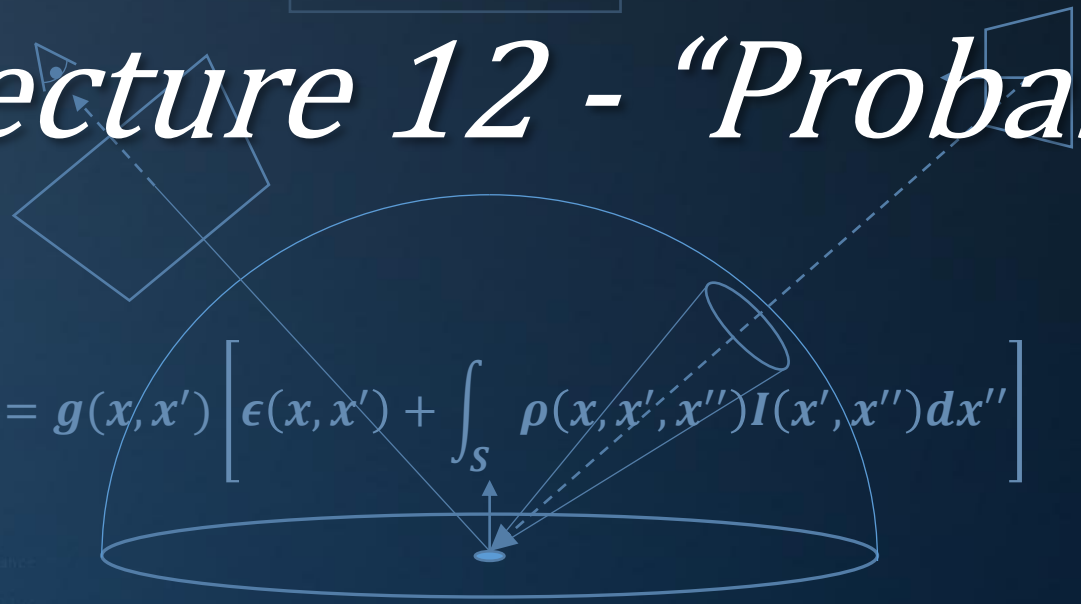
INFOMAGR – Advanced Graphics

Jacco Bikker - November 2022 - February 2023

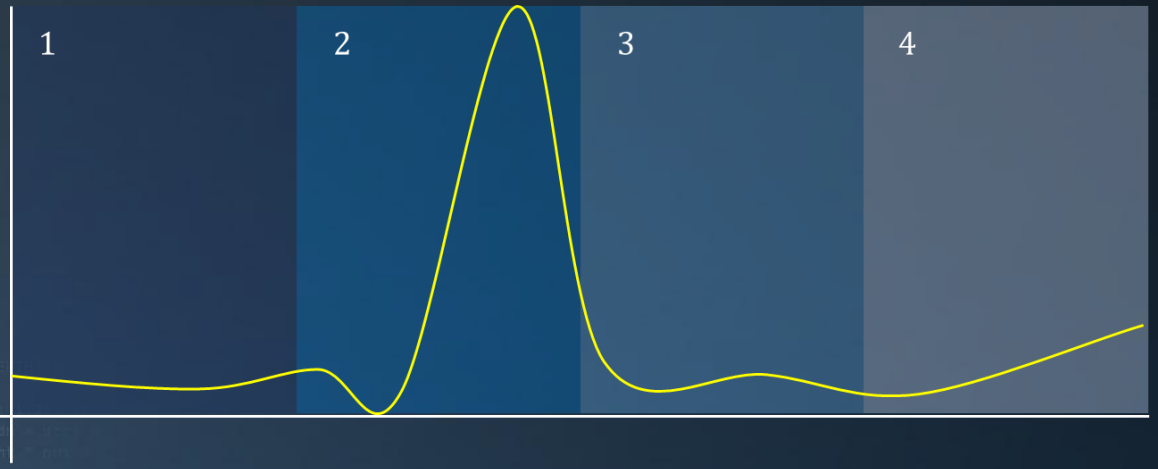
Lecture 12 - "Probability Theory"



$$I(x,x') = g(x,x')\left[\epsilon(x,x') + \int_S \rho(x,x',x'')I(x',x'')dx''\right]$$

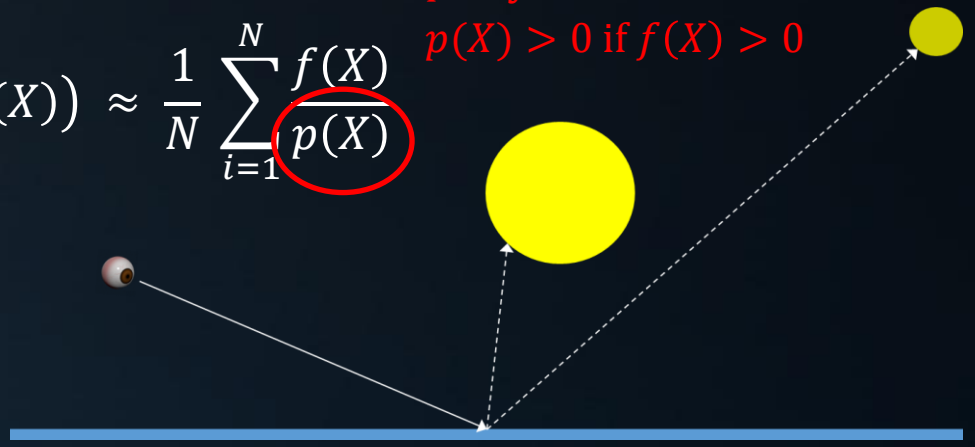
Welcome!

Previously in Advanced Graphics…



$$E\big(f(X)\big) \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(X)}{p(X)}$$

pdf: $\int = 1$,
$p(X) > 0$ if $f(X) > 0$



$$L_o(p,\omega_i) \approx \frac{lights}{N}\sum_{i=1}^{N} f_r(p,\omega_o,P)\, L_d^J(p,P)\, V(p \leftrightarrow P)\frac{A_{L_d^J}\cos\theta_i \cos\theta_o}{\| p-P \|^2}$$

