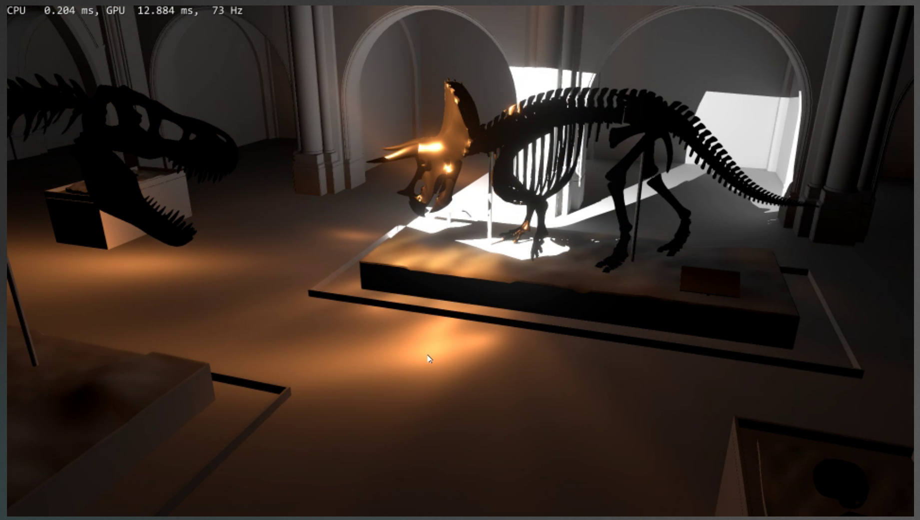
Bidirectional Path Tracing Using Backward Stochastic Light Culling

YUSUKE TOKUYOSHI (SQUARE ENIX CO., LTD.)
TAKAHIRO HARADA (ADVANCED MICRO DEVICES, INC.)



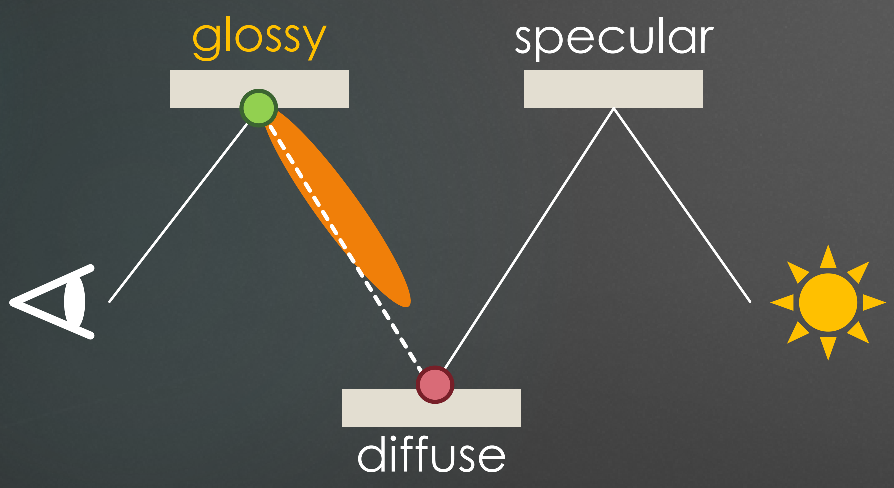
Stochastic light culling for glossy caustics
[Tokuyoshi17]

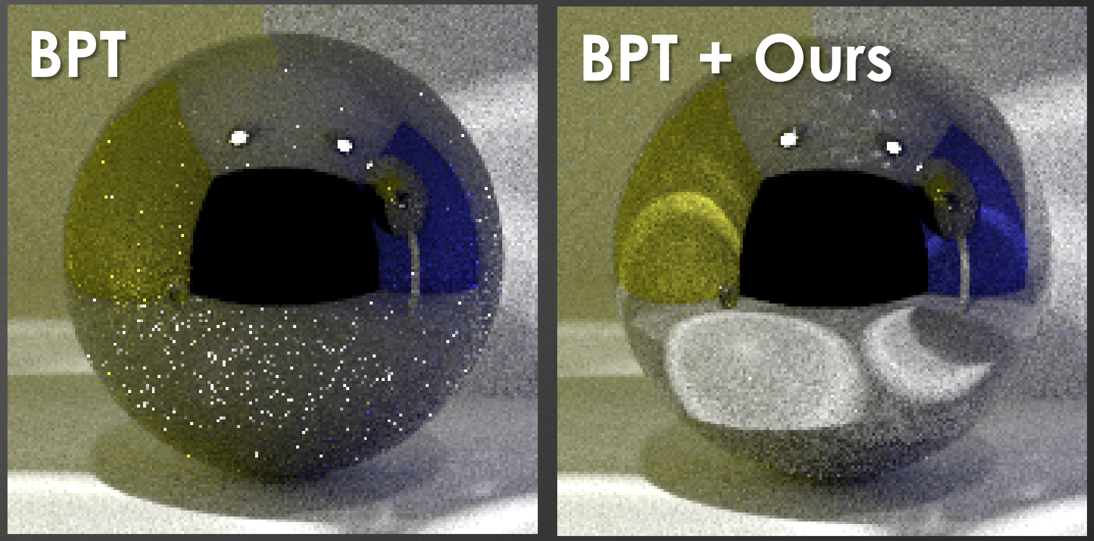
Real-time rendering

Offline bidirectional path tracing
(BPT)

Specular-Diffuse-Glossy Paths in BPT

- Connectable
- Inefficient for very highly glossy reflections ☹

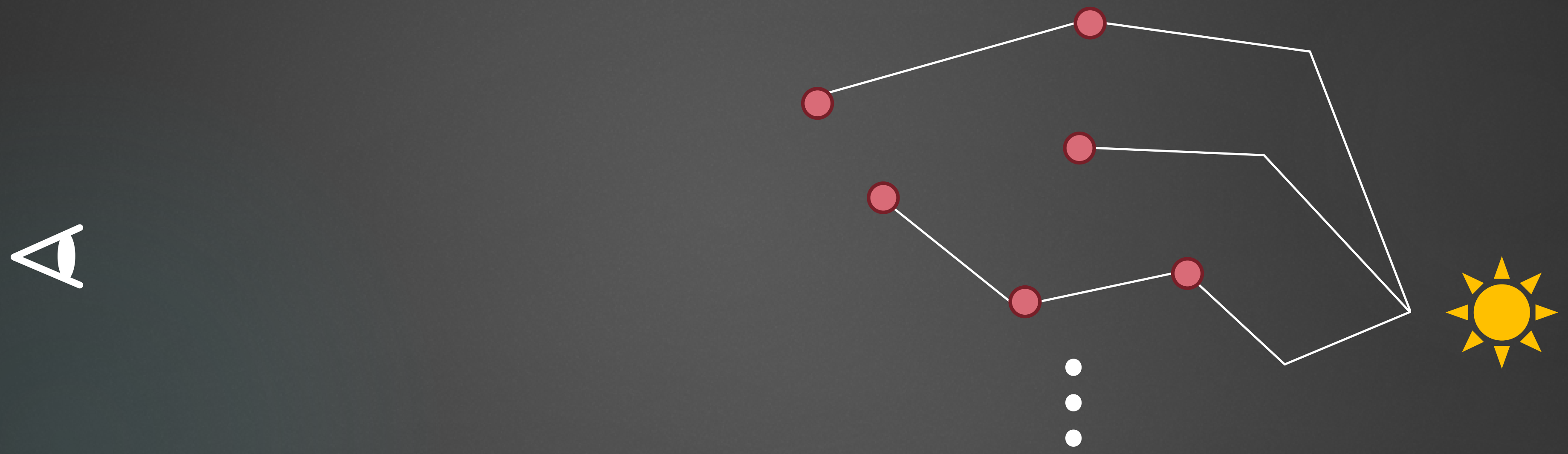




GGX roughness: 0.001

Algorithm Overview

Light-subpath tracing pass

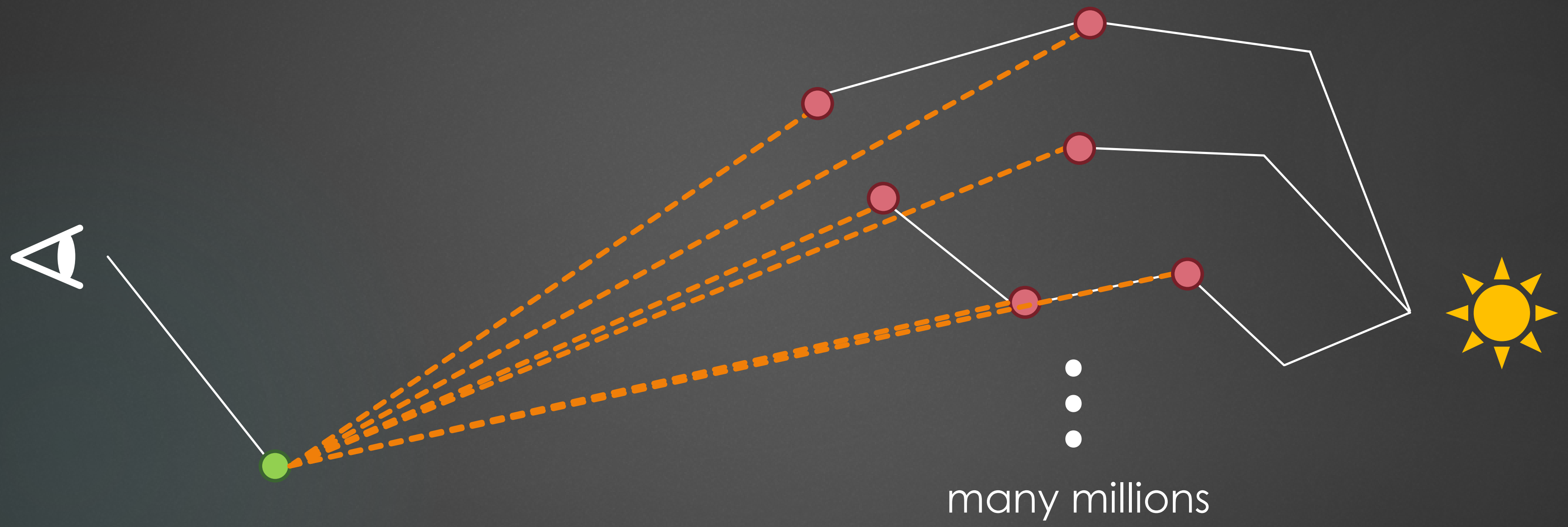


many millions

Store many light vertices in a cache
similar to virtual point lights [Keller97]

