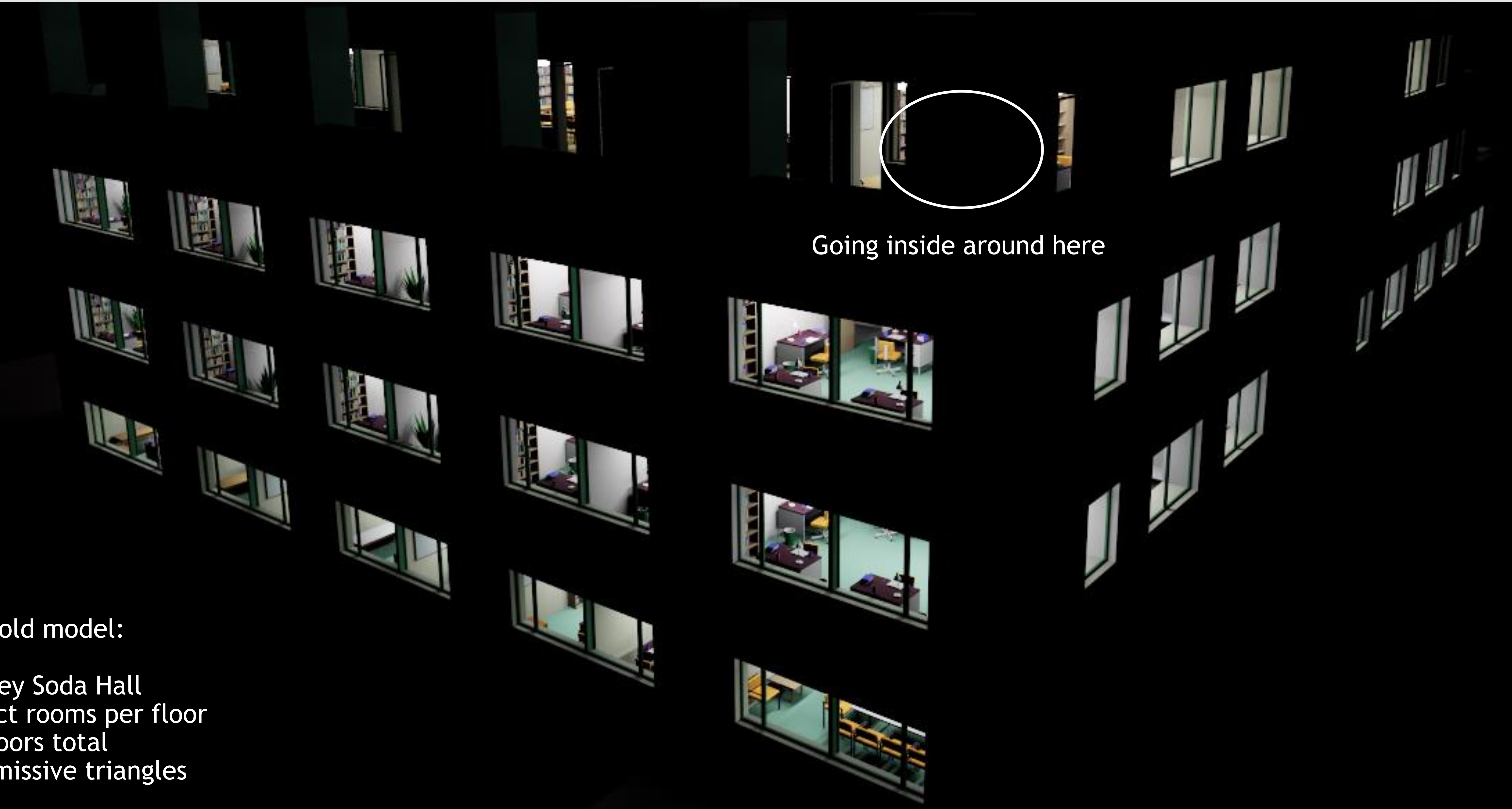
Visibility one of the most important, most expensive, most ignored aspects of rendering

Very old model:

Berkeley Soda Hall
20-40 distinct rooms per floor
6 floors total
50,000+ emissive triangles

EXPLORING THE DESIGN SPACE: VISIBILITY

Visibility one of the most important, most expensive, most ignored aspects of rendering



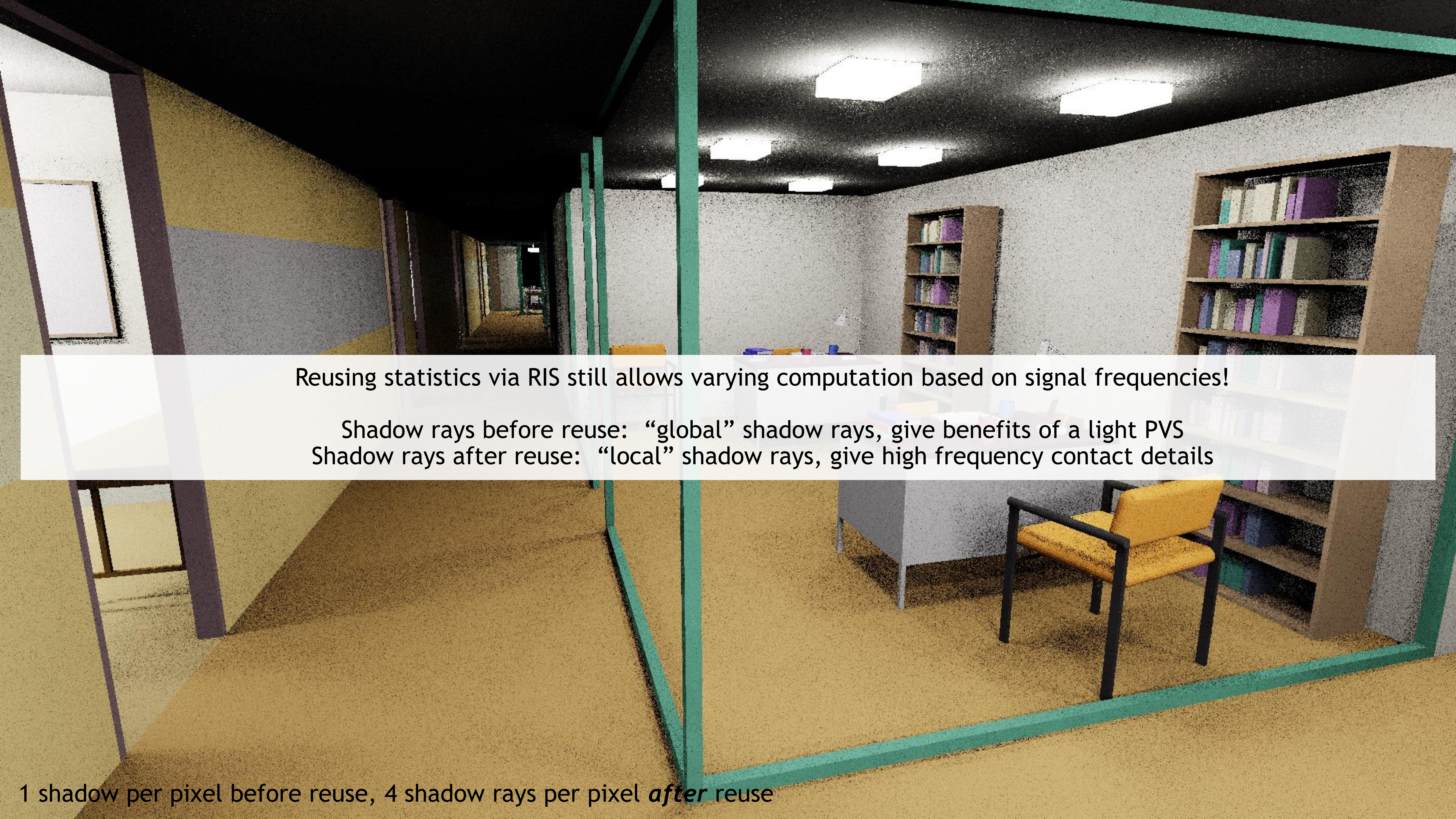
Very old model:

Berkeley Soda Hall
20-40 distinct rooms per floor
6 floors total
50,000+ emissive triangles

Considering all lights in scene, without visibility

Add 1 shadow ray per pixel ***before*** spatiotemporal reuse

1 shadow per pixel before reuse, 4 shadow rays per pixel ***after*** reuse

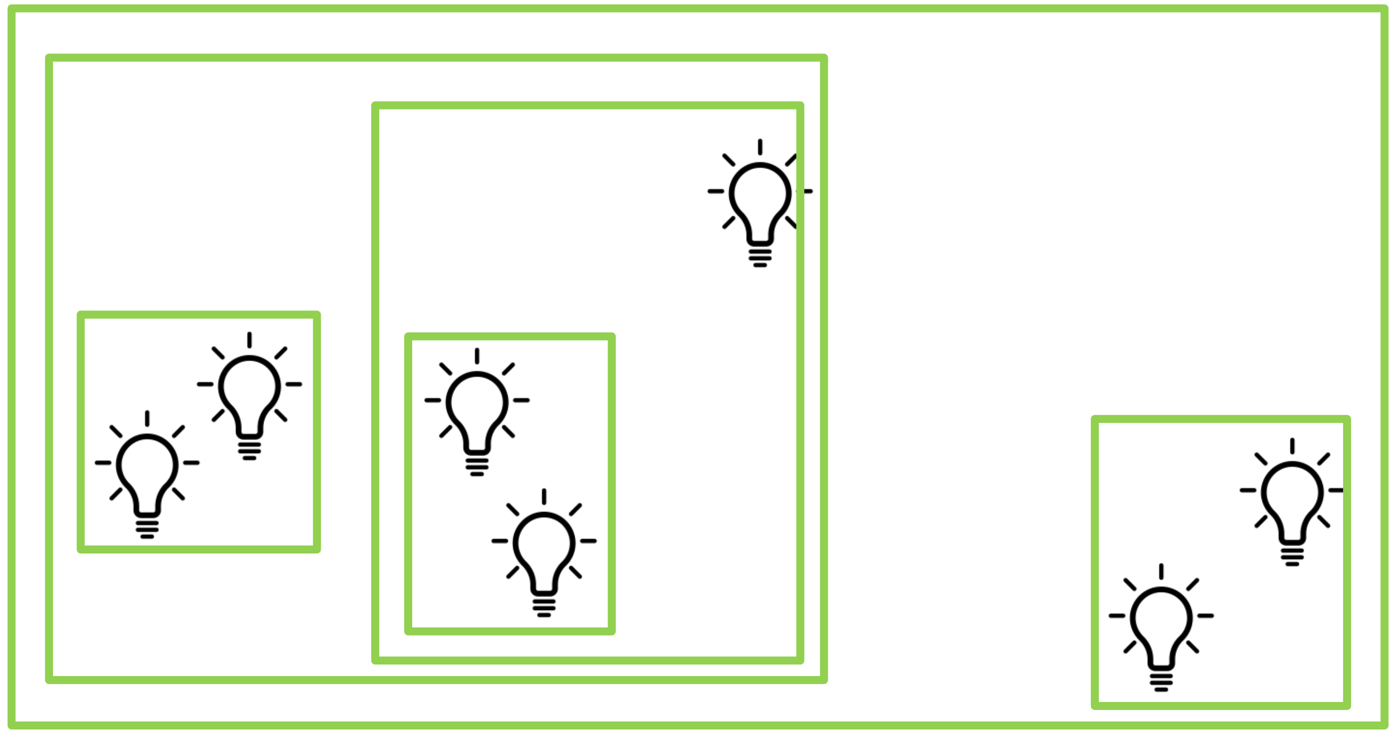

Reusing statistics via RIS still allows varying computation based on signal frequencies!
Shadow rays before reuse: “global” shadow rays, give benefits of a light PVS
Shadow rays after reuse: “local” shadow rays, give high frequency contact details
1 shadow per pixel before reuse, 4 shadow rays per pixel *after* reuse

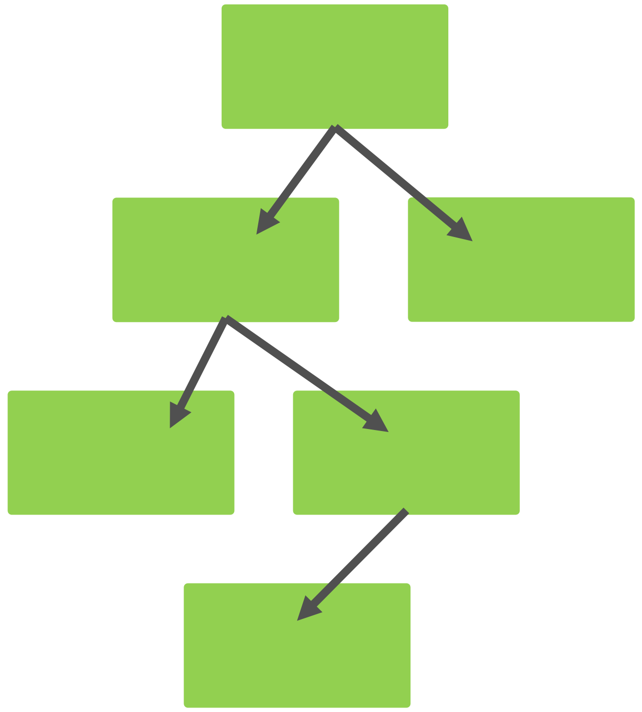
ANOTHER WAY TO ABSTRACT THE PROBLEM

We need to accelerate certain sampling processes; how?