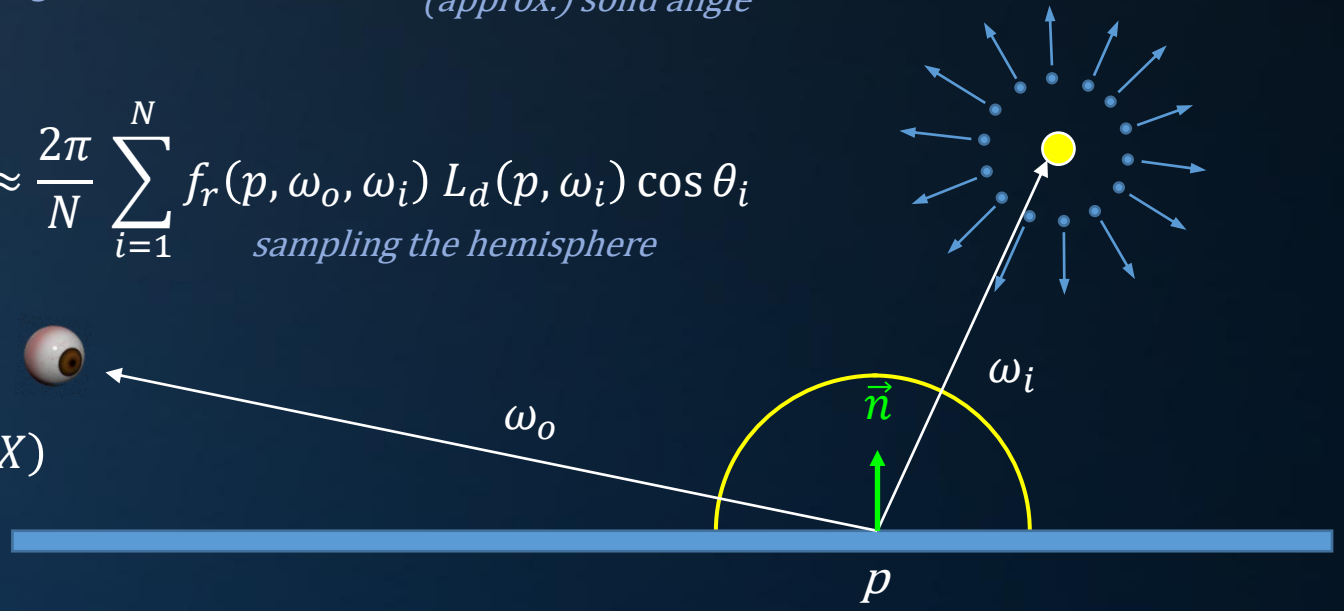
sampling a light

(approx.) solid angle

$$L_o(p,\omega_o) \approx \frac{2\pi}{N}\sum_{i=1}^{N} f_r(p,\omega_o,\omega_i)\, L_d(p,\omega_i)\cos\theta_i$$

sampling the hemisphere

$$E\big(f(X)\big) \approx \frac{B-A}{N}\sum_{i=1}^{N} f(X)$$



Today's Agenda: *Monte Carlo*

- Sampling an Area Light with One Ray
- Sampling Multiple Area Lights with One Ray
- Difficult Cases: Spherical Lights, Occluded Lights
- The Random Walk
- Random Walk with Next Event Estimation
- Digest
- P3 Topics

Lights

Case 1: Point Light

Situation:

- surface point: location $\boldsymbol{p}$, normal $\vec{N}$;
- point light: location $\boldsymbol{e}$, intensity $\boldsymbol{I}$ (in Watt, or joule per second);
- distance between $\boldsymbol{p}$ and $\boldsymbol{e}$: d.
- unit vector from $\boldsymbol{p}$ to e: $\vec{L}$.

Flux leaving $\boldsymbol{e}$: $\boldsymbol{I}$ joules per second.
Flux arriving at a sphere, radius r, surface $4\pi r^2$ around $\boldsymbol{e}$: $\boldsymbol{I}$
(Ir)radiance arriving on that sphere: $\frac{\boldsymbol{I}}{4\pi r^2}$ (W/m^2)
Flux arriving per steradian: $\frac{\boldsymbol{I}}{4\pi}$

Steradians for a unit area surface patch at location $\boldsymbol{p}$: $\sim \frac{\vec{N}\cdot\vec{L}}{d^2}$

This is the solid angle of the unit area surface patch as seen from $\boldsymbol{e}$, or:
The area of the patch projected on the unit sphere around $\boldsymbol{e}$.

Light arriving at p from a point light at distance d:

$\boldsymbol{I}\ \frac{\vec{N}\cdot\vec{L}}{4\pi d^2}$ per unit surface area.

This is the *projected radiance*, i.e. *irradiance* from the light at e arriving at point p.

The contribution of multiple lights is summed.

Lights

Case 2: Area Light

SA (in steradians): $0..2\pi$; area of the projection of the light on the hemisphere.

Situation:

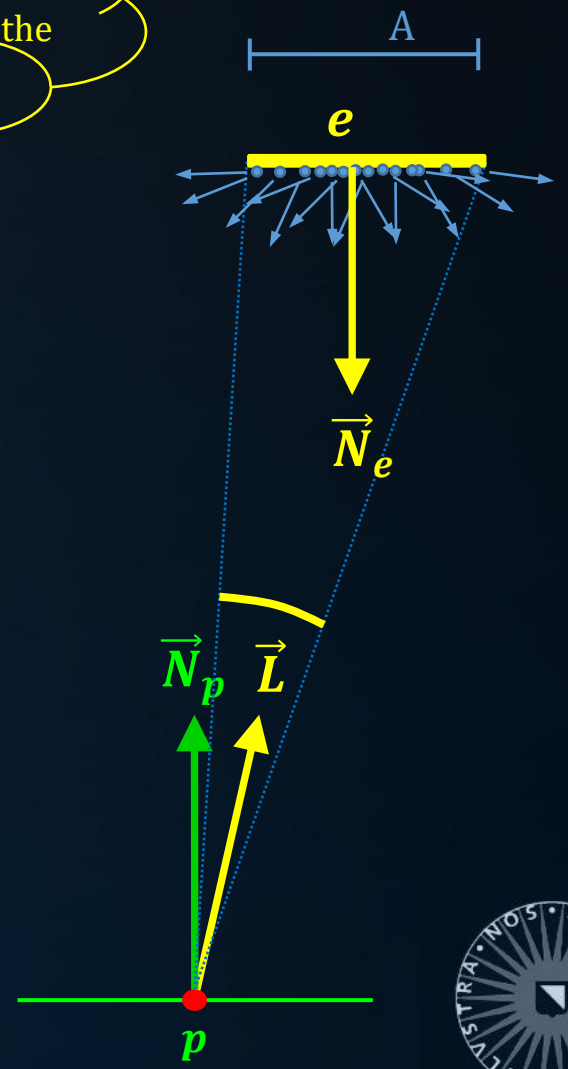
- surface point, location $\boldsymbol{p}$, normal $\vec{N}_p$;
- single-sided area light **e**, intensity $\boldsymbol{I}$, area **A**, normal $\vec{N}_e$.

Steradians for the area light, as seen from $\boldsymbol{p}$: $\frac{A(\vec{N}_e \cdot -\vec{L})}{d^2}$ *(approximately)*.

The *radiance* arriving from **e** at $\boldsymbol{p}$ is: $\boldsymbol{I}\,\boldsymbol{SA}$, $\boldsymbol{SA} \approx \frac{A(\vec{N}_e \cdot -\vec{L})}{d^2}$

The *irradiance* (joules per second per unit area) from **e** at $\boldsymbol{p}$ is:

$$\boldsymbol{I}\ SA\ (\vec{N}_p \cdot \vec{L}) \approx \boldsymbol{I}\ \frac{A(\vec{N}_e \cdot -\vec{L})(\vec{N}_p \cdot \vec{L})}{d^2}\ .$$



Lights

Sampling an Area Light

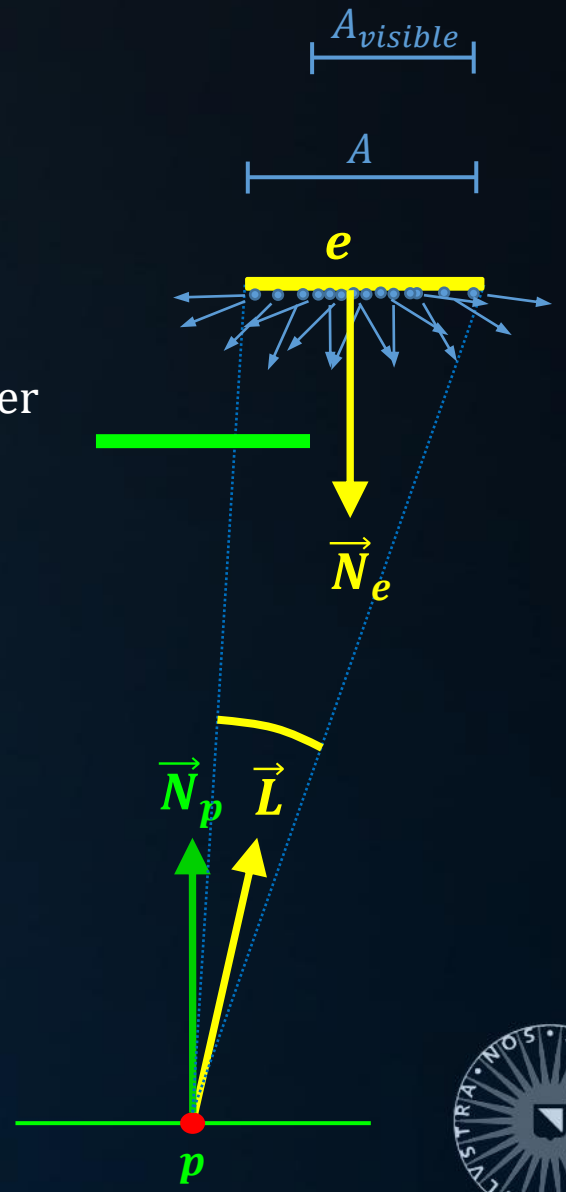
The irradiance (joules per second per unit area) is: $I \frac{A_{visible}(\vec{N}_e \cdot -\vec{L})(\vec{N}_p \cdot \vec{L})}{d^2}$.