Algorithm Overview

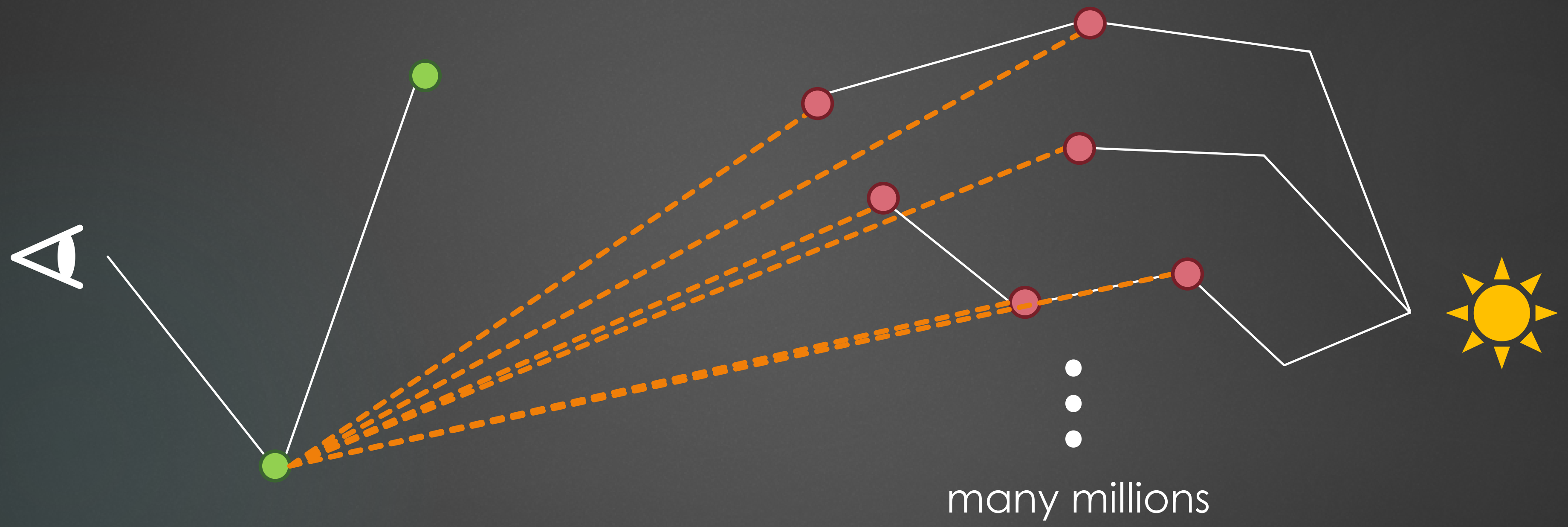
Eye-subpath tracing pass



Russian roulette [Arvo90] for all the cached light vertices

Algorithm Overview

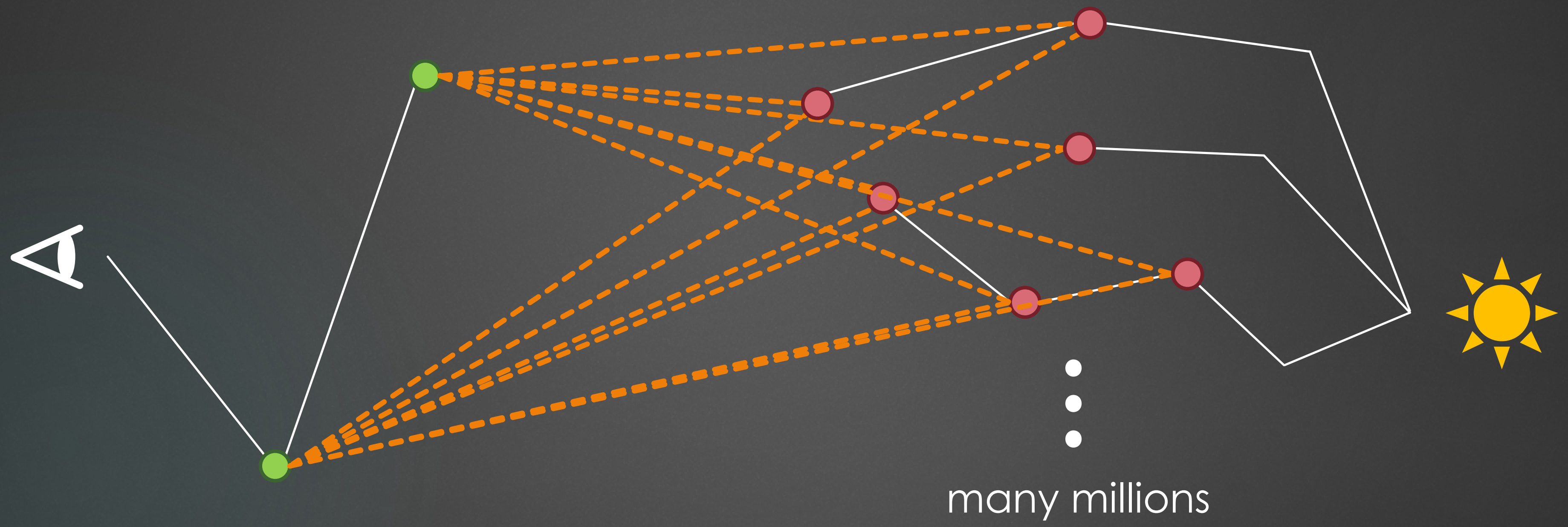
Eye-subpath tracing pass



Russian roulette [Arvo90] for all the cached light vertices

Algorithm Overview

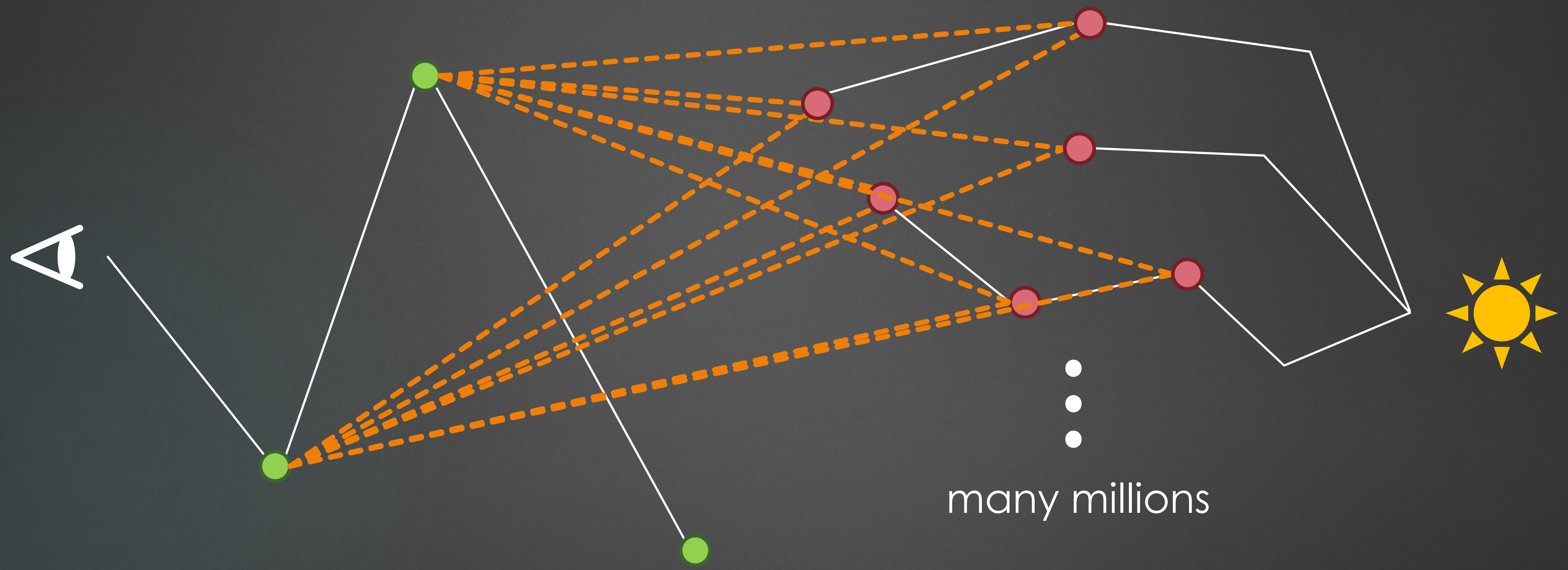
Eye-subpath tracing pass



Russian roulette [Arvo90] for all the cached light vertices

Algorithm Overview

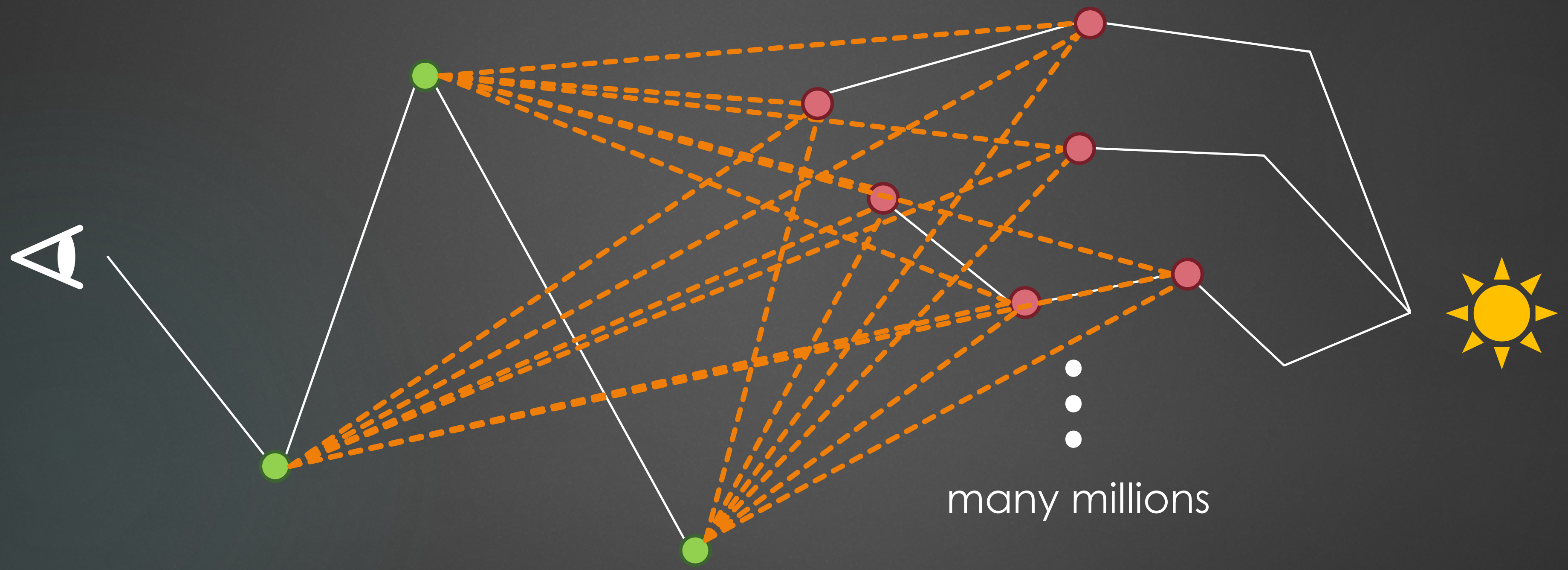
Eye-subpath tracing pass



Russian roulette [Arvo90] for all the cached light vertices

Algorithm Overview

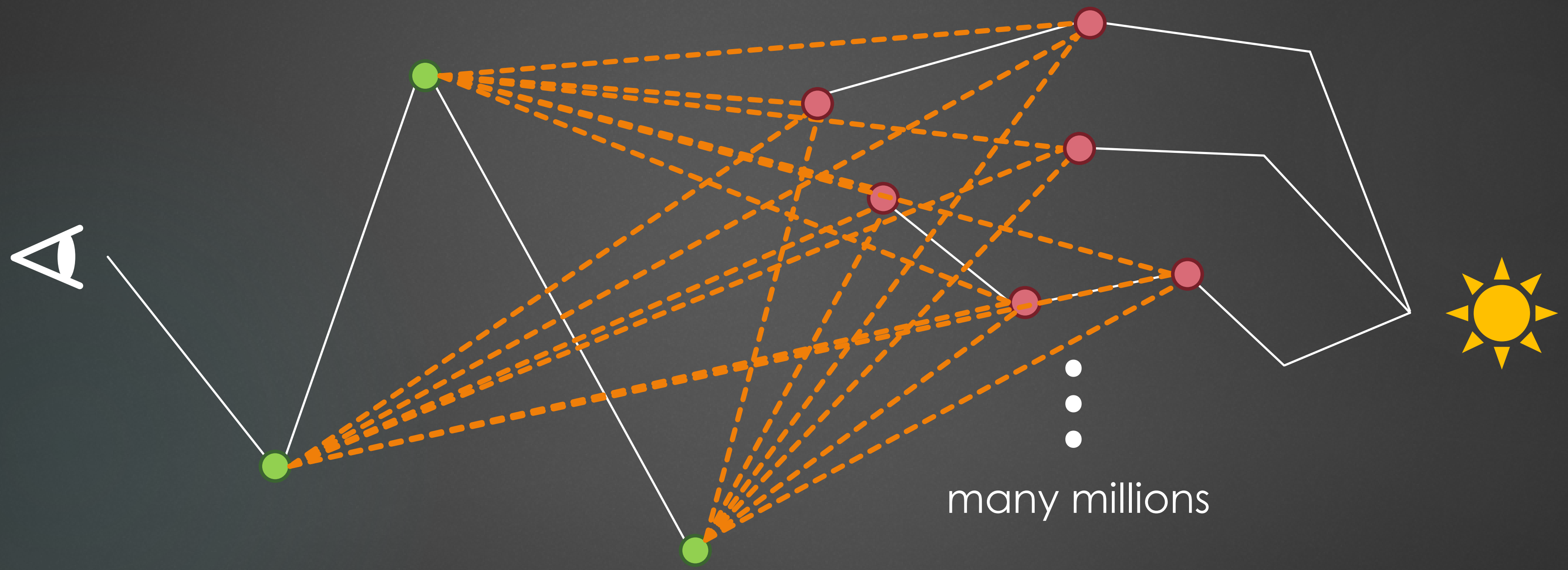
Eye-subpath tracing pass



Russian roulette [Arvo90] for all the cached light vertices

Algorithm Overview

Eye-subpath tracing pass

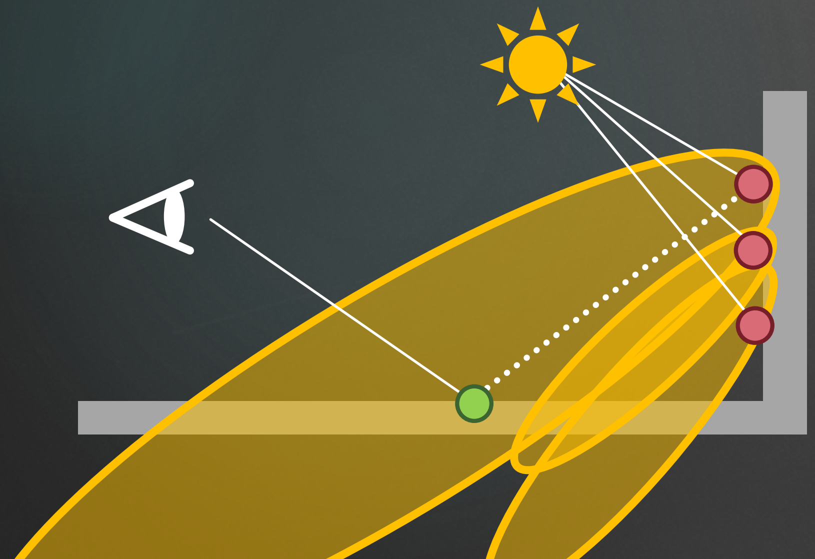


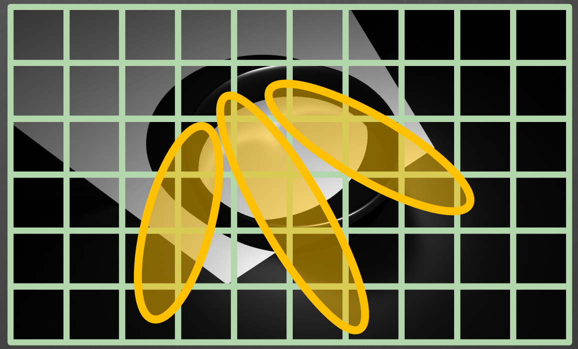
Russian roulette [Arvo90] for all the cached light vertices

Accelerated by stochastic light culling

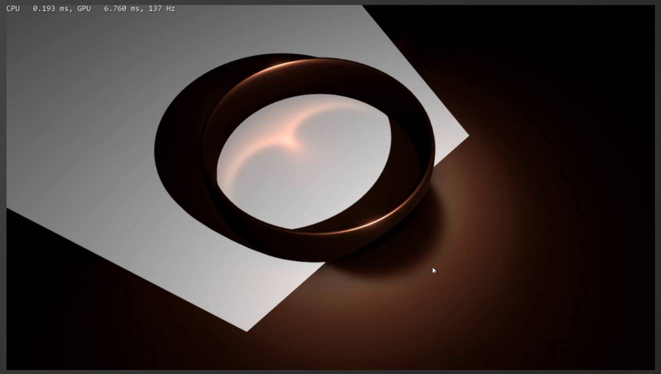
Stochastic Light Culling [Tokuyoshi16]

- Restrict the range of influence of a light vertex based on Russian roulette
- Ellipsoidal range for glossy reflections
  - Computable analytically for Phong and GGX BRDFs [Dachsbacher06, Tokuyoshi17]
- Real-time culling algorithms are available (e.g., tiled culling)