ANOTHER WAY TO ABSTRACT THE PROBLEM

We need to accelerate certain sampling processes; how?



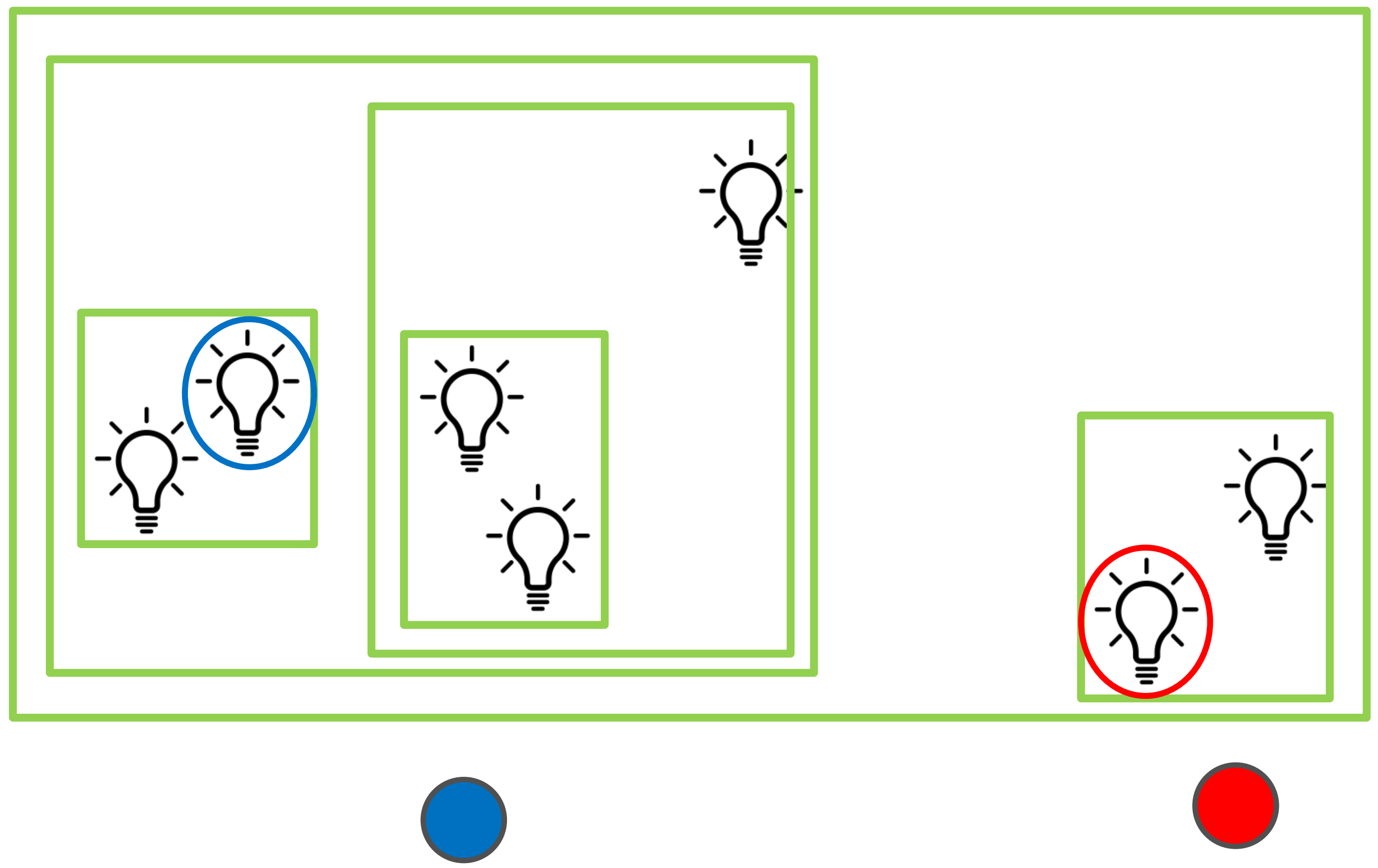


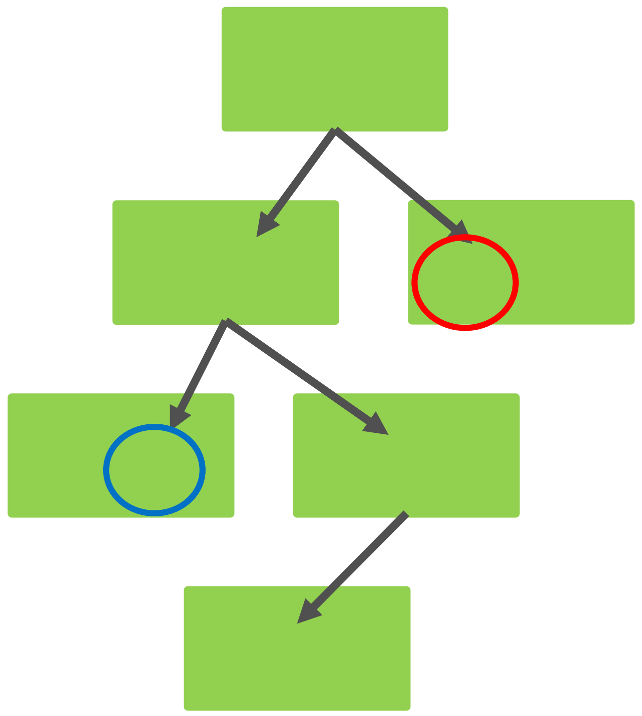
Frequently some tree

Deterministic build / update

ANOTHER WAY TO ABSTRACT THE PROBLEM

We need to accelerate certain sampling processes; how?