Here, $A_{visible}$ is the *visible* area of the light source. $A_{visible}$ may be smaller than A in the presence of occluders.

We send 1 million rays to the light source. N rays reach the light source. The visible area is estimated as $A_{visible} = A \frac{N}{1{,}000{,}000}$.

Now, we send a *single* ray to the light source. The probability of a ray reaching the light source is ρ. Now, $A_{visible} = A\,\rho$.

For this single ray, the answer is usually wrong. However, *on average* the answer is correct.



Today's Agenda: *Monte Carlo*

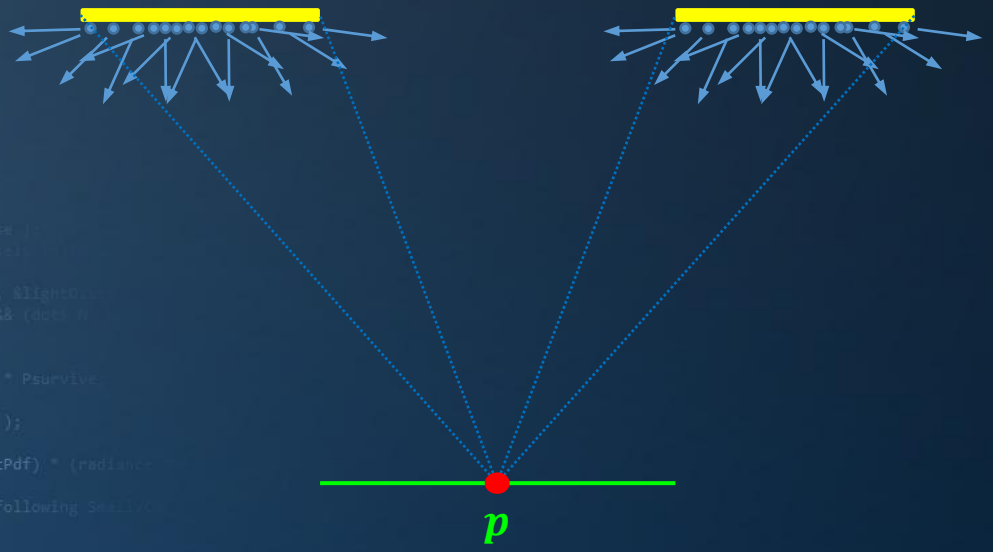
- Sampling an Area Light with One Ray
- Sampling Multiple Area Lights with One Ray
- Difficult Cases: Spherical Lights, Occluded Lights
- The Random Walk
- Random Walk with Next Event Estimation
- Digest
- P3 Topics

Lights

Sampling Multiple Lights

"To sample N lights with a single ray, chose a random light, and multiply whatever the ray returns by N".

Situation: *two lights*.



Generalized:

If we have N lights, and we sample each with a probability ρ_i, we scale the contribution by $\frac{1}{\rho_i}$ to get an unbiased sample of the set of N lights.

Any ρ_i is valid, as long as $\sum \rho_i = 1$ and $\rho_i > 0$ unless we know the sample will yield 0.

Mental steps:

- If the lights would have been point lights, we would have sampled both and summed the results.
- We know that we can sample an area light with a single ray. So, we sample both using a single ray, and sum the results.
- Using one ray, we could sample alternating lights. Since each light is now sampled in half the cases, we should increase the result we get each time by 2.
- Or, we can sample a randomly selected light. On average, each light is again sampled in half of the cases, so we scale by 2.
- In other words, we scale by 1/50%=2, where 50% is the probability of selecting a light.

Today's Agenda: *Monte Carlo*

- Sampling an Area Light with One Ray
- Sampling Multiple Area Lights with One Ray
- Difficult Cases: Spherical Lights, Occluded Lights
- The Random Walk
- Random Walk with Next Event Estimation
- Digest
- P3 Topics

Lights

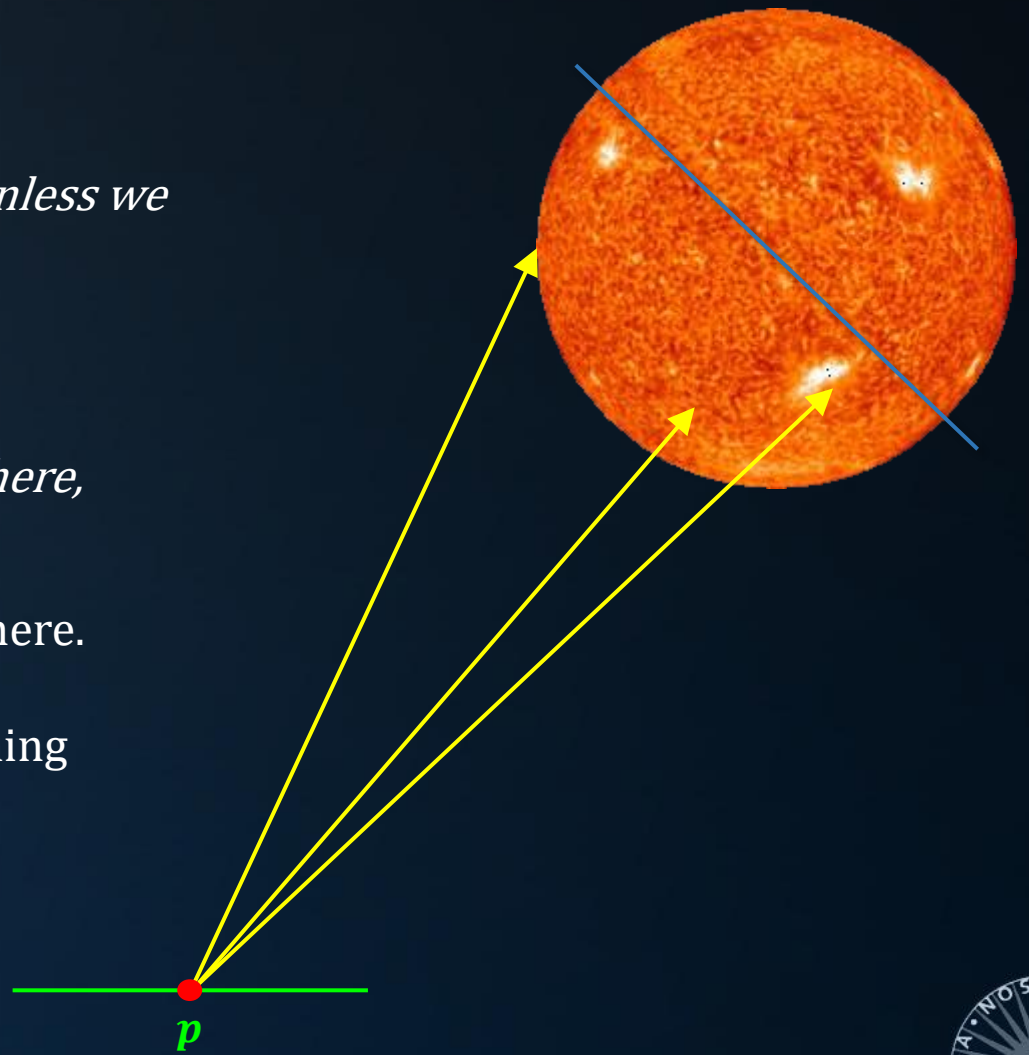
Sampling a Spherical Light

"Any ρ_i is valid, as long as $\sum \rho_i = 1$ and $\rho_i > 0$ unless we know that the sample will yield 0."