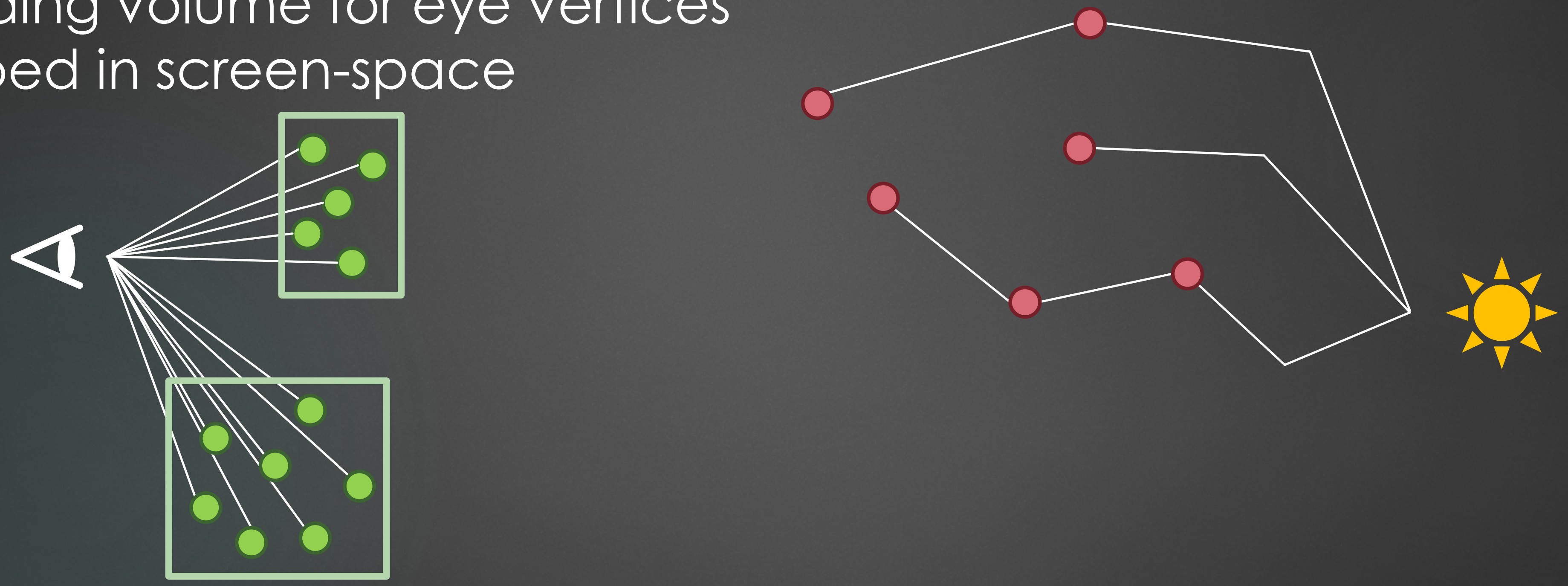
tiled culling for ellipsoids



65536 light vertices

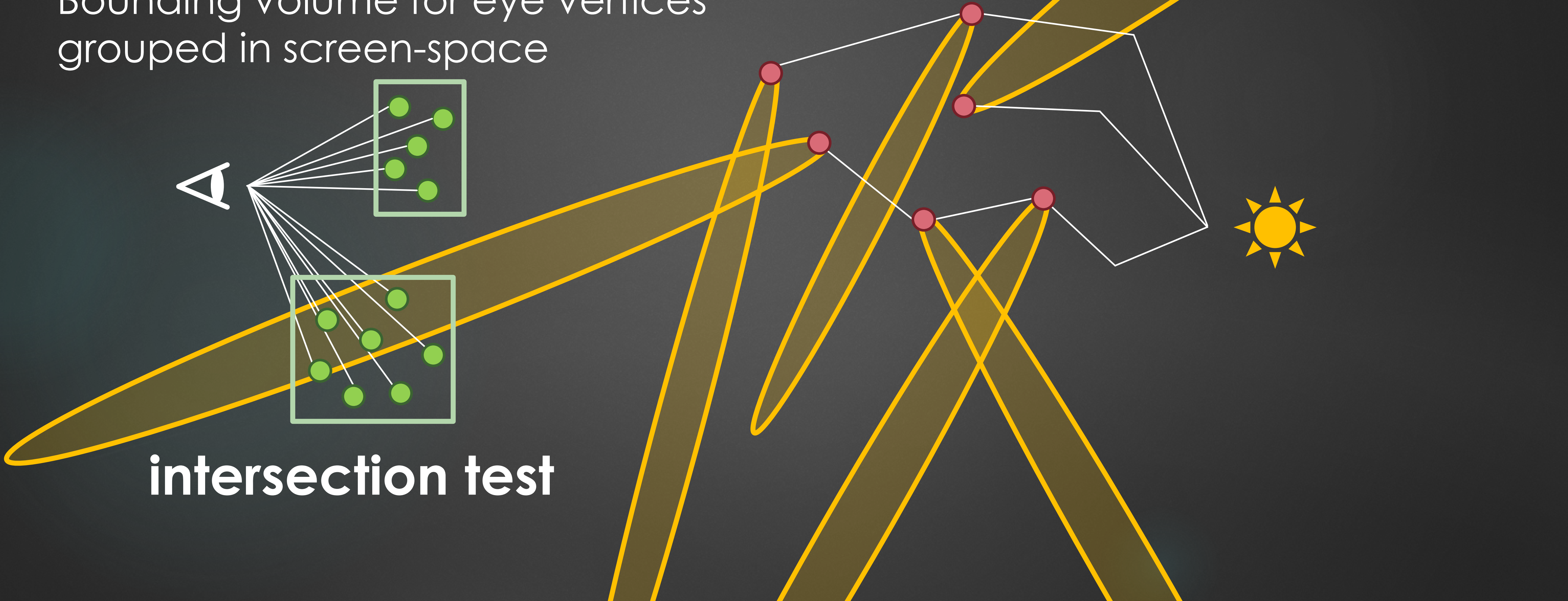
Tiled Culling

Bounding volume for eye vertices grouped in screen-space



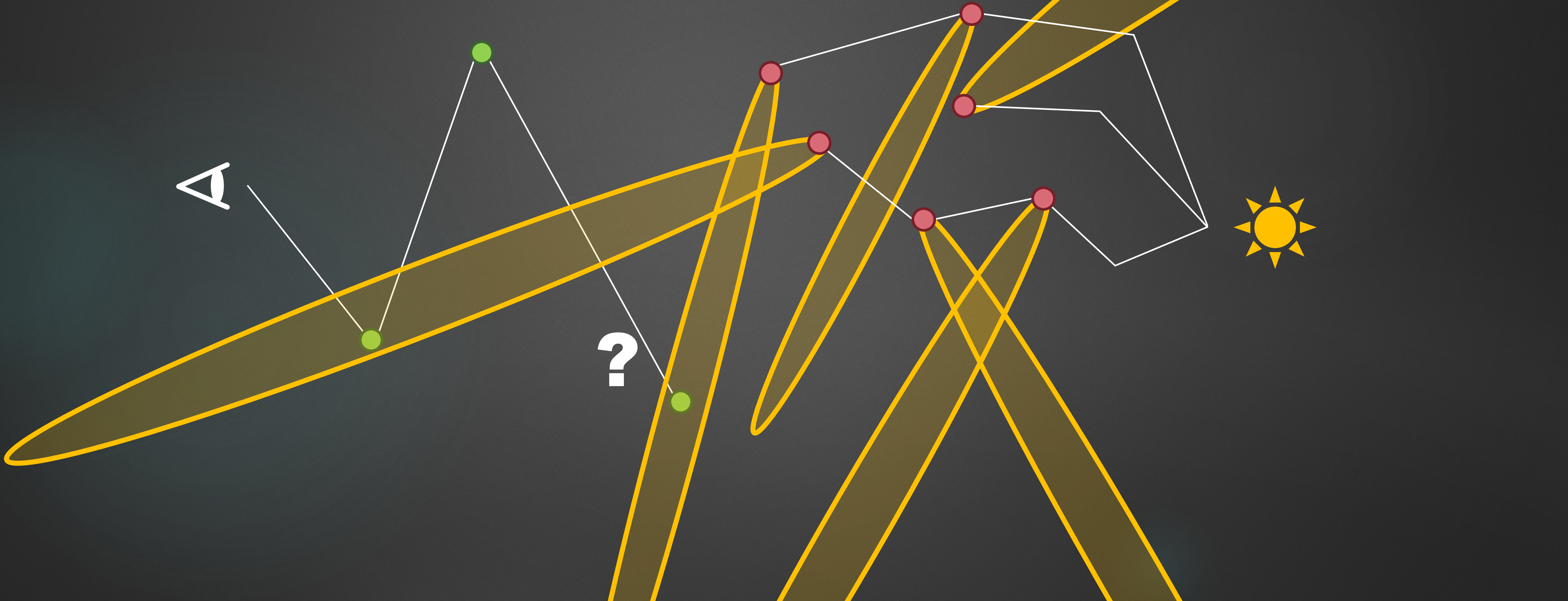
Tiled Culling

Bounding volume for eye vertices grouped in screen-space

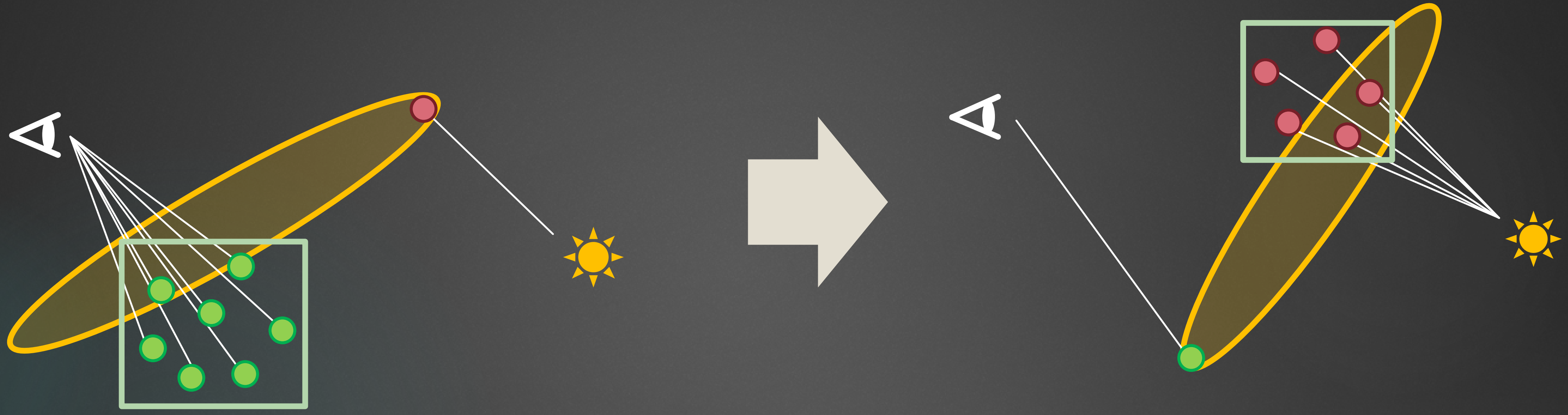


intersection test

Problem: Indirect Eye Vertices



Opposite Approach



Range from each light vertex
&
Bounding volume for eye vertices

Range from each eye vertex
&
Bounding volume for light vertices

Range from an Eye Vertex

- Probability of Russian roulette:

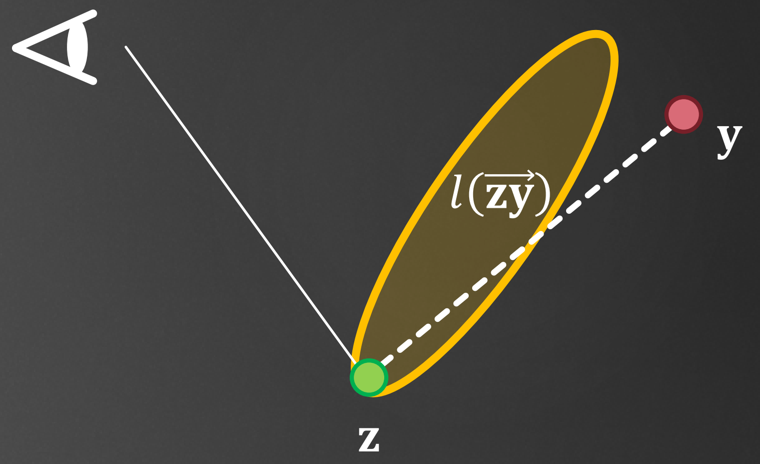
user-defined parameter GGX lobe

$$P(\mathbf{z},\mathbf{y}) = \min\left(\frac{Cq(\overrightarrow{\mathbf{zy}})}{\|\mathbf{y}-\mathbf{z}\|^2}, 1\right)$$

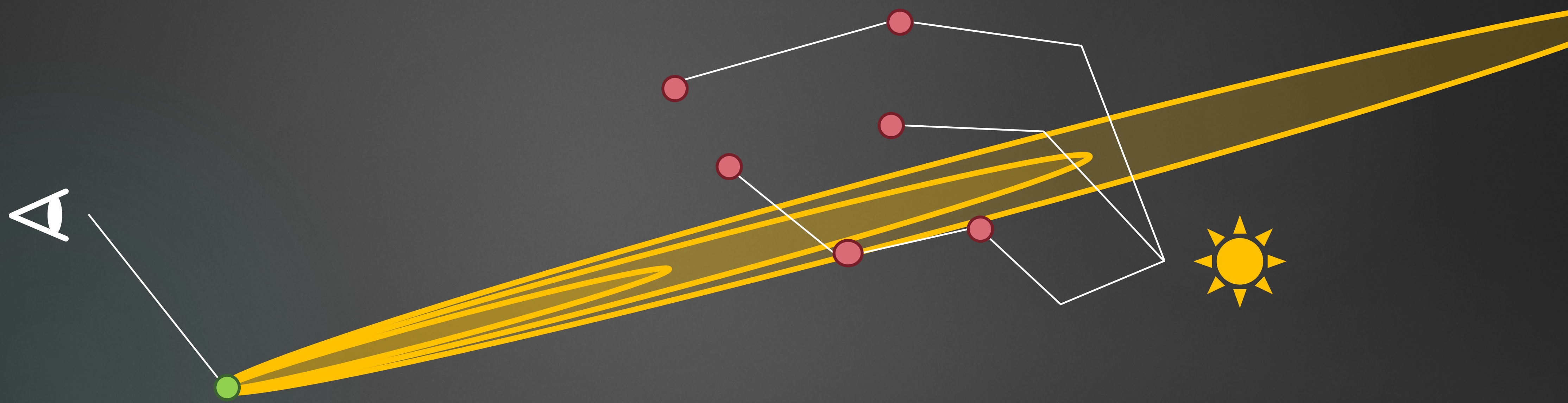
- Acceptance range from the eye vertex:

$$l(\overrightarrow{\mathbf{zy}}) \geq P^{-1}(\xi) = \sqrt{\frac{Cq(\overrightarrow{\mathbf{zy}})}{\xi}}$$