Frequently some tree

Deterministic build / update

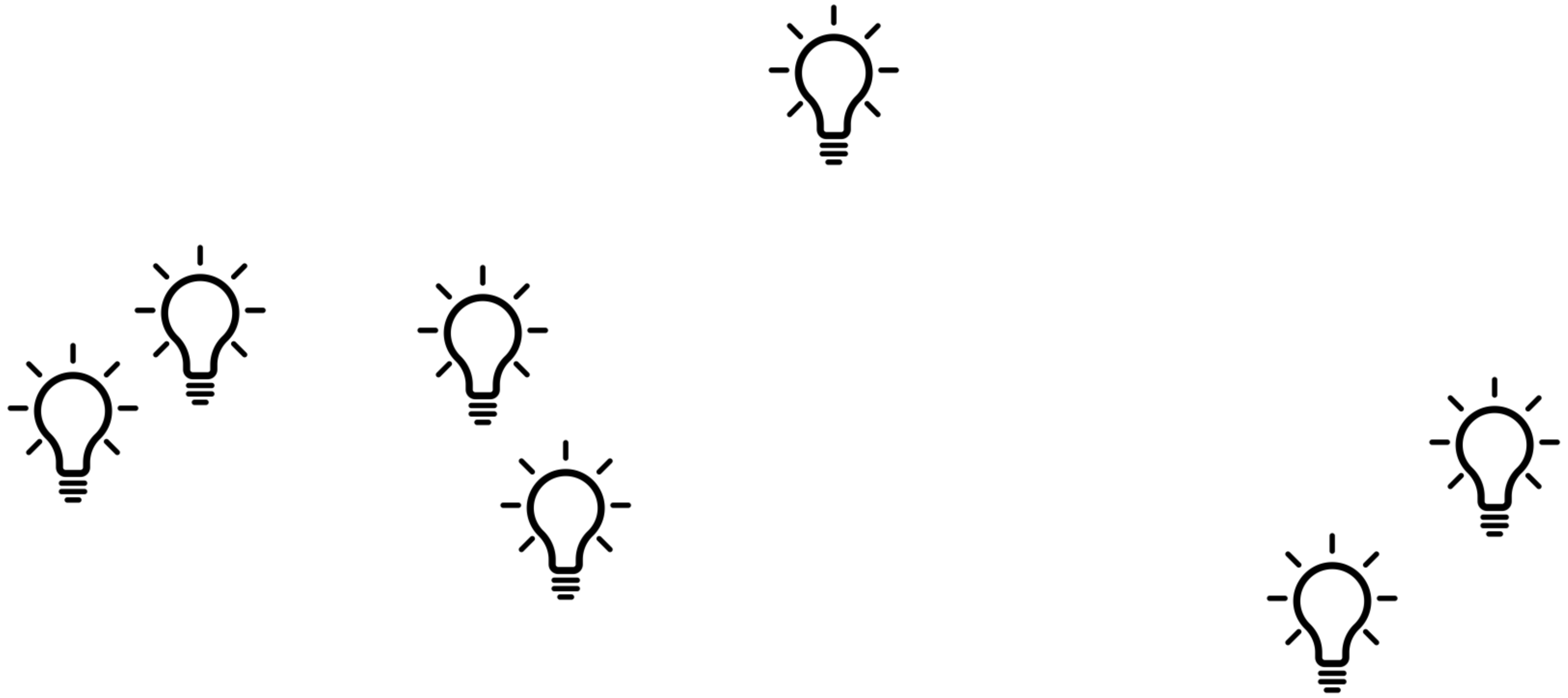
Traversal is randomized

Blue pixel more likely to pick from left, red pixel more from the right

Traverse tree randomly, proportional to (likely) contribution from each child node

ANOTHER WAY TO ABSTRACT THE PROBLEM

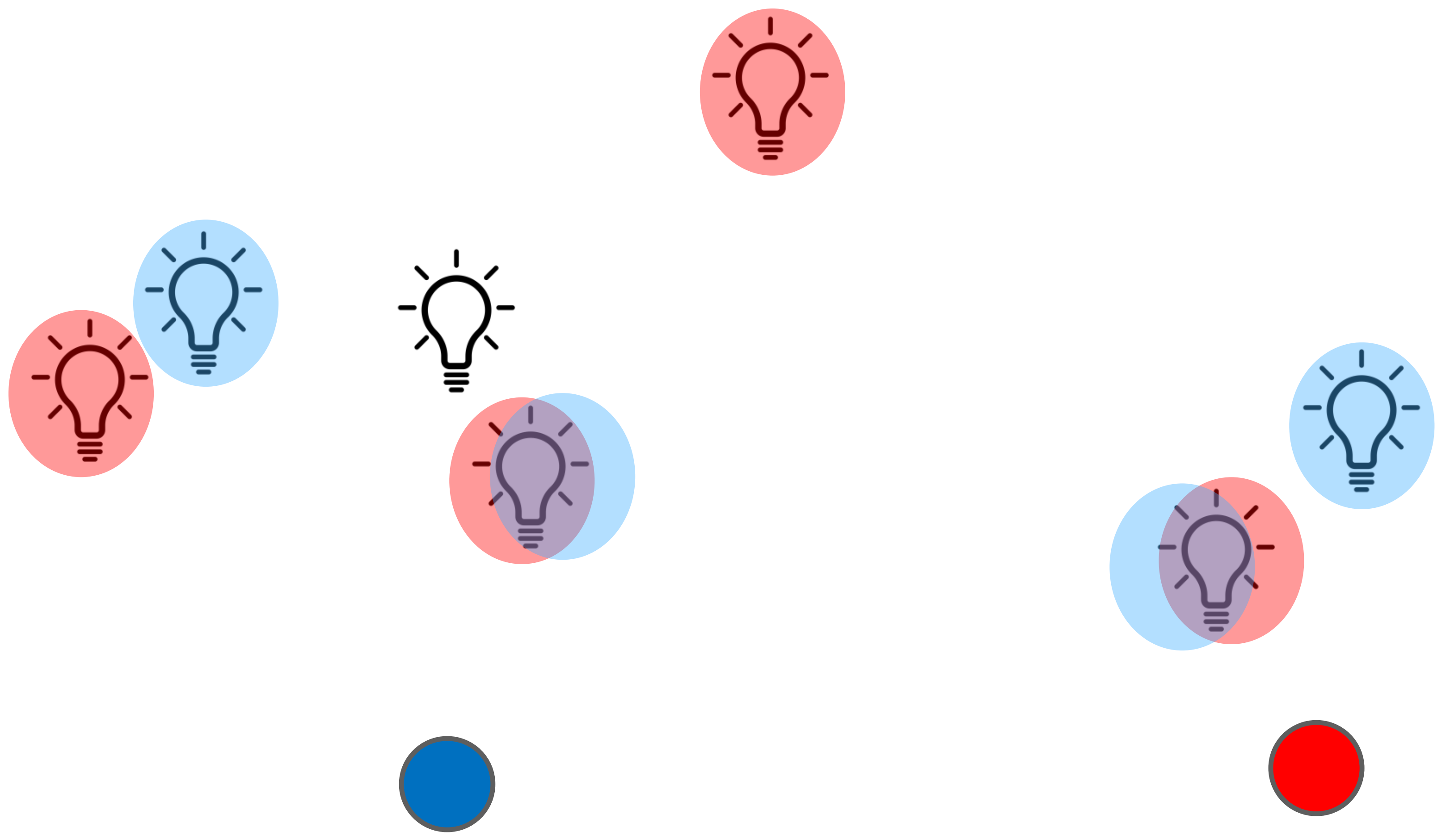
Importance resampling with reuse relies a different (statistical) approach