Situation: *spherical light source*.

Starting point: *selecting N=10 points on the sphere, uniformly, so* $\rho_i = 1/N = 0.1$.

- Now, suppose we skip points on one hemisphere.
- Skipping points means $\rho_i = 0$.
- To ensure that $\sum \rho_i = 1$: double ρ_i for remaining points, to 0.2.
- Before: $\mathrm{E} = \frac{1}{N}\sum_{i=1}^{N} \frac{V_i=0.5}{\rho_i=0.1}$, after: $\frac{1}{N}\sum_{i=1}^{N} \frac{V_i=1}{\rho_i=0.2}$



Similar situation: when evaluating the Lambertian BRDF, we skip the hemisphere below the surface. We account for this by reducing the domain to 2π.

Lights

Sampling Occluded Lights

Situation 1:

We have no information about occlusion.

NEE probes each light with 50% probability.

- samples are scaled up by 1/50%=2;
- rays to light 2 always yields 0;

➔ **point *p* receives energy from light 1 in 50% of the cases, but the light is multiplied by 2.**

Situation 2:

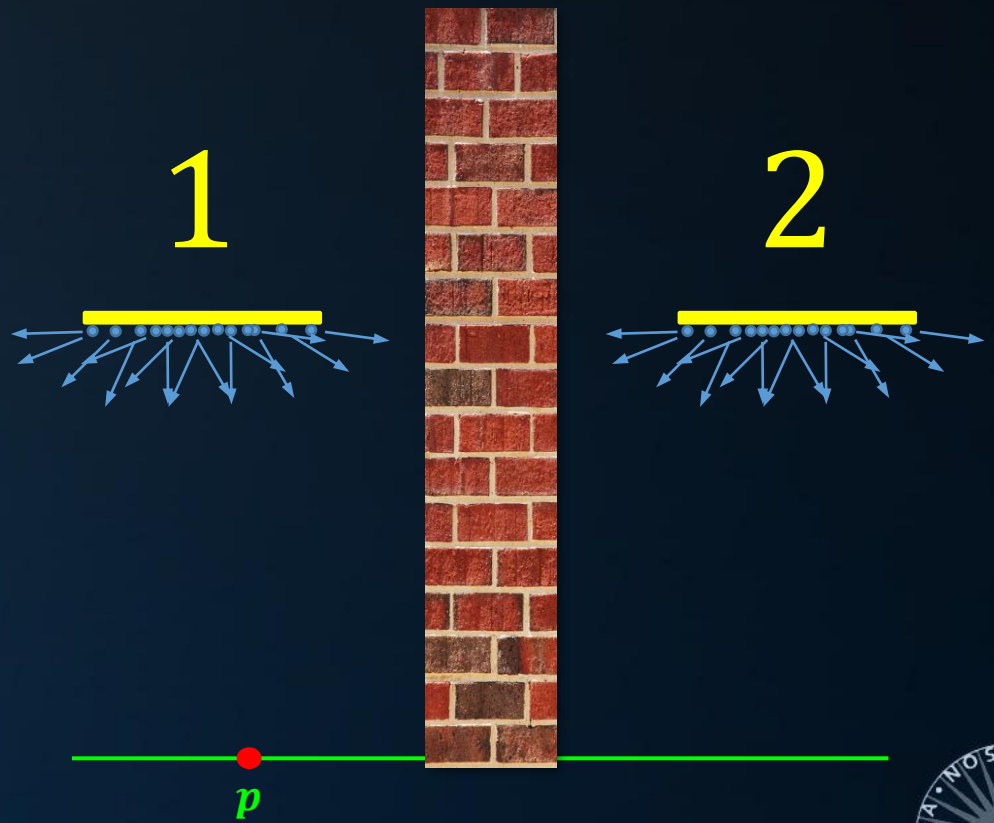
We know light 2 is occluded.

NEE probes light 1 with 100% probability.

- samples are scaled by 1;

➔ **point *p* receives energy from light 1 in 100% of the cases, multiplier is 1.**

The only difference between situation 1 and 2 is variance: in situation 1, we get twice the energy each time we sample light 1, but it gets sampled in only 50% of the cases.
In situation 2, we get a much more even amount of energy for each sample.



Today's Agenda: *Monte Carlo*

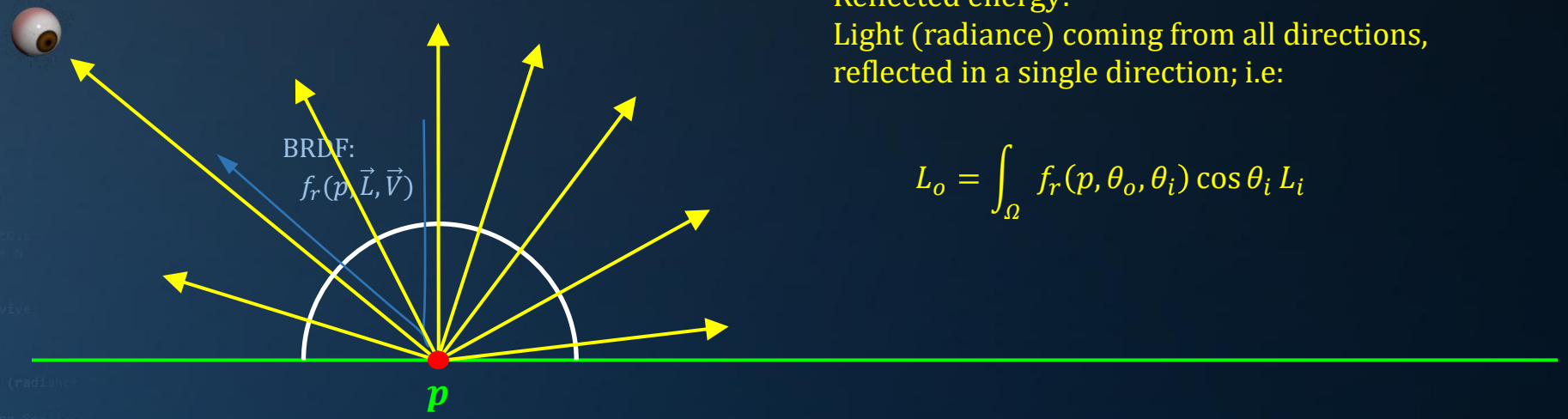
- Sampling an Area Light with One Ray
- Sampling Multiple Area Lights with One Ray
- Difficult Cases: Spherical Lights, Occluded Lights
- The Random Walk
- Random Walk with Next Event Estimation
- Digest
- P3 Topics

Walk

The Random Walk

How much light gets transported to the eye?