uniform random number $\in [0,1)$ generated for each light vertex

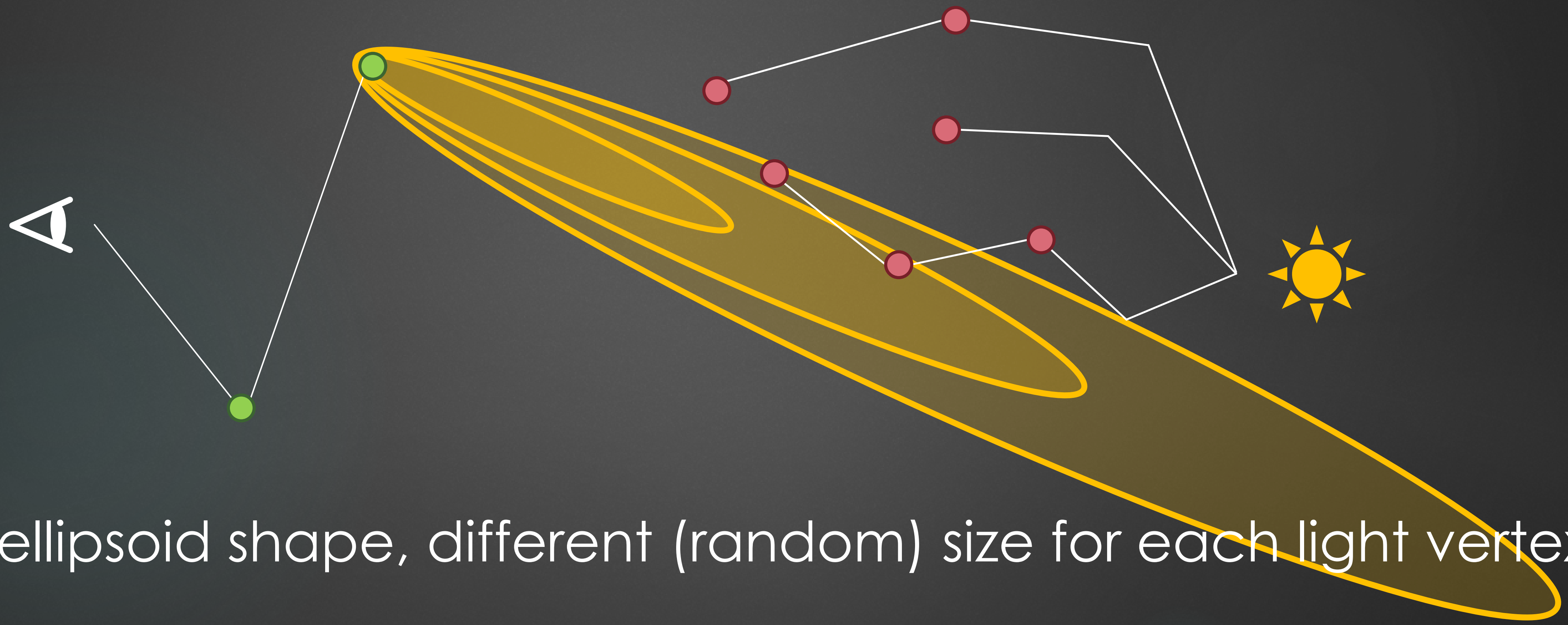


Backward Stochastic Light Culling



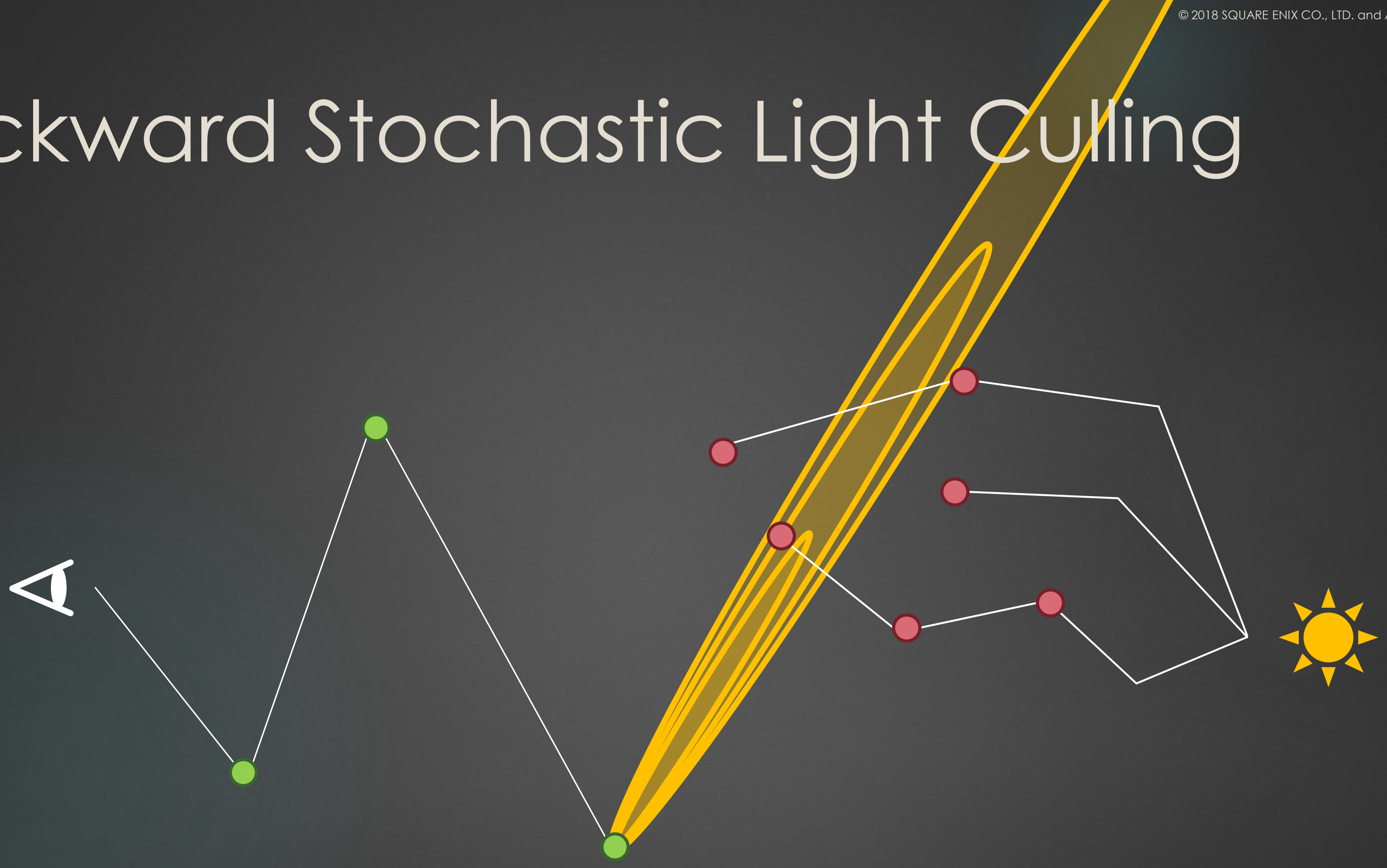
Same ellipsoid shape, different (random) size for each light vertex

Backward Stochastic Light Culling



Same ellipsoid shape, different (random) size for each light vertex

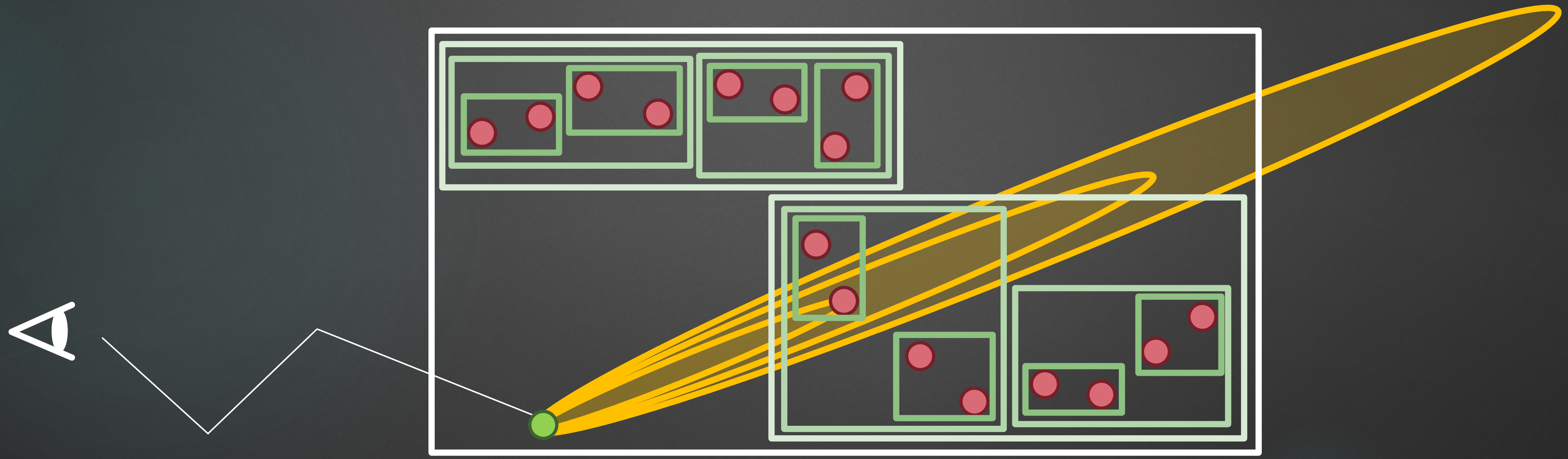
Backward Stochastic Light Culling



Same ellipsoid shape, different (random) size for each light vertex

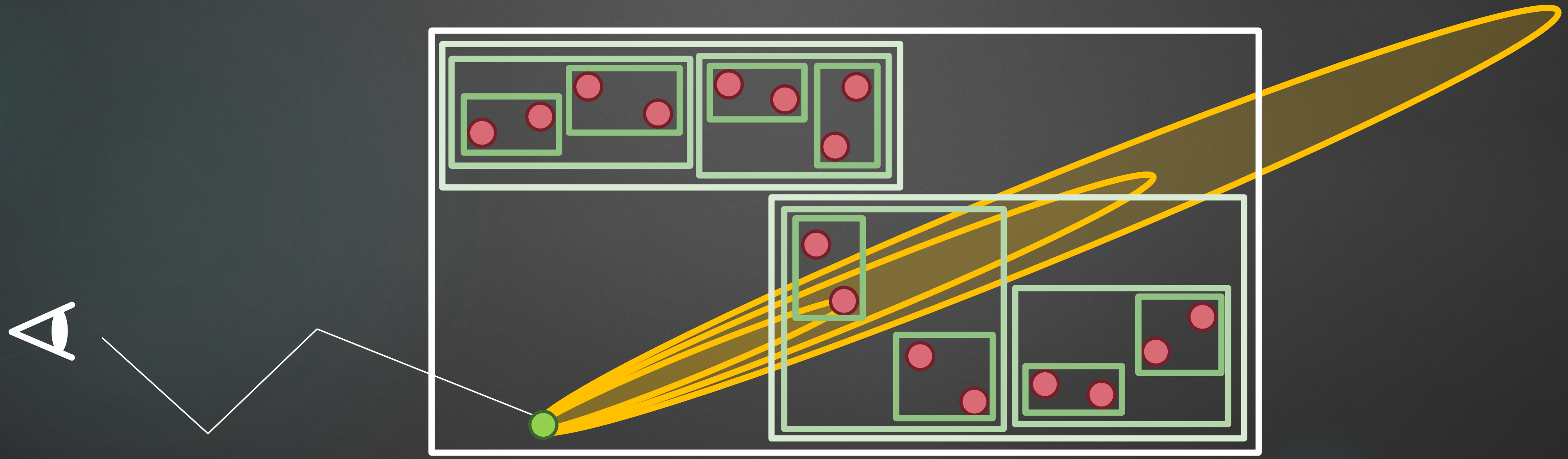
Culling Using BVH

- Build BVH for cached light vertices
- Hierarchical intersection tests between the ellipsoid and each BVH node



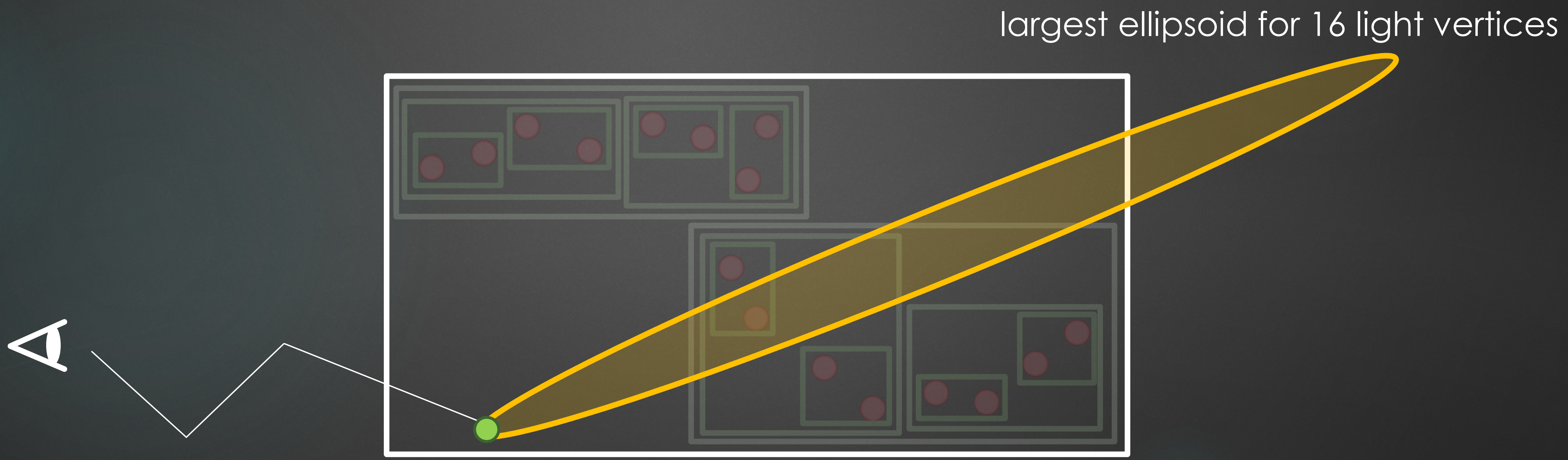
Random Ellipsoid Size

- Random ellipsoid size for each light vertex (i.e., leaf node)
- Use the largest size in each node for conservative intersection test