No structure built in advance

ANOTHER WAY TO ABSTRACT THE PROBLEM

Importance resampling with reuse relies a different (statistical) approach



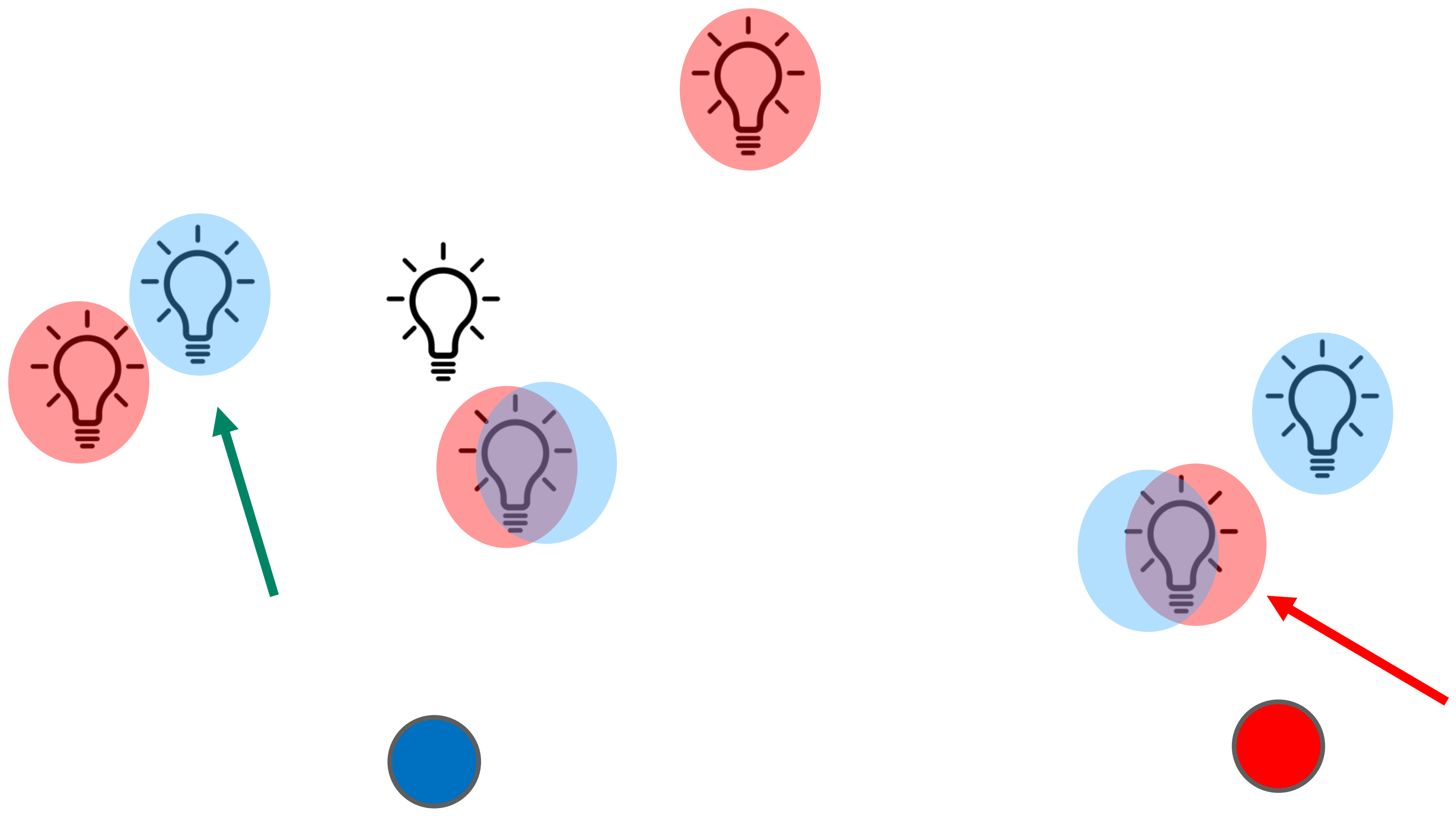
No structure built in advance

Each pixel starts by

- Randomly selecting "candidates"
- Independently; may overlap

ANOTHER WAY TO ABSTRACT THE PROBLEM

Importance resampling with reuse relies a different (statistical) approach



No structure built in advance

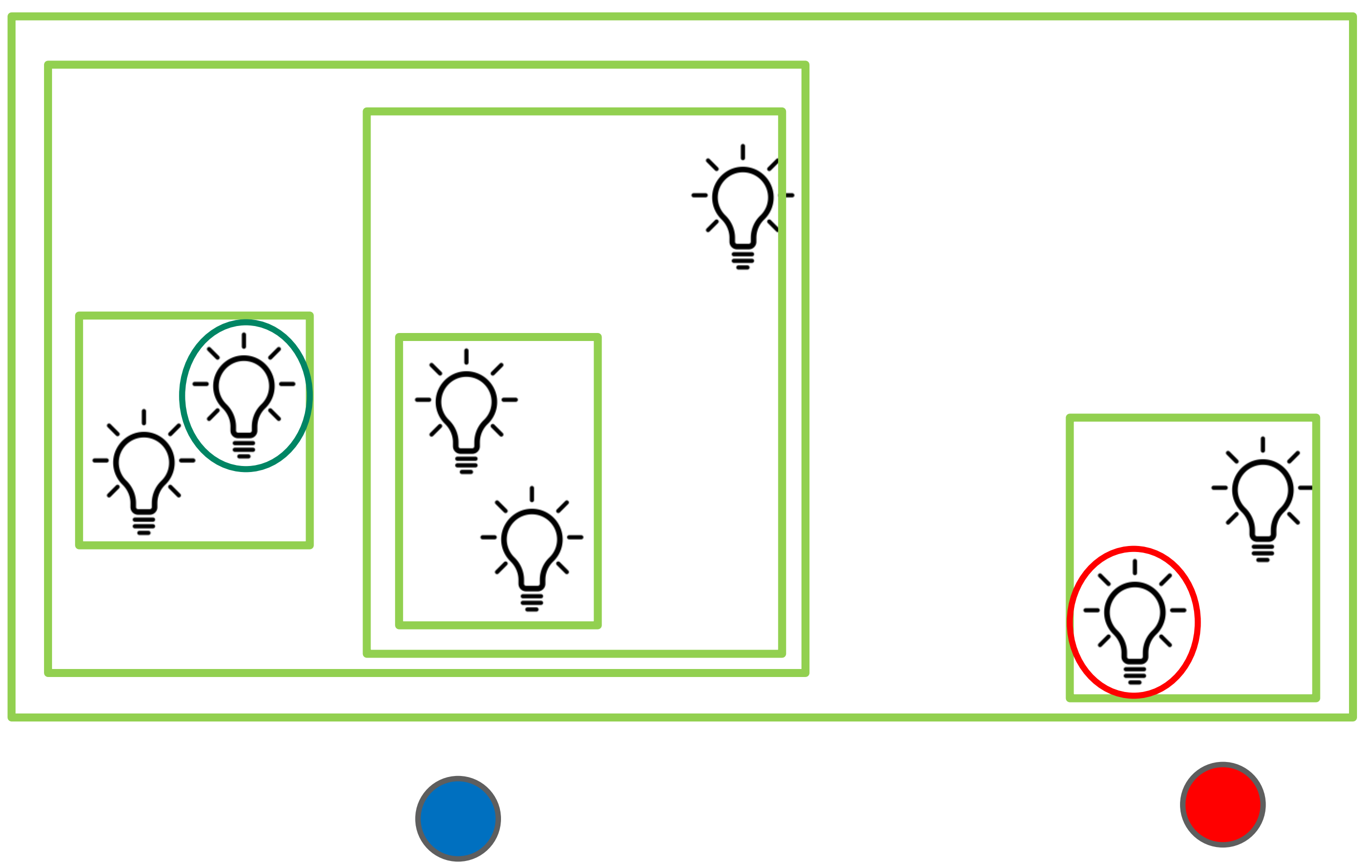