1. The light that p emits towards the eye (typically: nothing); plus:
2. The light that p reflects towards the eye.



Reflected energy:
Light (radiance) coming from all directions, reflected in a single direction; i.e:

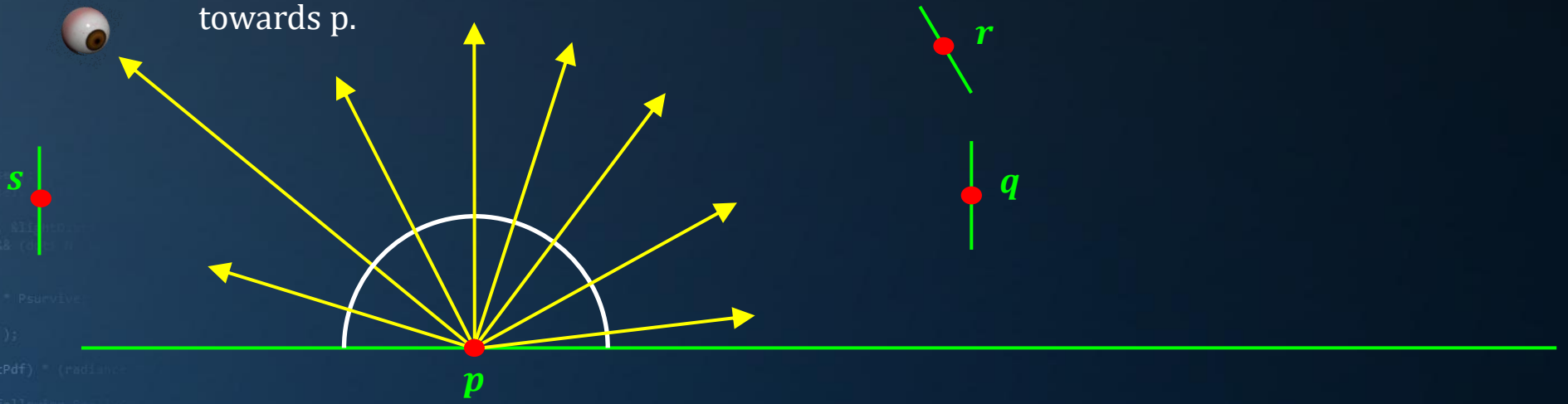
$$L_o = \int_\Omega f_r(p, \theta_o, \theta_i) \cos\theta_i \, L_i$$

Walk

The Random Walk

How much light gets transported to the eye?

1. The light that p emits towards the eye (typically: nothing);
2. The light that p reflects towards the eye:
   a) That is: the light that q, r, s emit towards p, plus
   b) The light that q, r, s (and all other scene surface points) reflect towards p.



Regarding 2b:

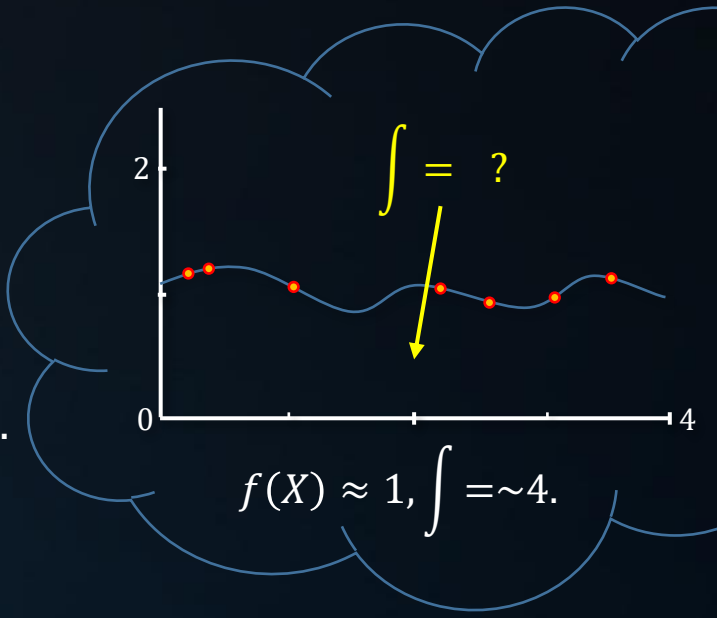
- The further away a point, the lower the probability that a random ray from p strikes it.
- The probability is also proportional to $\vec{N}_{s,p,q} \cdot -\vec{L}$.
- At p, we scale by $\vec{N}_p \cdot \vec{L}$ to compensate for the fact that we sample radiance, while in fact we need irradiance for the BRDF.

Walk

Sampling the Hemisphere using a Single Ray

The light being reflected towards the eye is the light arriving from all directions over the hemisphere, scaled by the BRDF: $\int_\Omega f_r(p, \theta_o, \theta_i)\, E_i$.

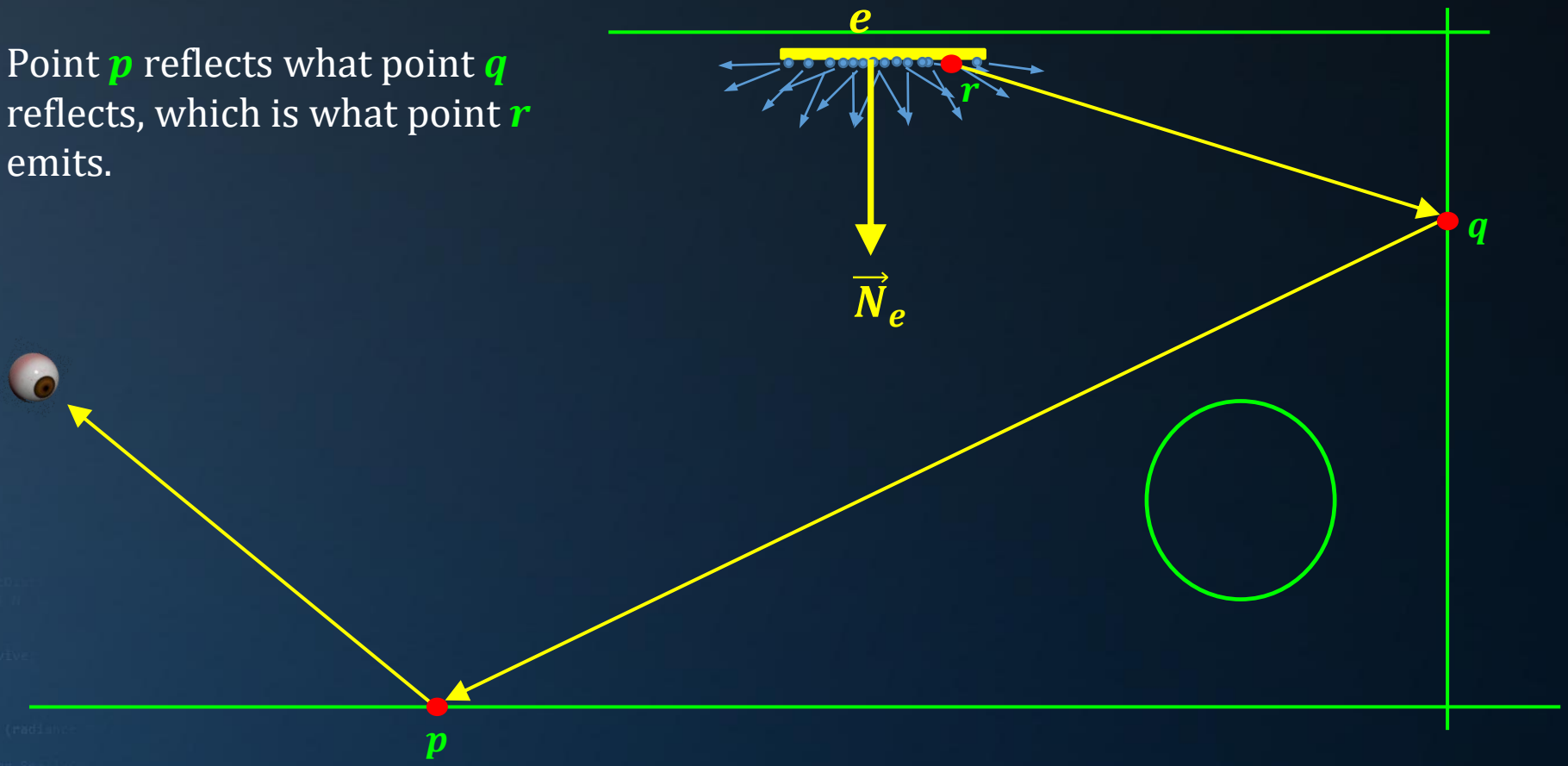
Sampling the integral using a single random ray: *Scale up by* 2π.