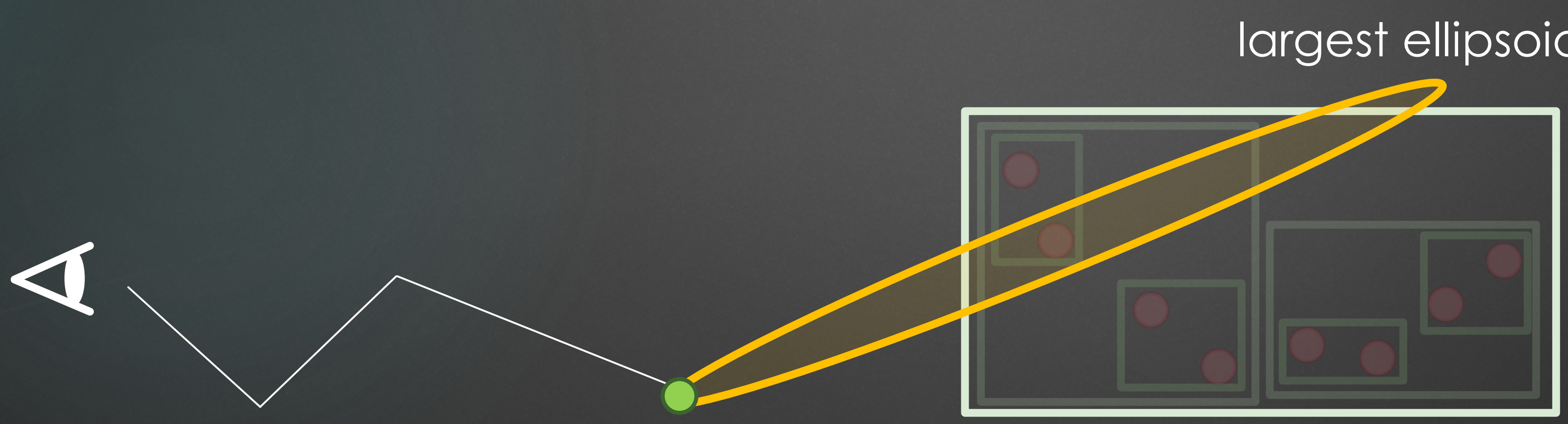
Random Ellipsoid Size

- Random ellipsoid size for each light vertex (i.e., leaf node)
- Use the largest size in each node for conservative intersection test



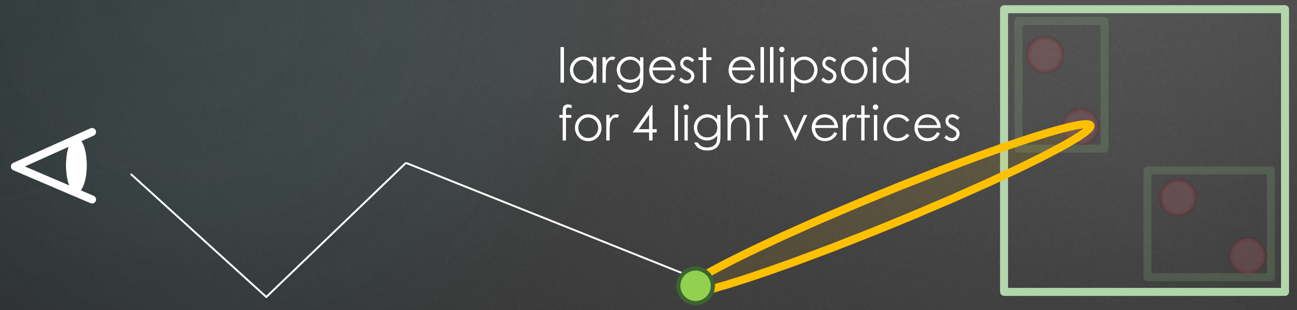
Random Ellipsoid Size

- Random ellipsoid size for each light vertex (i.e., leaf node)
- Use the largest size in each node for conservative intersection test



Random Ellipsoid Size

- Random ellipsoid size for each light vertex (i.e., leaf node)
- Use the largest size in each node for conservative intersection test



Random Ellipsoid Size

- Random ellipsoid size for each light vertex (i.e., leaf node)
- Use the largest size in each node for conservative intersection test

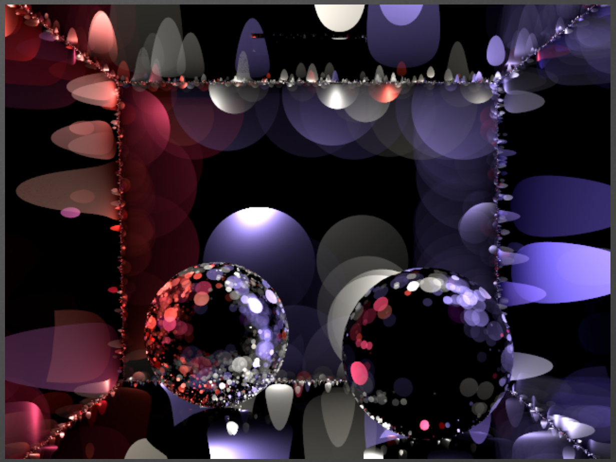
$$\max_{i \in L} l(\boldsymbol{\omega}) = \sqrt{\frac{Cq(\boldsymbol{\omega})}{\min_{i \in L} \xi_i}}$$

The largest size is given by the minimum of uniform random numbers

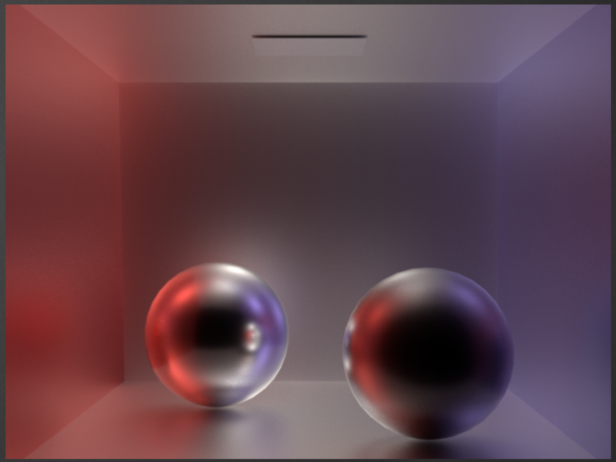
Precomputing Random Numbers?

- Assign a single random number to each light vertex in preprocessing
  - Similar to previous stochastic light culling [Tokuyoshi16,17]
- Store the minimum random number into each BVH node

Correlation of variance ☹



Precomputed



Reference

Generate random numbers on-the-fly

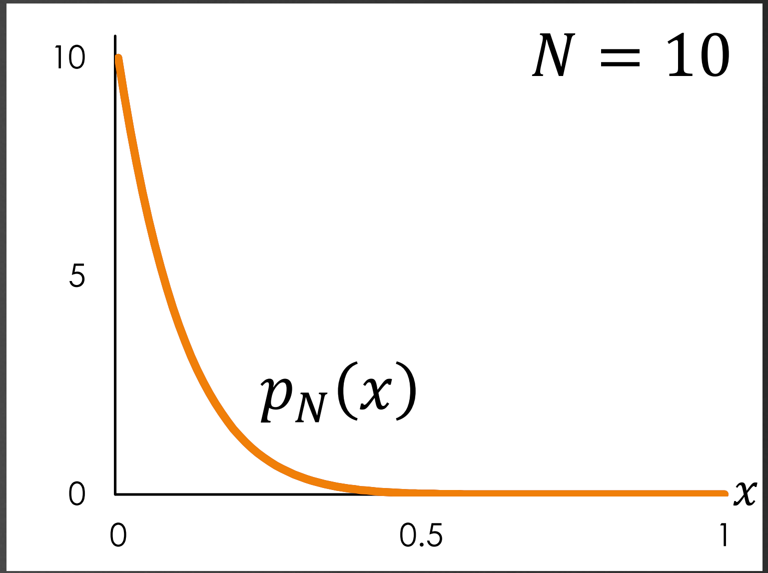
Distribution of the Minimum Random Number

PDF of the minimum value among N uniform random numbers:

$$p_N(x) = N(1-x)^{N-1}$$

Inverse of the CDF

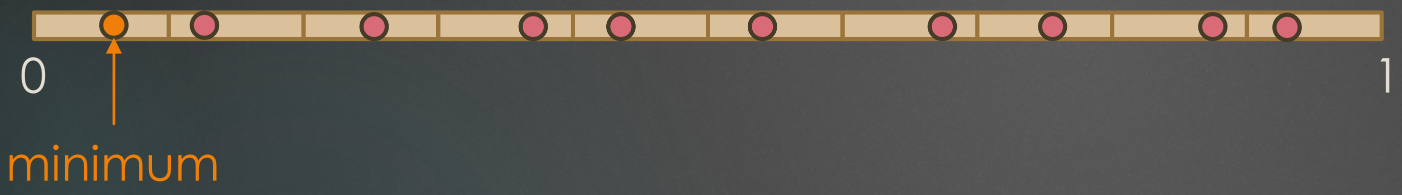
$$\min_{i \in [0,N)} \xi_i = c_N^{-1}(\xi) = 1 - (1-\xi)^{\frac{1}{N}}$$