Each pixel starts by

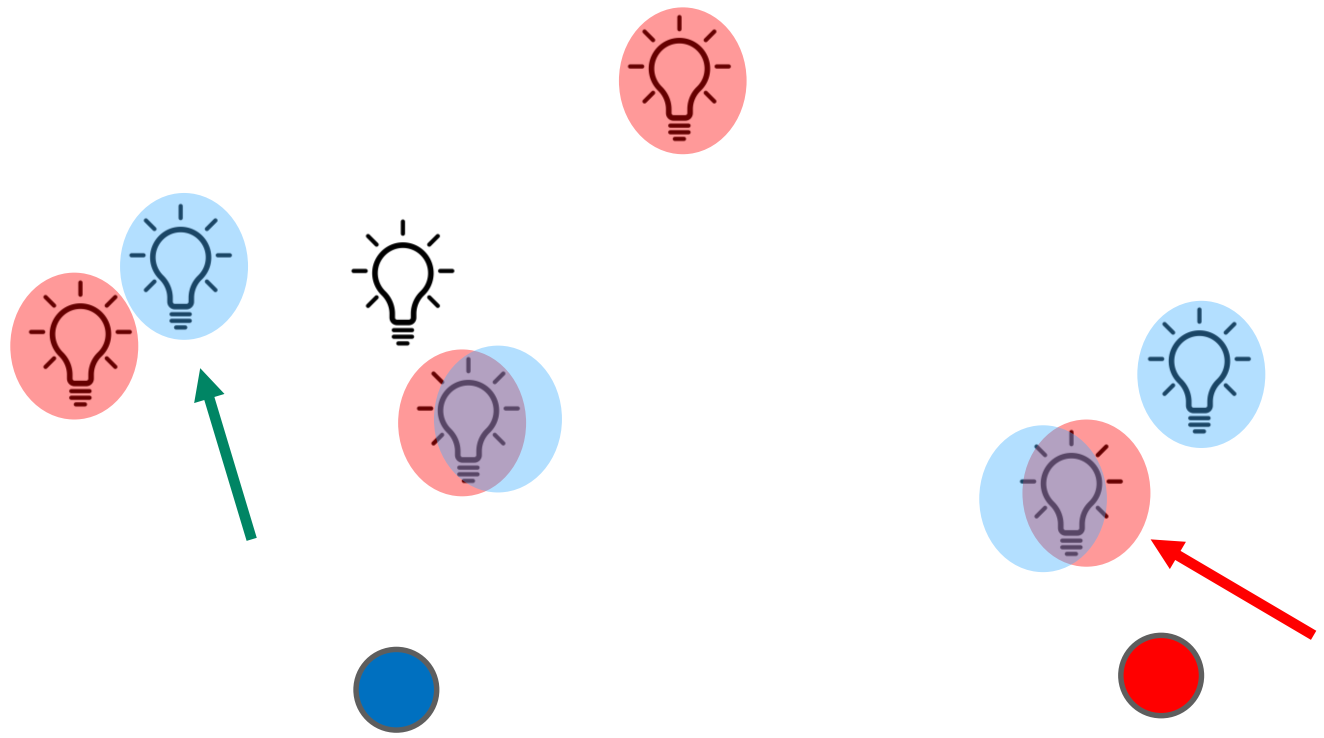
- Randomly selecting “candidates”
- Independently; may overlap

Evaluate likelihood of each candidate

- Select one
- Proportional to its likelihood

COMPARISON





Rebuild structure for dynamic lights, O(n log n)

Random memory reads (stochastic traversal)

Dependent memory reads (parent → child nodes)

Variable cost / divergence (non balanced tree)

No upfront costs

Random memory reads (candidate selection)