Walk

Random Walk

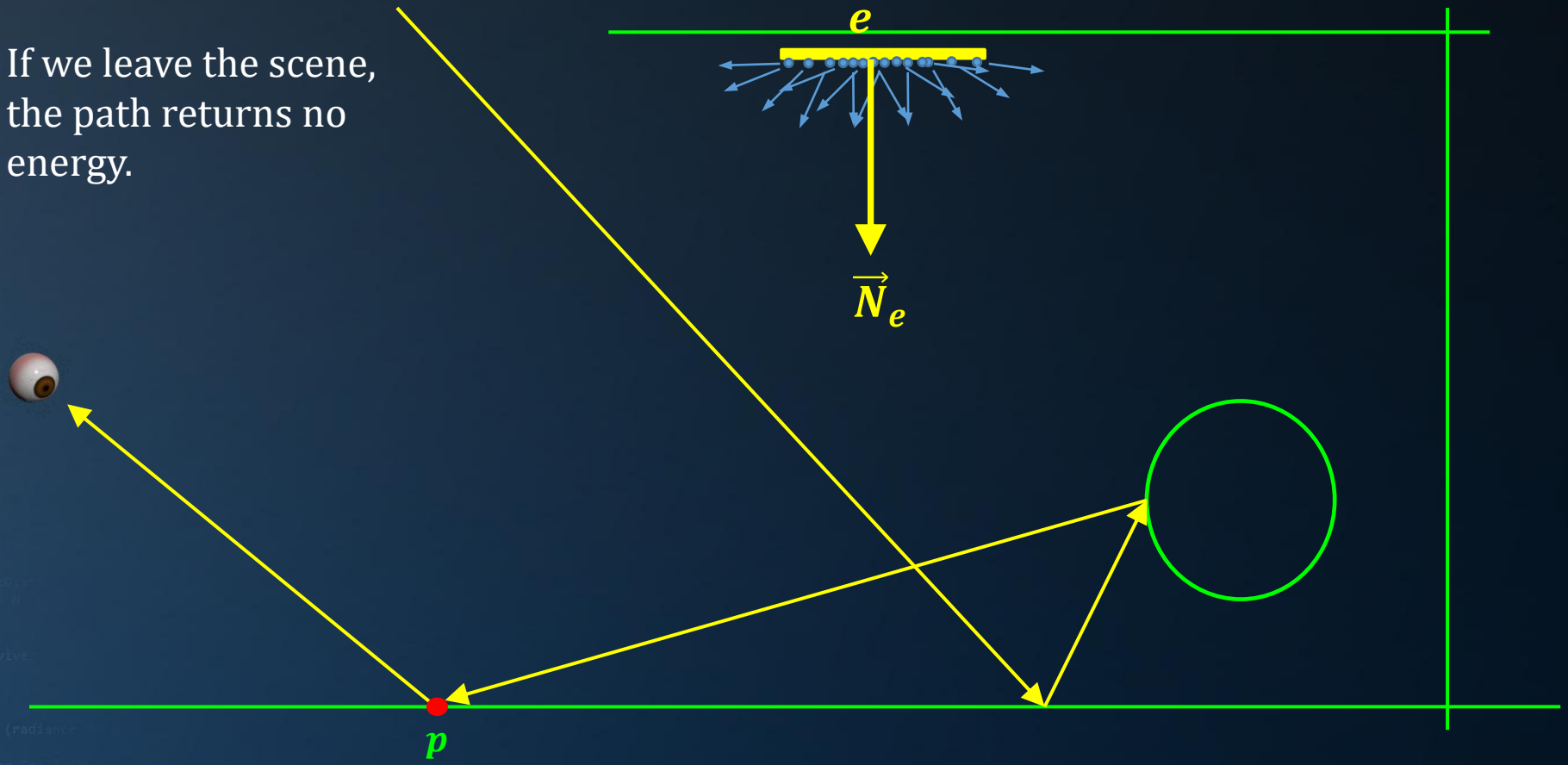
Point p reflects what point q reflects, which is what point r emits.



Walk

Random Walk

If we leave the scene, the path returns no energy.



Today's Agenda: *Monte Carlo*

- Sampling an Area Light with One Ray
- Sampling Multiple Area Lights with One Ray
- Difficult Cases: Spherical Lights, Occluded Lights
- The Random Walk
- Random Walk with Next Event Estimation
- Digest
- P3 Topics

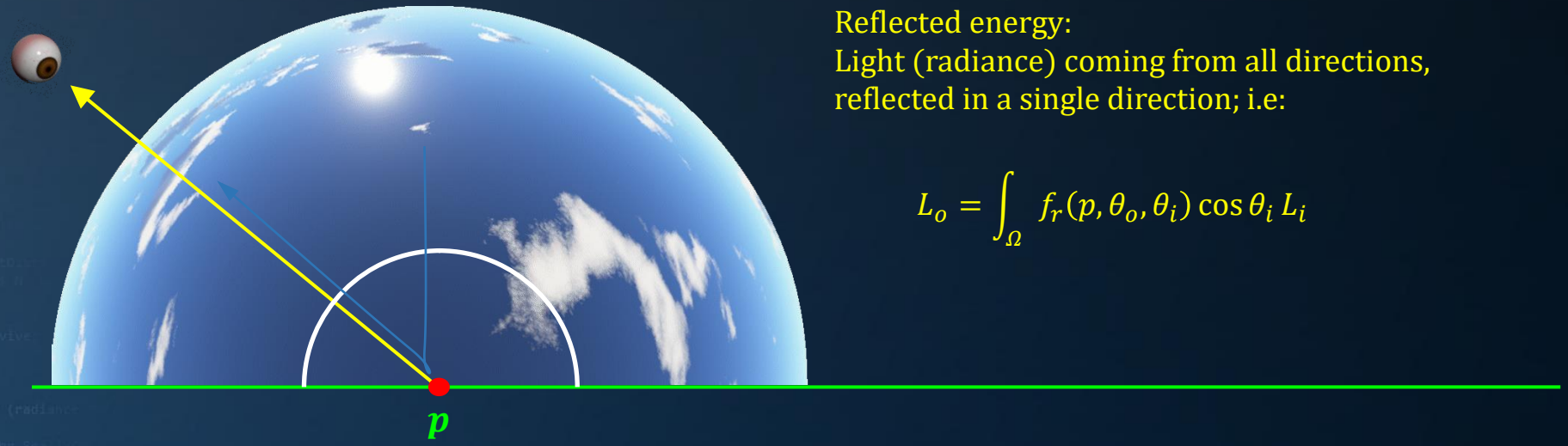
NEE

$$E(f(X)) \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(X)}{p(X)}$$

Importance-Sampling the Hemisphere

How much light gets transported to the eye?

1. The light that p emits towards the eye (typically: nothing); plus:
2. The light that p reflects towards the eye.
3. In practice: we have no idea. But we can guess.