Computable using only a single uniform random number $\xi \in [0,1)$

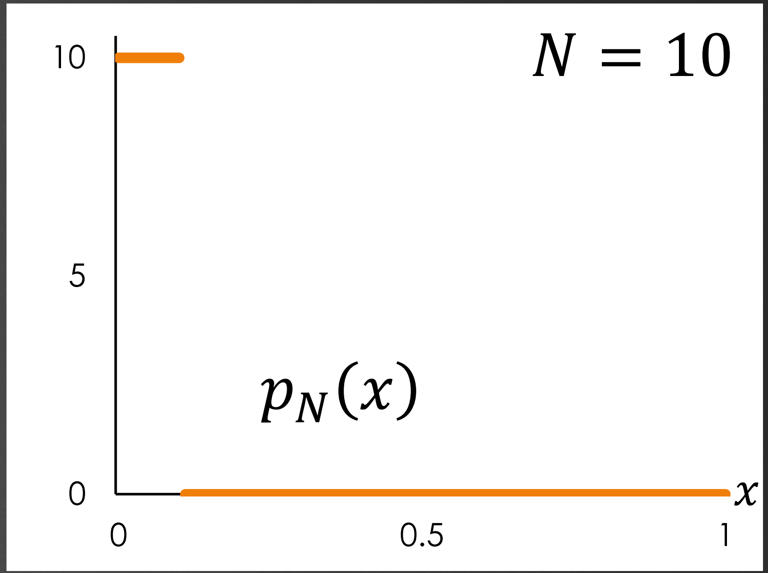
Case of 1D Stratified Sampling

- The minimum value is always within the lowest stratum

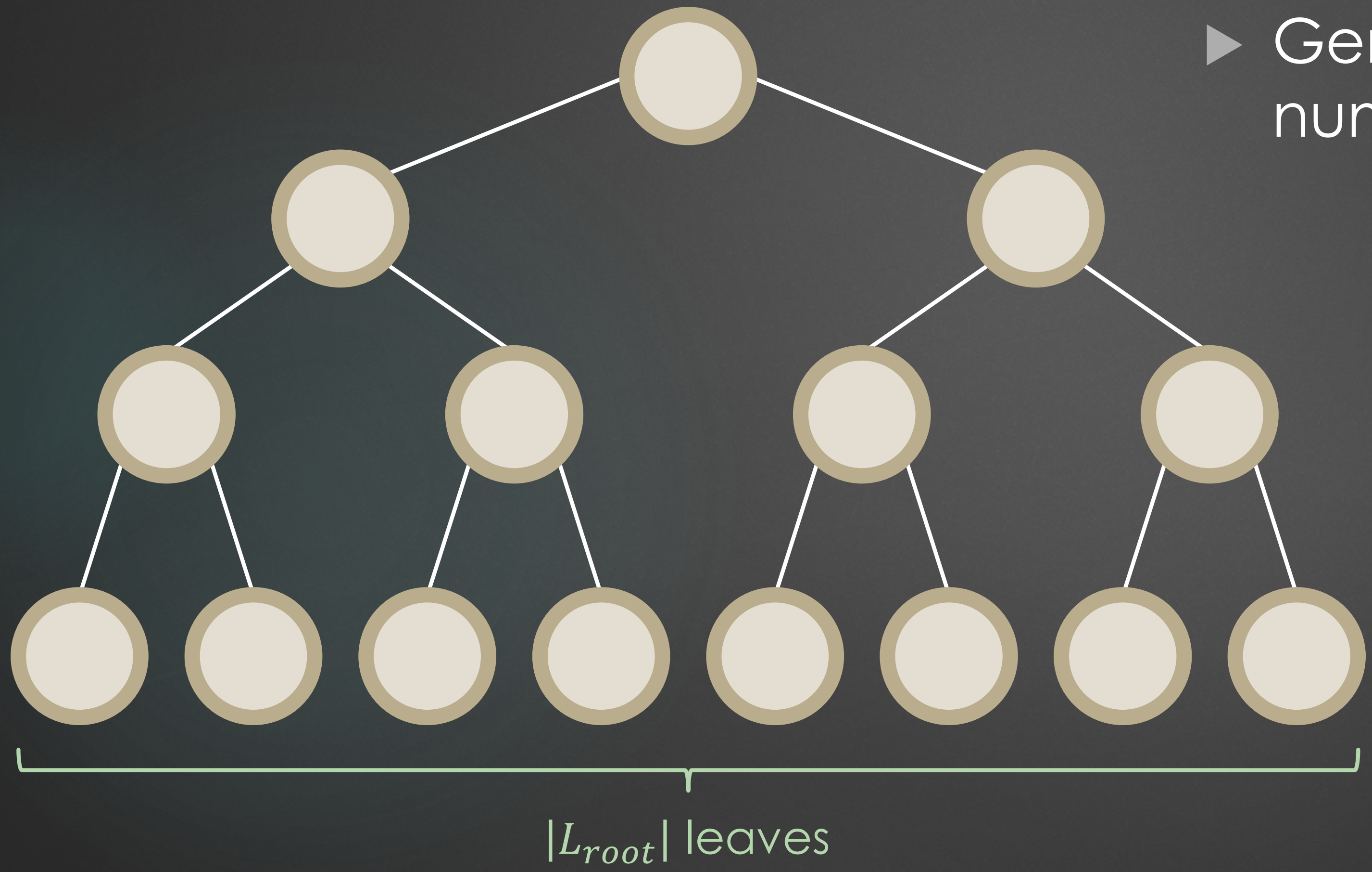


$$\min_{i\in[0,N)} \xi_i = c_N^{-1}(\xi) = \frac{\xi}{N}$$

Simple ☺



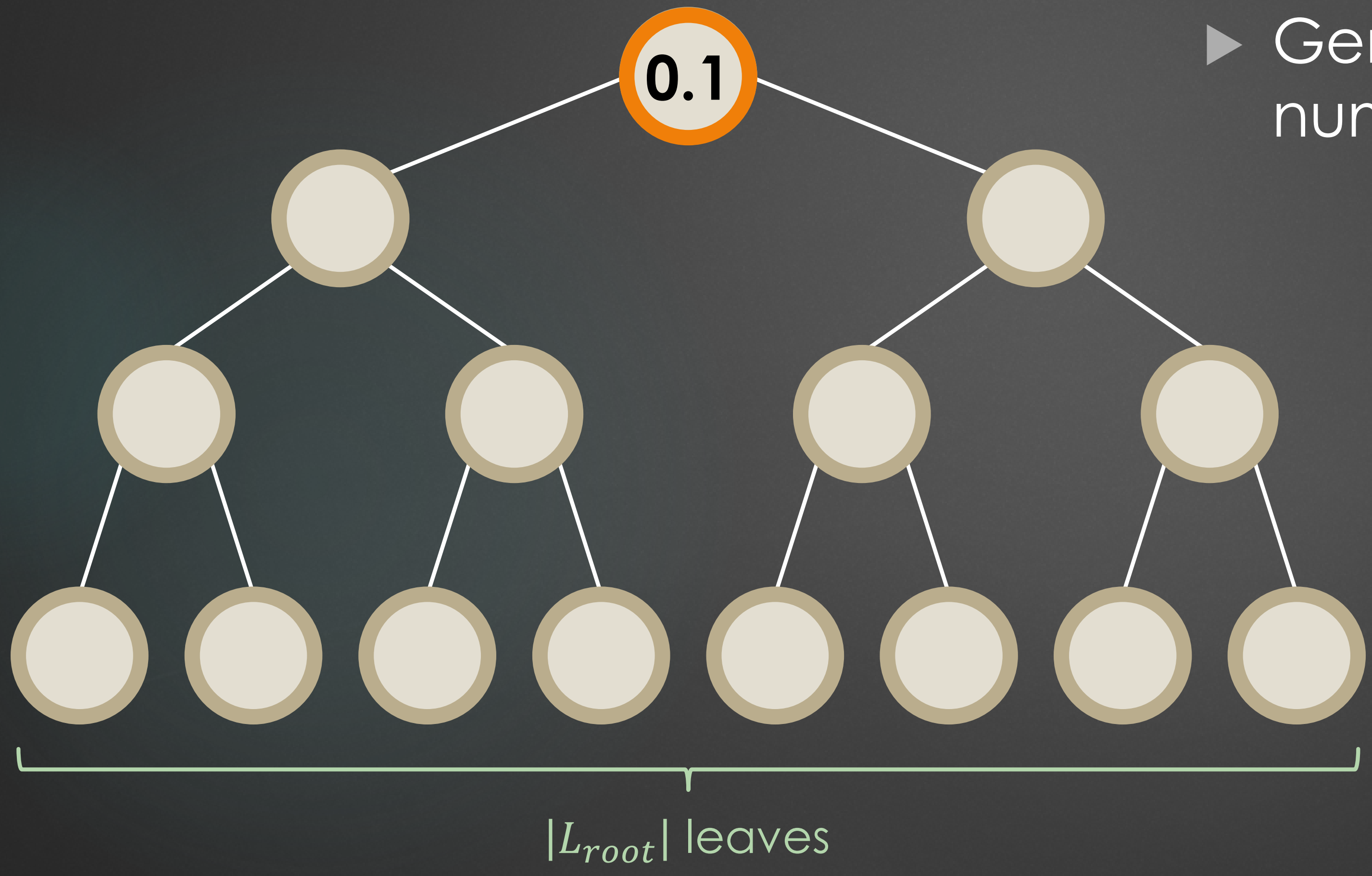
Semi-Stratified Random Number Assignment in Top-Down BVH Traversal



- Generate a minimum random number at the root node

$$\min_{i \in L_{root}} \xi_i = \frac{\xi}{|L_{root}|}$$

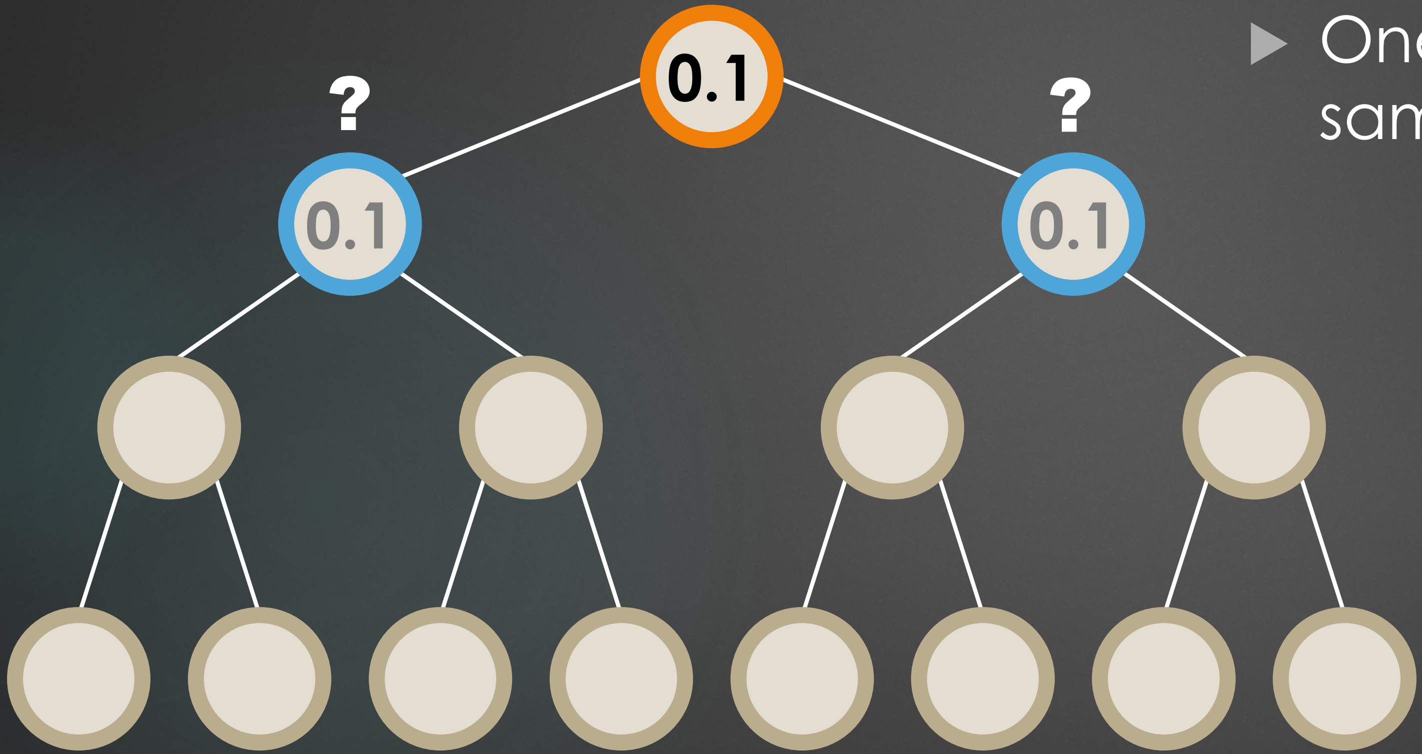
Semi-Stratified Random Number Assignment in Top-Down BVH Traversal



- Generate a minimum random number at the root node

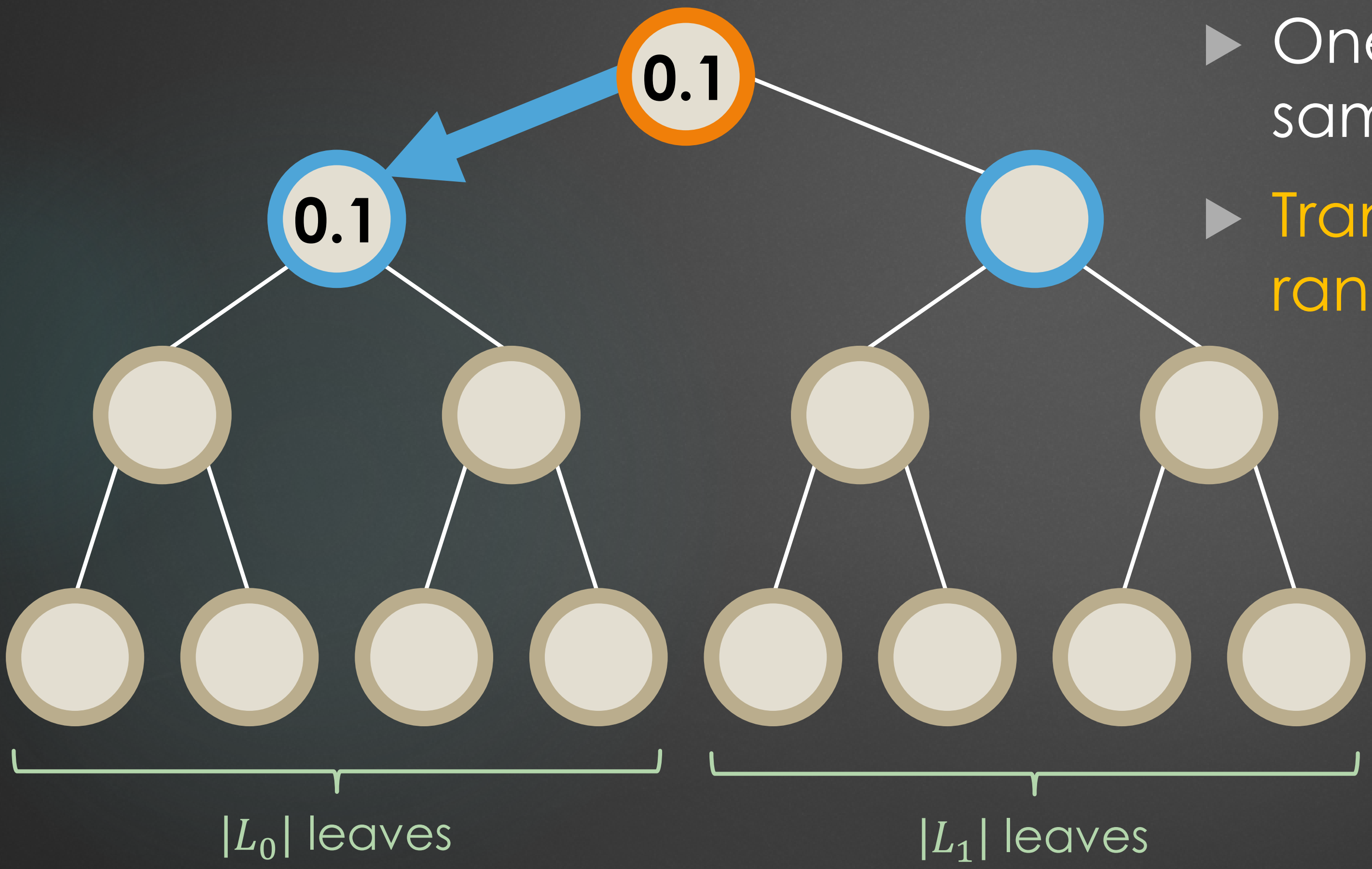
$$\min_{i \in L_{root}} \xi_i = \frac{\xi}{|L_{root}|}$$

Semi-Stratified Random Number Assignment in Top-Down BVH Traversal



- One of the child values must be the same as the parent's value

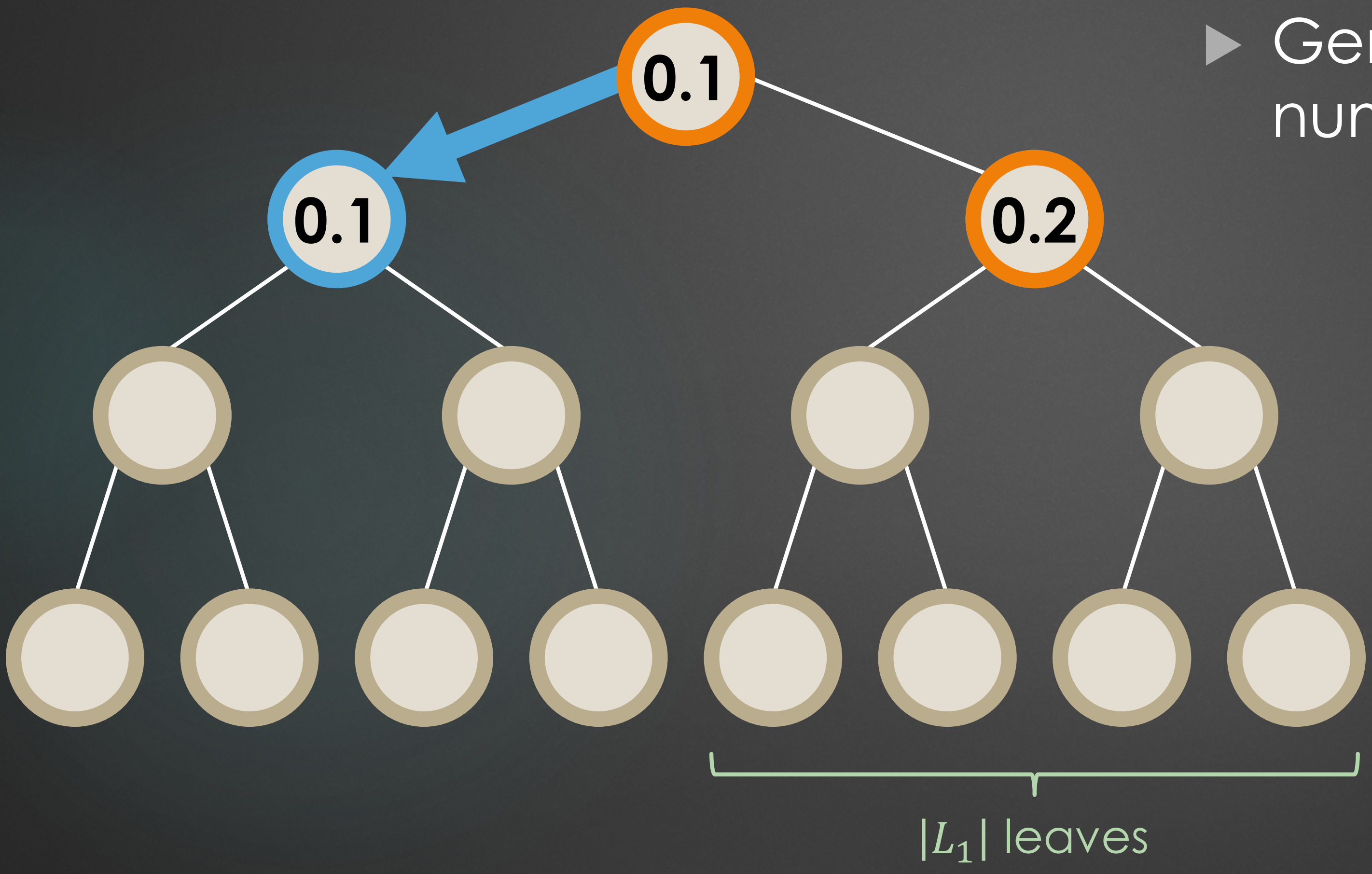
Semi-Stratified Random Number Assignment in Top-Down BVH Traversal



- One of the child values must be the same as the parent's value
- Transmit the parent's value to a randomly selected single child