Independent memory (do candidates in parallel)

Fixed cost (set candidate count; may be larger)

INTERESTING TAKEAWAYS

Both relying on randomization

INTERESTING TAKEAWAYS

Both relying on randomization

Both an example of a randomized algorithm

INTERESTING TAKEAWAYS

Both relying on randomization

Both an example of a randomized algorithm

- This is hole in my knowledge of complexity theory
- Different classes of polynomial-time algorithms, with and w/o randomness
- Unclear if classes are identical

INTERESTING TAKEAWAYS

Both relying on randomization

Both an example of a randomized algorithm

- This is hole in my knowledge of complexity theory
- Different classes of polynomial-time algorithms, with and w/o randomness
- Unclear if classes are identical

Seems important; randomized algorithms you use:

- Quicksort, Monte Carlo integration, most light sampling acceleration structures, neural nets

INTERESTING TAKEAWAYS

Lots of statistics theory to mine

INTERESTING TAKEAWAYS

Lots of statistics theory to mine

Resampled importance sampling

- A form of “sampling importance resampling”

INTERESTING TAKEAWAYS

Lots of statistics theory to mine

Resampled importance sampling

- A form of “sampling importance resampling”
- A form of rejection sampling, bootstrap filters, particle filtering
- With reuse, seems related to sequential Monte Carlo and population Monte Carlo methods

INTERESTING TAKEAWAYS

Lots of statistics theory to mine

Resampled importance sampling

- A form of "sampling importance resampling"
- A form of rejection sampling, bootstrap filters, particle filtering
- With reuse, seems related to sequential Monte Carlo and population Monte Carlo methods

Also, interesting properties of weighted reservoir sampling

- Independent v.s. dependent sampling
- Alternate form using randomized exponentiated weights

INTERESTING TAKEAWAYS

Data structure builds take lots of time

INTERESTING TAKEAWAYS

Data structure builds take lots of time

Need to build it (costly)

INTERESTING TAKEAWAYS

Data structure builds take lots of time

Need to build it	(costly)
Need to maintain it	(costly)

INTERESTING TAKEAWAYS

Data structure builds take lots of time

Need to build it	(costly)
Need to maintain it	(costly)
Even parts unused this frame	(wasteful)

INTERESTING TAKEAWAYS

Data structure builds take lots of time

Need to build it	(costly)
Need to maintain it	(costly)
Even parts unused this frame	(wasteful)
Dependent reads during traversal	(costly)

INTERESTING TAKEAWAYS

Data structure builds take lots of time

Need to build it	(costly)
Need to maintain it	(costly)
Even parts unused this frame	(wasteful)
Dependent reads during traversal	(costly)
Traversals of varying cost	(incoherent)

INTERESTING TAKEAWAYS

Data structure builds take lots of time

Need to build it	(costly)
Need to maintain it	(costly)
Even parts unused this frame	(wasteful)
Dependent reads during traversal	(costly)
Traversals of varying cost	(incoherent)
Sometimes traversal is stochastic	(incoherent)

INTERESTING TAKEAWAYS

Data structure builds take lots of time

Need to build it	(costly)
Need to maintain it	(costly)
Even parts unused this frame	(wasteful)
Dependent reads during traversal	(costly)
Traversals of varying cost	(incoherent)
Sometimes traversal is stochastic	(incoherent)

Why not use a “randomized” data structure?

INTERESTING TAKEAWAYS

Data structure builds take lots of time

Need to build it	(costly)
Need to maintain it	(costly)
Even parts unused this frame	(wasteful)
Dependent reads during traversal	(costly)
Traversals of varying cost	(incoherent)
Sometimes traversal is stochastic	(incoherent)

Why not use a "randomized" data structure?

Not a deterministic build followed by stochastic traversal

Instead, a stochastic build with a deterministic traversal

INTERESTING TAKEAWAYS

Data structure builds take lots of time

Need to build it	(costly)
Need to maintain it	(costly)
Even parts unused this frame	(wasteful)
Dependent reads during traversal	(costly)
Traversals of varying cost	(incoherent)
Sometimes traversal is stochastic	(incoherent)

Why not use a "randomized" data structure?

Not a deterministic build followed by stochastic traversal

Instead, a stochastic build with a deterministic traversal

… more an ideology than a concrete idea

But iterative RIS starts to resemble this

TAKES THIS PHILOSOPHY TO THE EXTREME

(From "Real-time Stochastic Light Cuts," Lin & Yuksel, I3D 2020)

Our goal is to minimize the light tree construction and light sample selection times. The choices we make for achieving this goal, however, may adversely affect the quality of the tree and the light sample distribution. Yet, this reduction in sampling quality can be offset by using more light samples. Thus, our ultimate goal is

TAKES THIS PHILOSOPHY TO THE EXTREME

(From "Real-time Stochastic Light Cuts," Lin & Yuksel, I3D 2020)

Our goal is to minimize the light tree construction and light sample selection times. The choices we make for achieving this goal, however, may adversely affect the quality of the tree and the light sample distribution. Yet, this reduction in sampling quality can be offset by using more light samples. Thus, our ultimate goal is

TAKES THIS PHILOSOPHY TO THE EXTREME

(From “Real-time Stochastic Light Cuts,” Lin & Yuksel, I3D 2020)

Our goal is to minimize the light tree construction and light sample selection times. The choices we make for achieving this goal, however, may adversely affect the quality of the tree and the light sample distribution. Yet, this reduction in sampling quality can be offset by using more light samples. Thus, our ultimate goal is

TAKES THIS PHILOSOPHY TO THE EXTREME

(From "Real-time Stochastic Light Cuts," Lin & Yuksel, I3D 2020)

Our goal is to minimize the light tree construction and light sample selection times. The choices we make for achieving this goal, however, may adversely affect the quality of the tree and the light sample distribution. Yet, this reduction in sampling quality can be offset by using more light samples. Thus, our ultimate goal is

TAKES THIS PHILOSOPHY TO THE EXTREME

(From "Real-time Stochastic Light Cuts," Lin & Yuksel, I3D 2020)

Our goal is to minimize the light tree construction and light sample selection times. The choices we make for achieving this goal, however, may adversely affect the quality of the tree and the light sample distribution. Yet, this reduction in sampling quality can be offset by using more light samples. Thus, our ultimate goal is

Stochastic reuse: Builds tree so loosely it's not recognizable; samples are cheap