Reflected energy:
Light (radiance) coming from all directions, reflected in a single direction; i.e:

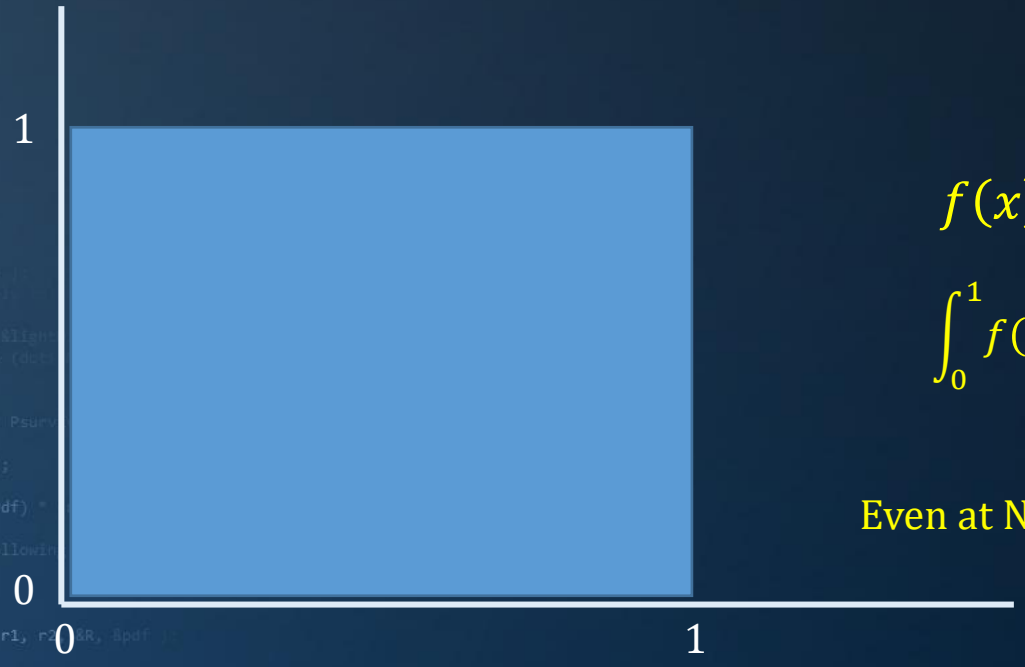
$$L_o = \int_{\Omega} f_r(p, \theta_o, \theta_i) \cos\theta_i \, L_i$$

NEE

By the way…

There exists a class of estimators known as zero variance estimators.

Example 1:



$$f(x) = 1, x \in [0,1].$$

$$\int_0^1 f(x) = 1 \quad = E\big(f(X)\big) \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(X)}{p(X)}$$

Even at N=1 we reach the expected value.

NEE

By the way…

There exists a class of estimators known as zero variance estimators.

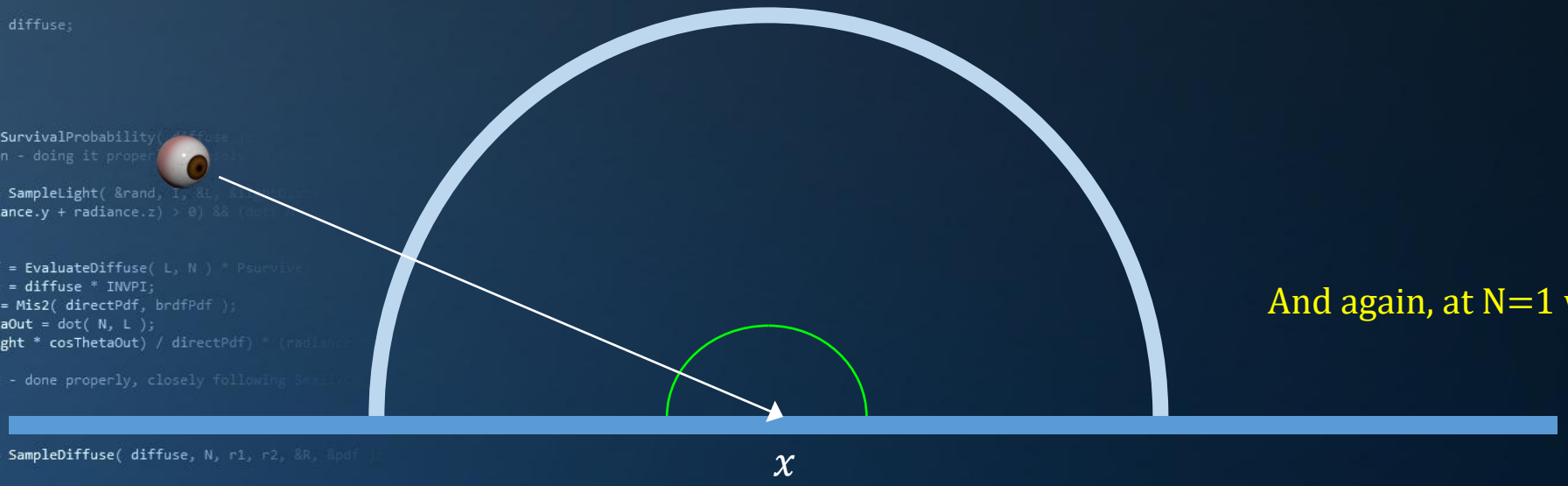
Example 2:

Point x is illuminated solely by a uniform blue sky.

$$E(f(X)) \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(X)}{p(X)}$$

$$f(x) = C\cos\theta$$

$$p(x) = \frac{\cos\theta}{\pi}$$



And again, at N=1 we reach the expected value.

NEE

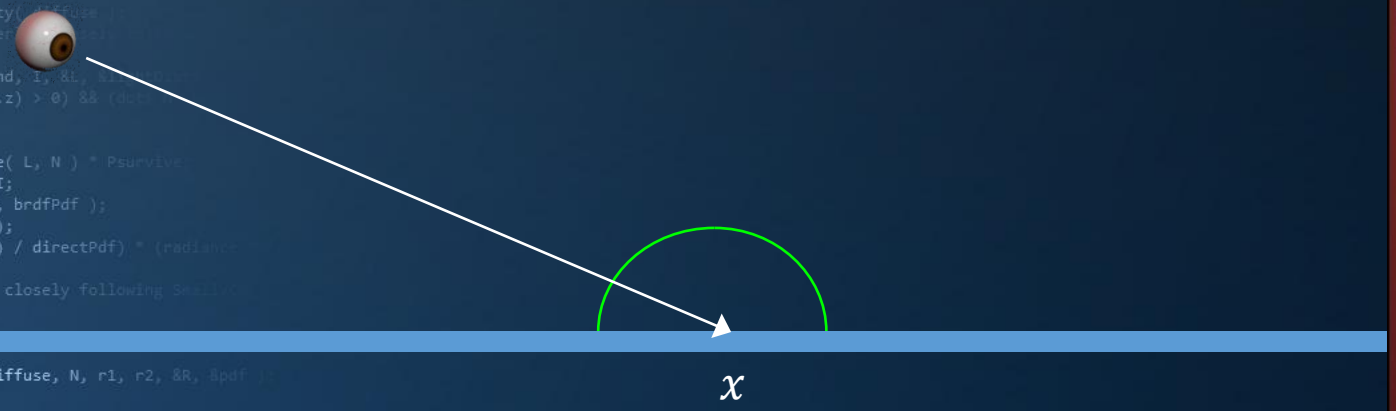
The noise on the ceiling is caused by randomly sampling the area of the light: $\cos\theta$ will have variance.