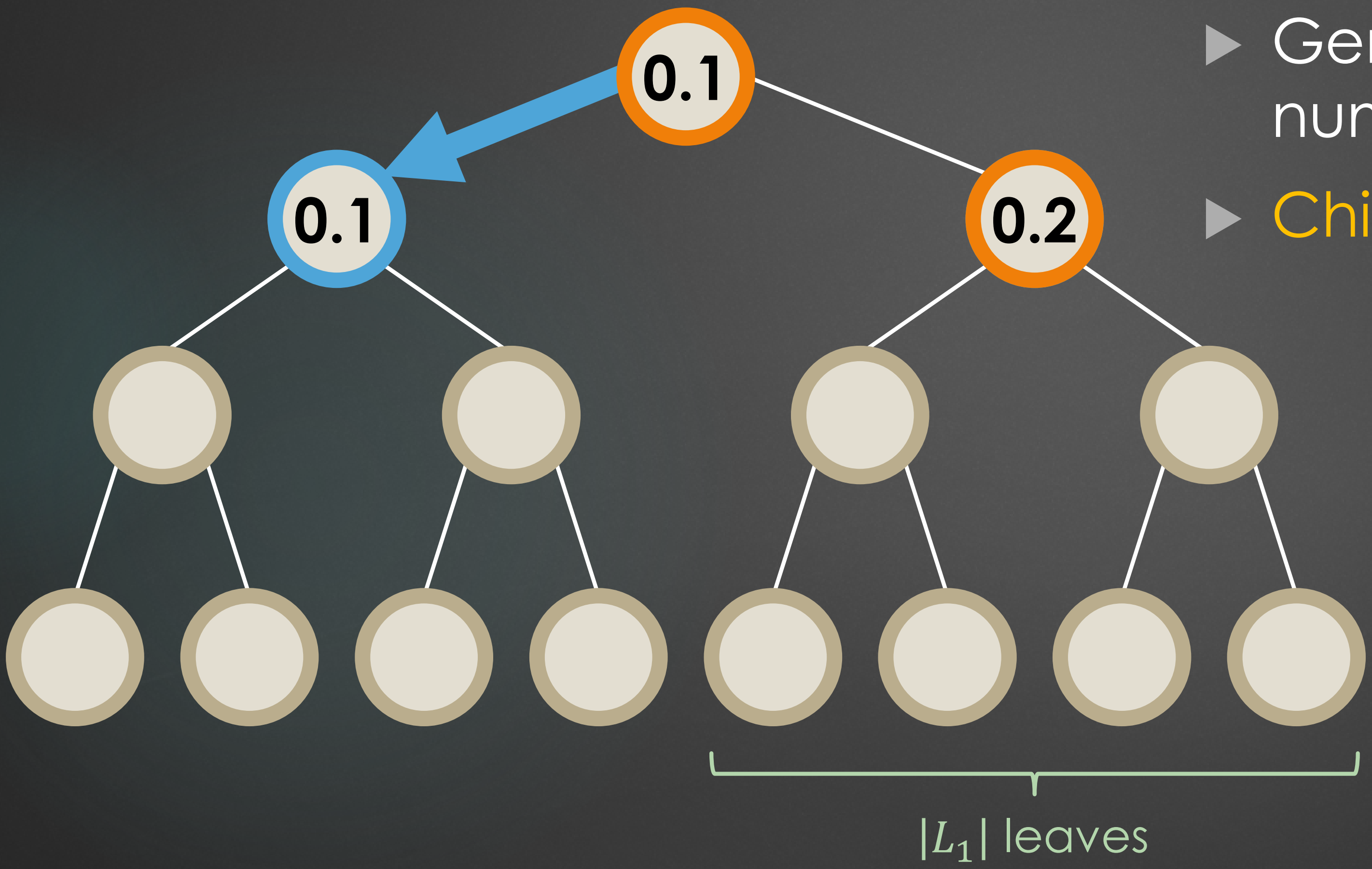
$$\text{Probability} = \frac{|L_0|}{|L_0| + |L_1|}$$

Semi-Stratified Random Number Assignment in Top-Down BVH Traversal



- Generate a new minimum random number for the other child

Semi-Stratified Random Number Assignment in Top-Down BVH Traversal



- Generate a new minimum random number for the other child
- Child's value ≥ parent's value

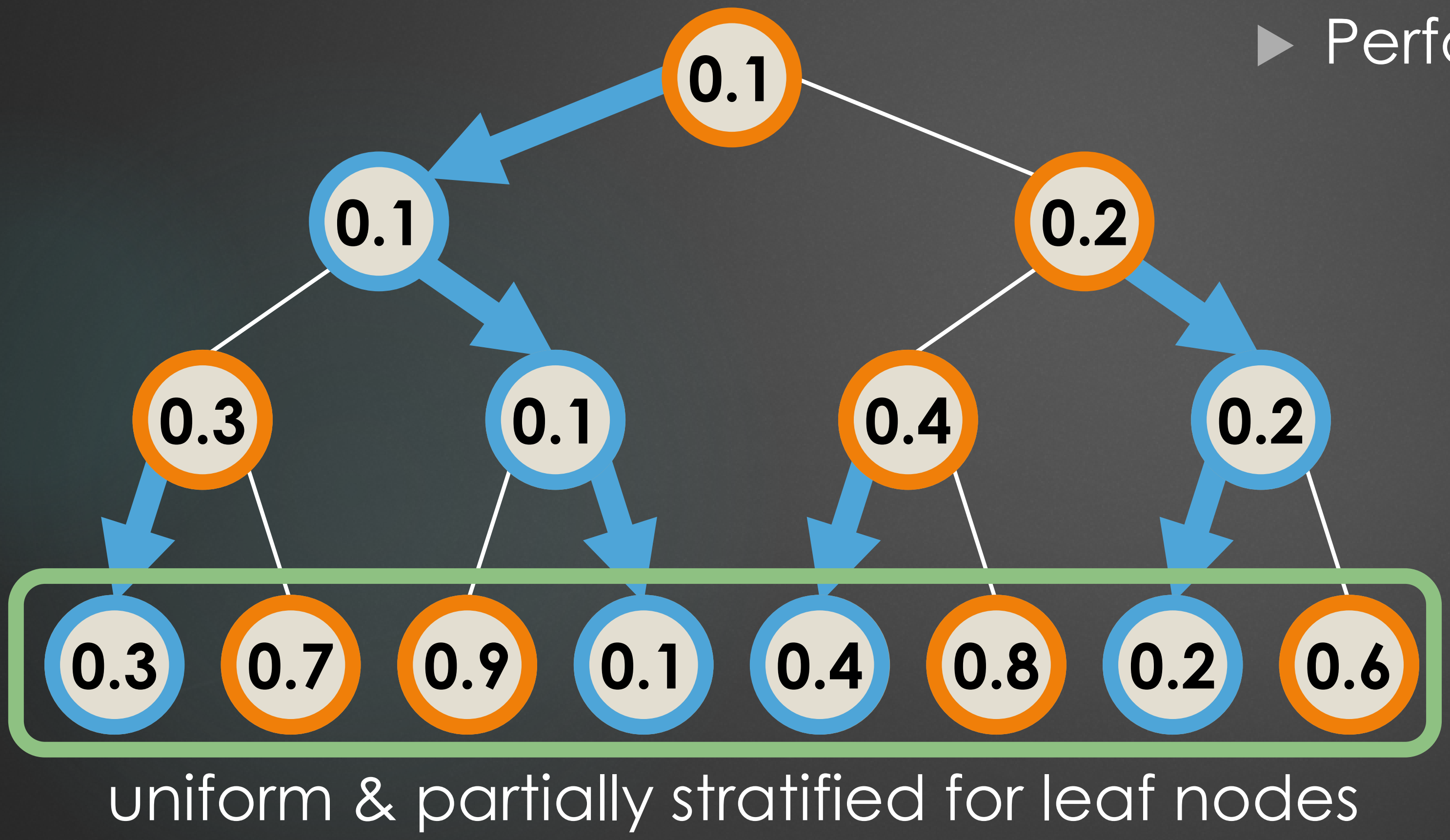
$$\min_{i \in L_1} \xi_i = s + (1 - s)\frac{\xi}{|L_1|}$$

Upper bound of the parent's stratum

Semi-Stratified Random Number Assignment in Top-Down BVH Traversal

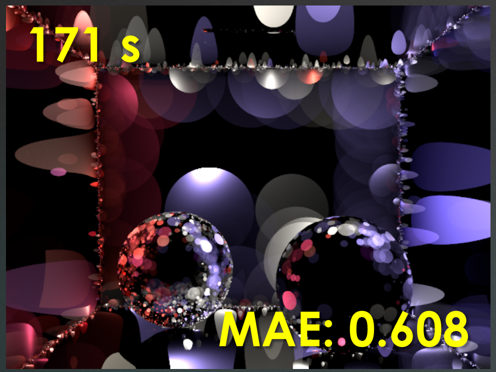
- Perform recursively in BVH traversal

Semi-Stratified Random Number Assignment in Top-Down BVH Traversal

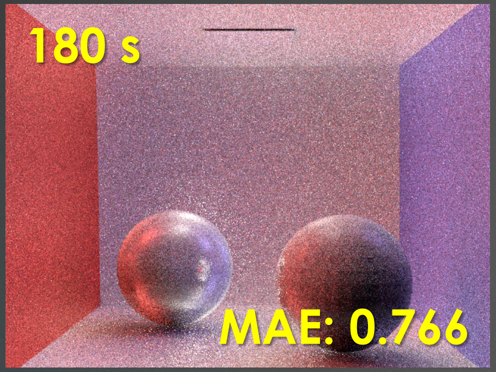


- Perform recursively in BVH traversal

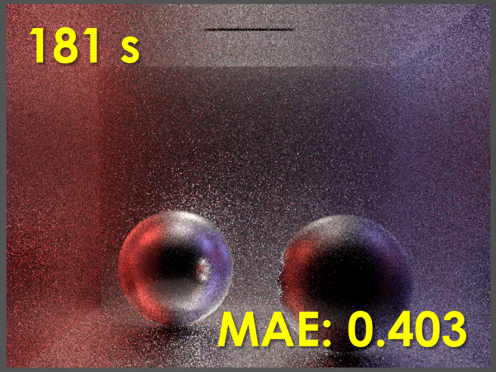
Comparison of Random Number Assignment Methods



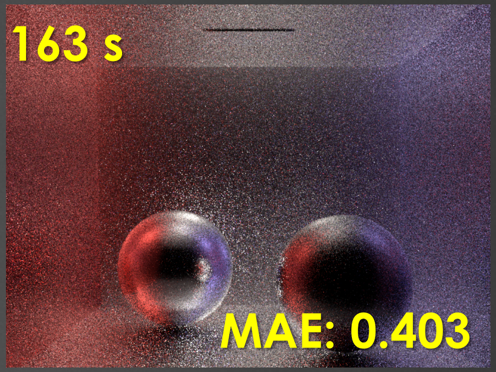
Precomputed



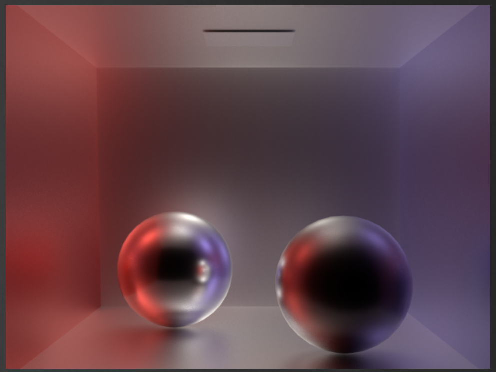
Non-stratified (single precision)



Non-stratified (double precision)



Semi-stratified (single precision)



Reference

$$c_N^{-1}(\xi) = 1 - (1 - \xi)^{\frac{1}{N}}$$

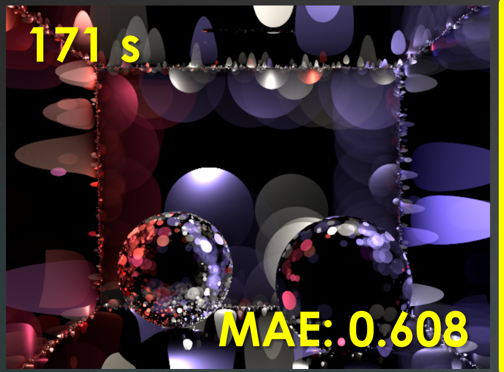
Noticeable precision error for large N ☹

$$c_N^{-1}(\xi) = \frac{\xi}{N}$$

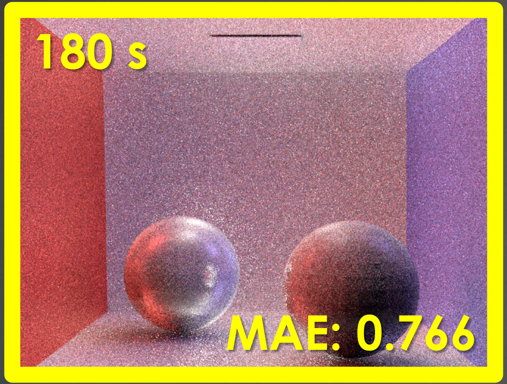
Fast & precise ☺

3.13M light vertices/iteration on average, CPU: AMD Ryzen™ Threadripper™ 1950X

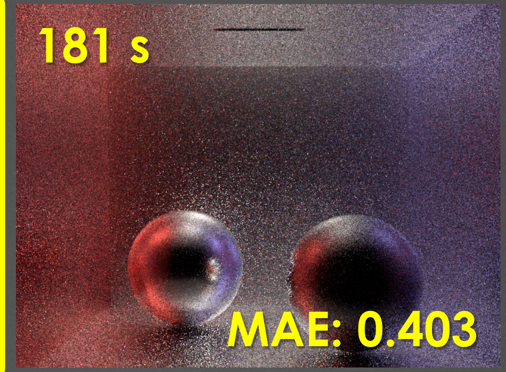
Comparison of Random Number Assignment Methods



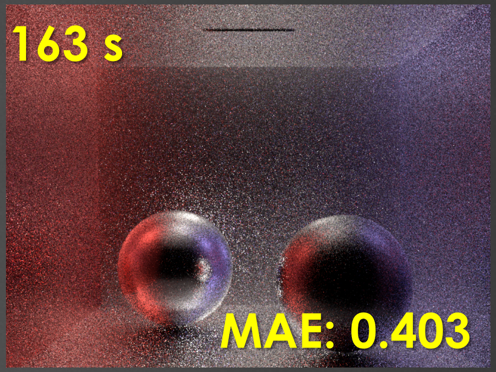
Precomputed



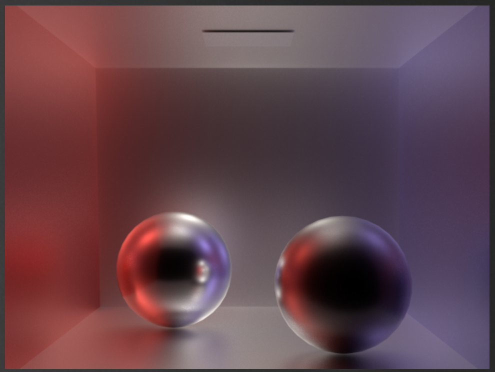
Non-stratified (single precision)