TAKES THIS PHILOSOPHY TO THE EXTREME

(From “Real-time Stochastic Light Cuts,” Lin & Yuksel, I3D 2020)

> Our goal is to minimize the light tree construction and light sample selection times. The choices we make for achieving this goal, however, may adversely affect the quality of the tree and the light sample distribution. Yet, this reduction in sampling quality can be offset by using more light samples. Thus, our ultimate goal is

Stochastic reuse: Builds tree so loosely it’s not recognizable; samples are cheap

Definitely impacts sample quality

TAKES THIS PHILOSOPHY TO THE EXTREME

(From “Real-time Stochastic Light Cuts,” Lin & Yuksel, I3D 2020)

> Our goal is to minimize the light tree construction and light sample selection times. The choices we make for achieving this goal, however, may adversely affect the quality of the tree and the light sample distribution. Yet, this reduction in sampling quality can be offset by using more light samples. Thus, our ultimate goal is

Stochastic reuse: Builds tree so loosely it’s not recognizable; samples are cheap

Definitely impacts sample quality

But, it (effectively) uses 100,000+ samples per pixel

SUMMARY

Real-time capture, 10,400 area lights + environment map

SUMMARY

Real-time ray tracing is here today

- But… traditional CPU algorithms may not suit GPU-based stream processing

SUMMARY

Real-time ray tracing is here today

- But… traditional CPU algorithms may not suit GPU-based stream processing

Need to rethink our approach to optimize performance

- A big chance with game ray tracing, with different constraints than film

SUMMARY

Real-time ray tracing is here today

- But… traditional CPU algorithms may not suit GPU-based stream processing

Need to rethink our approach to optimize performance

- A big chance with game ray tracing, with different constraints than film

This talk, three big theses:

- Remember real-time constraints
- Rethink complex data structures
- Consider streaming statistics and PDFs (rather than triangles and rays)

SUMMARY

Real-time ray tracing is here today

- But… traditional CPU algorithms may not suit GPU-based stream processing

Need to rethink our approach to optimize performance

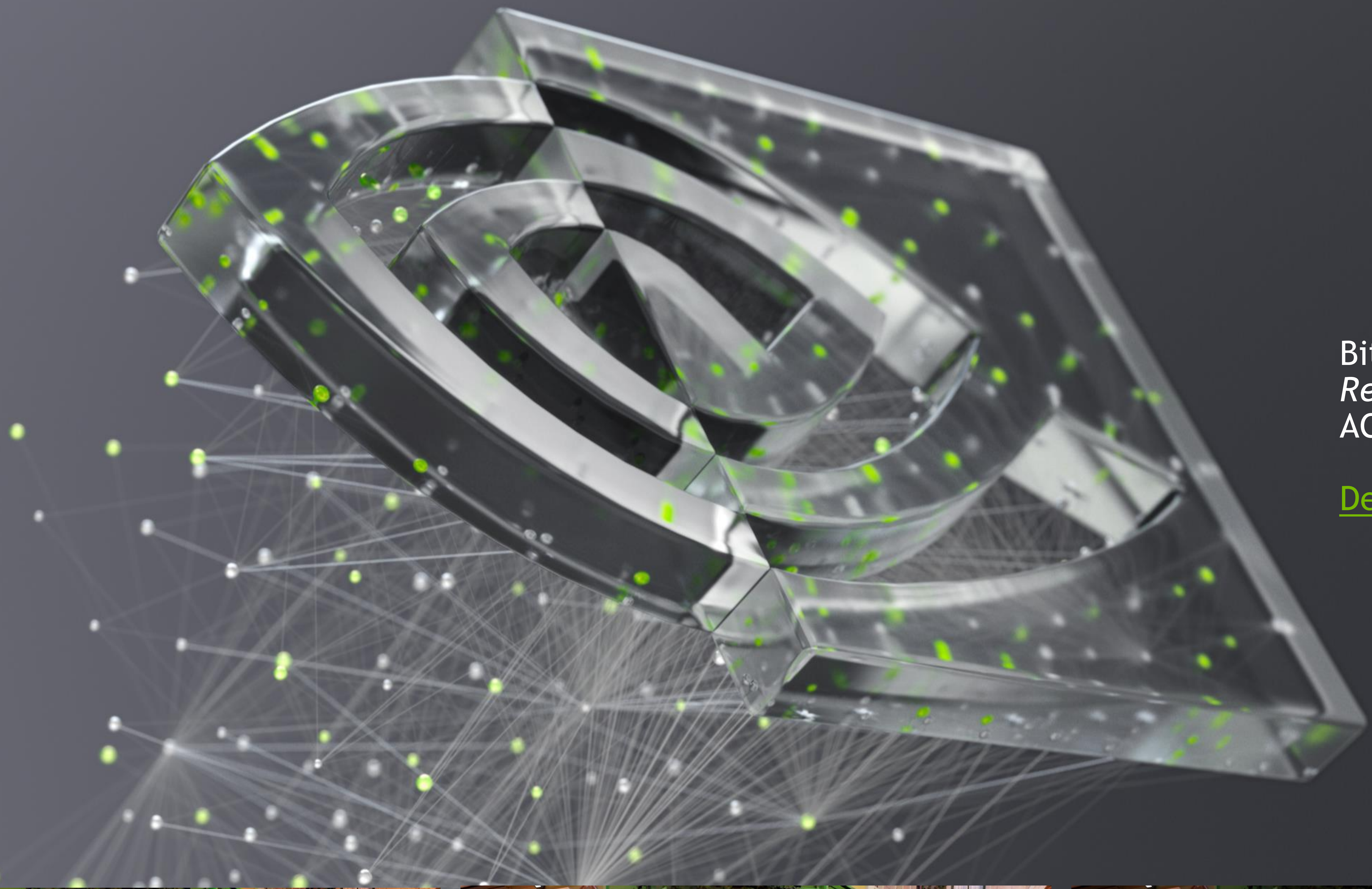
- A big chance with game ray tracing, with different constraints than film

This talk, three big theses:

- Remember real-time constraints
- Rethink complex data structures
- Consider streaming statistics and PDFs (rather than triangles and rays)

Likely other ways to reframe the problem

- … and people are beating down the doors to try them out



Questions?

Twitter: @_cwyman_
Email: cwyman@nvidia.com

Bitterli et al., "*Spatiotemporal Reservoir Resampling for Real-Time Ray Tracing with Dynamic Direct Lighting,*" ACM Transactions on Graphics 39(4), Article 148

Demo Video (Google for "NVIDIA ReSTIR YouTube")