Non-stratified (double precision)



Semi-stratified (single precision)



Reference

$$c_N^{-1}(\xi) = 1 - (1 - \xi)^{\frac{1}{N}}$$

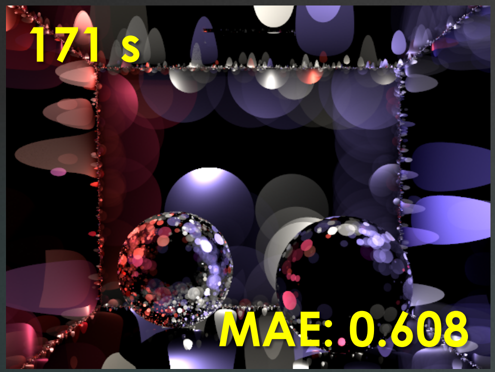
Noticeable precision error for large N ☹

$$c_N^{-1}(\xi) = \frac{\xi}{N}$$

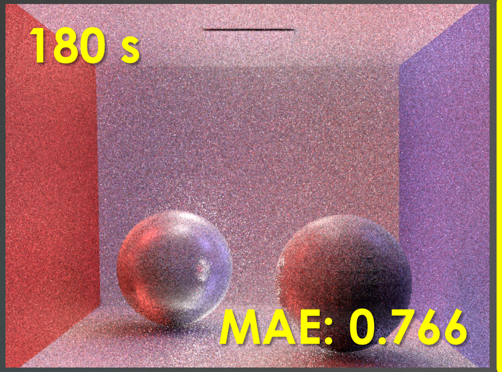
Fast & precise ☺

3.13M light vertices/iteration on average, CPU: AMD Ryzen™ Threadripper™ 1950X

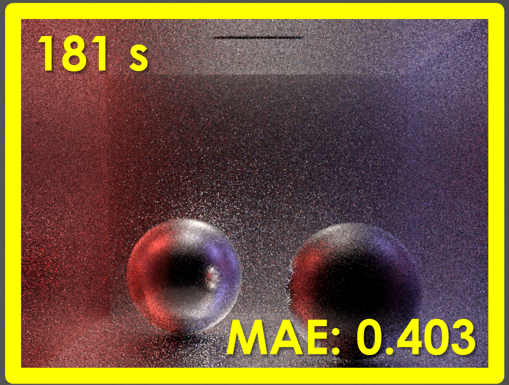
Comparison of Random Number Assignment Methods



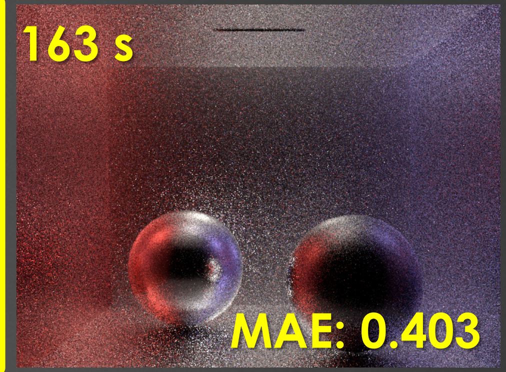
Precomputed



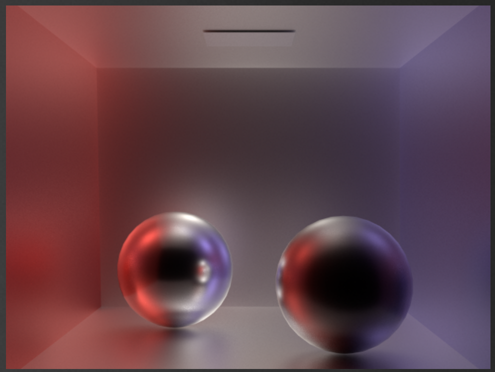
Non-stratified (single precision)



Non-stratified (double precision)



Semi-stratified (single precision)



Reference

$$c_N^{-1}(\xi) = 1 - (1 - \xi)^{\frac{1}{N}}$$

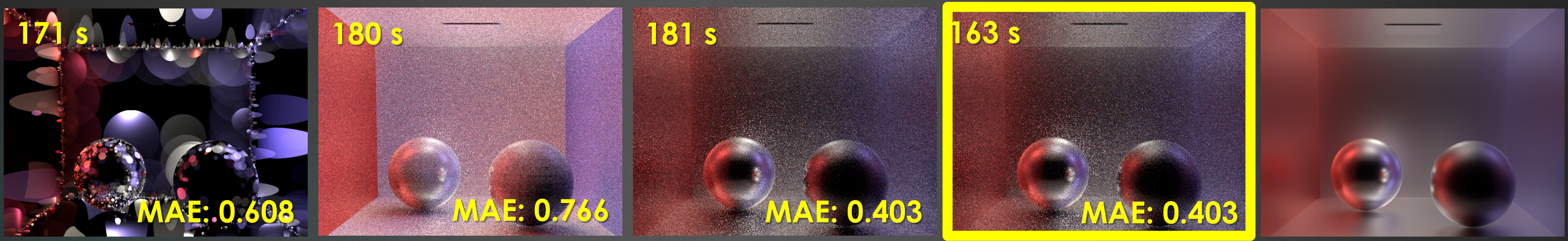
Noticeable precision error for large N ☹

$$c_N^{-1}(\xi) = \frac{\xi}{N}$$

Fast & precise ☺

3.13M light vertices/iteration on average, CPU: AMD Ryzen™ Threadripper™ 1950X

Comparison of Random Number Assignment Methods



Precomputed	Non-stratified (single precision)	Non-stratified (double precision)	Semi-stratified (single precision)	Reference

$$c_N^{-1}(\xi) = 1 - (1 - \xi)^{\frac{1}{N}}$$

Noticeable precision error for large N ☹

$$c_N^{-1}(\xi) = \frac{\xi}{N}$$

Fast & precise ☺

3.13M light vertices/iteration on average, CPU: AMD Ryzen™ Threadripper™ 1950X

Combination with BPT

+

=

Regular vertex connection

Stochastic light culling
(glossy reflection only)

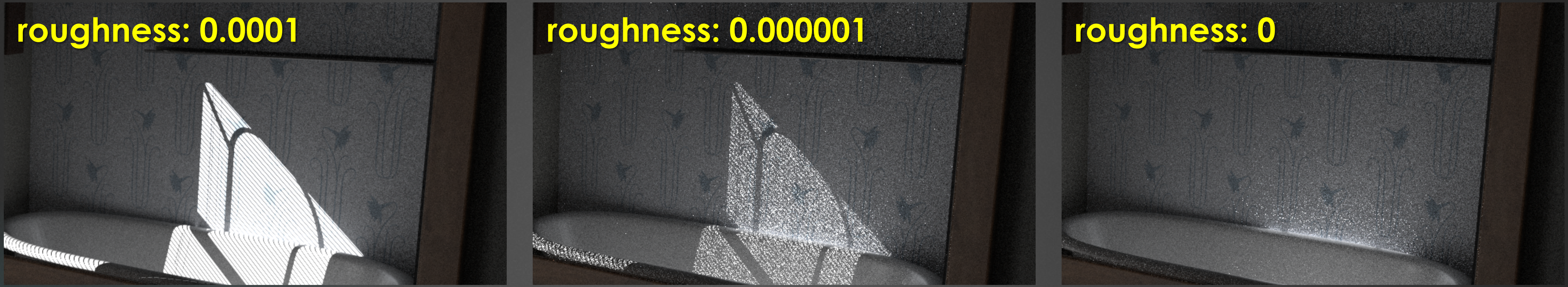
Multiple importance sampling (balance heuristic) [Veach95]

Equal-time Comparison (15 min)

AMD Ryzen™ Threadripper™ 1950X Processor

Caustics reflected on the mirror (GGX roughness: 0.0001)

Limitations



- Perfectly specular surfaces
- Ellipsoidal range to bound refractions & anisotropic BRDFs
  - Future work: efficient approximations [Belcour18, Conty18, Xu13]
- Correlation of paths due to path reuse
  - Balance heuristic does not take this correlation into account
  - Future work: correlation-aware MIS heuristics [Popov15, Jendersie18]

Conclusions

- Unbiased light vertex culling for BPT
  - Random range from each eye vertex
  - BVH of light vertices
- Decorrelation of variance
  - On-the-fly minimum random number generation in BVH traversal
- Limited to glossy reflections, but efficient for very highly glossy reflections
  - E.g., GGX roughness: 0.0001
  - Effective for specular-diffuse-glossy paths

Thank you for your attention

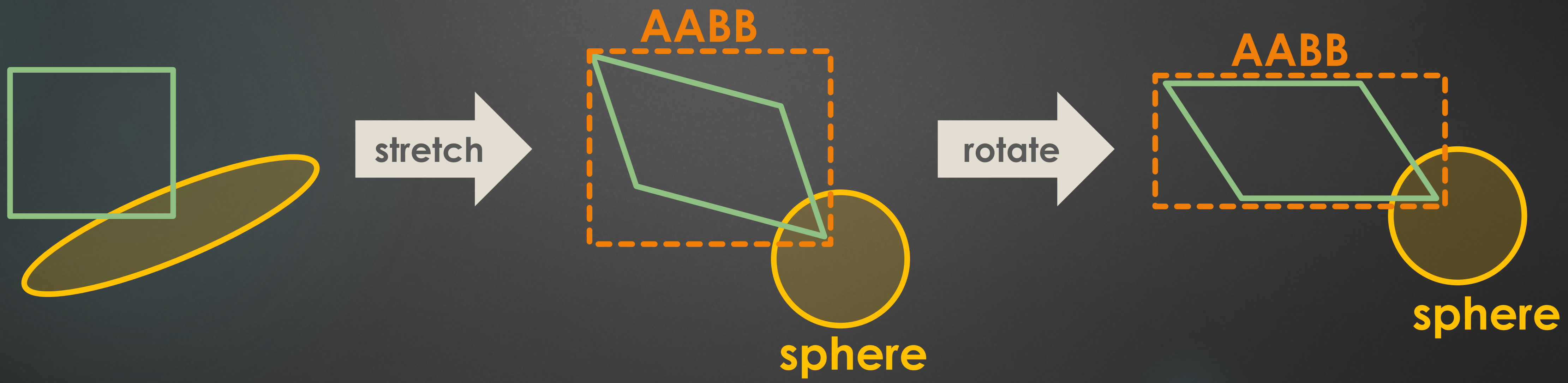
References

- Arvo J. 1990. A Simple Method for Box-Sphere Intersection Testing. In Graphics Gems, Academic Press Professional, Inc., 335–339
- Arvo J. and Kirk D. 1990. Particle Transport and Image Synthesis. SIGGRAPH Comput. Graph. 24, 4, 63–66
- Belcour L. 2018. Efficient Rendering of Layered Materials using an Atomic Decomposition with Statistical Operators. TOG 37, 4, 73:1–73:15
- Dachsbacher C. and Stamminger M. 2006. Splatting Indirect Illumination. In I3D '06. 93–100
- Conty A. and Lecocq P. 2018. Fast Product Importance Sampling of Environment Maps. In SIGGRAPH '18 Talks 69:1–69:2
- Jendersie J. and Grosch T. 2018. An Improved Multiple Importance Sampling Heuristic for Density Estimates in Light Transport Simulations. In EGSR '18 EI&I Track. 65–72
- Keller A. 1997. Instant Radiosity. In SIGGRAPH '97. 49–56
- Popov S., Ramamoorthi R., Durand F., and Drettakis G. 2015. Probabilistic Connections for Bidirectional Path Tracing. CGF 34, 4, 75–86
- Tokuyoshi Y. and Harada T. 2016. Stochastic Light Culling. JCGT. 5, 1, 35–60
- Tokuyoshi Y. and Harada T. 2017. Stochastic Light Culling for VPLs on GGX Microsurfaces. CGF 36, 4, 55–63
- Veach E. and Guibas L. J. 1995. Optimally Combining Sampling Techniques for Monte Carlo Rendering. In SIGGRAPH '95. 419–428
- Xu K., Sun W.-L., Dong Z., Zhao D.-Y., Wu R.-D. and Hu S.-M. 2013. Anisotropic Spherical Gaussians. TOG 32, 6, 209:1–209:11

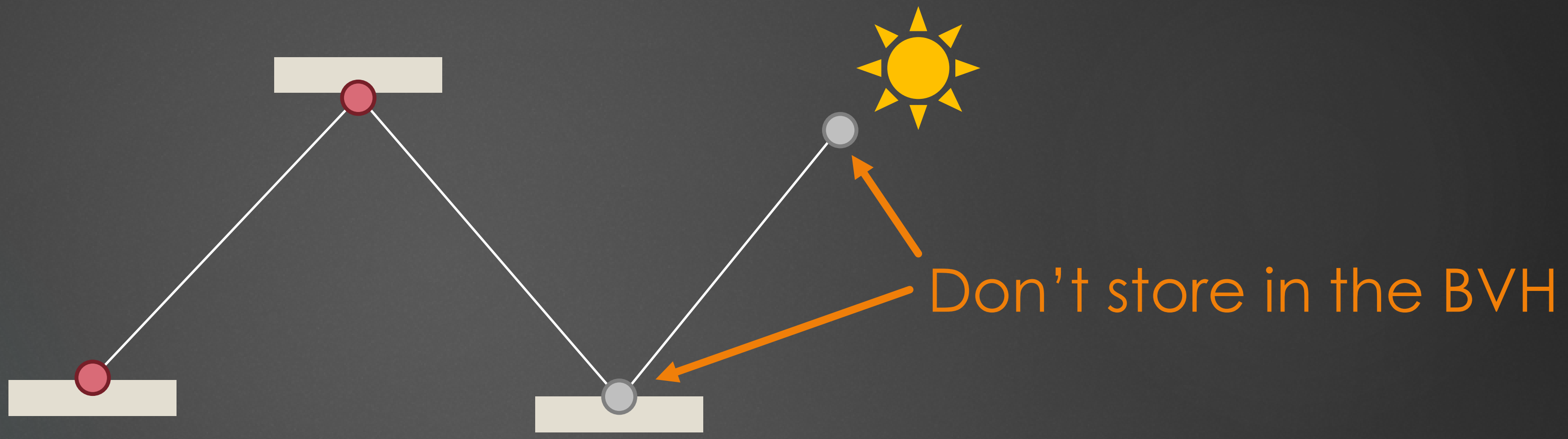
Backup

Rough Intersection Test for Culling

- Exact ellipsoid-box intersection test is expensive
- Replace with sphere-AABB intersection test [Arvo90] in a stretched space
- Rotate the test space to minimize the AABB



1st and 2nd Light Vertices



- Exclude 1st and 2nd light vertices from the BVH
- Efficiently sampled by other techniques in BPT
  - E.g., next event estimation