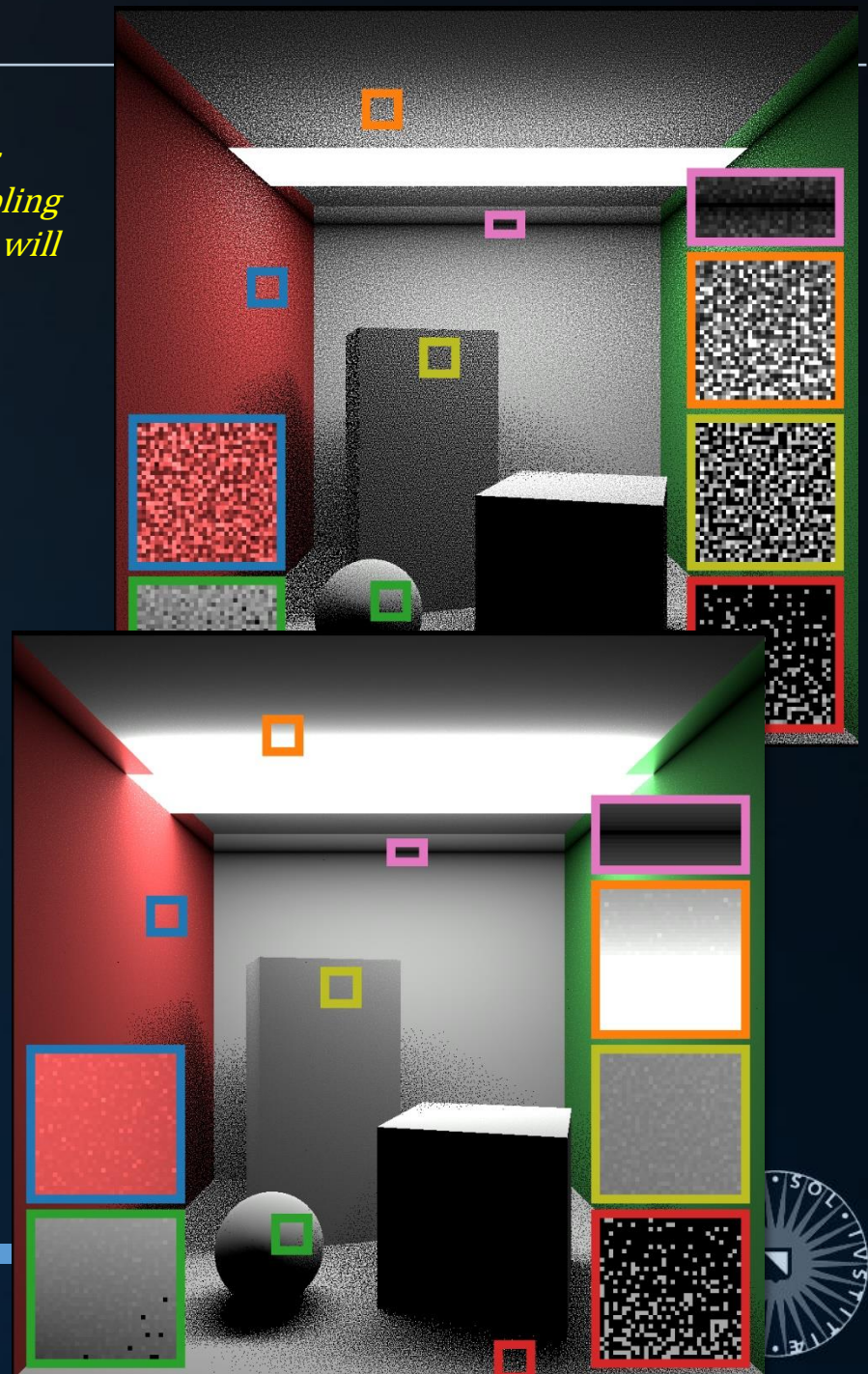
By the way…

There exists a class of estimators…

Example 3:

Point x is illuminated solely by a rectangular light source.

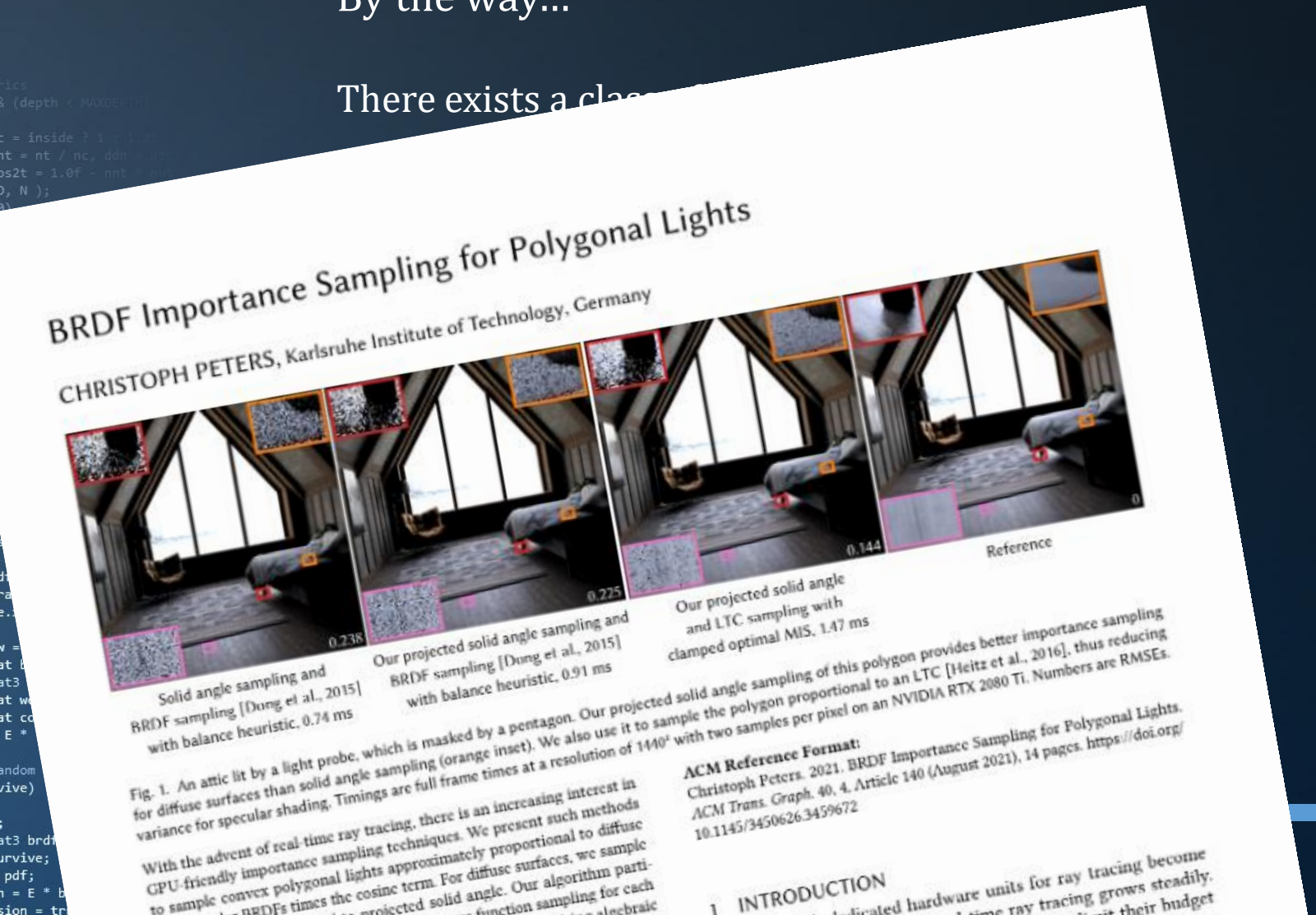




NEE

By the way…

There exists a clas



BRDF Importance Sampling for Polygonal Lights

CHRISTOPH PETERS, Karlsruhe Institute of Technology, Germany

Solid angle sampling and BRDF sampling [Dong et al., 2015] with balance heuristic, 0.74 ms

Our projected solid angle sampling and BRDF sampling [Dong et al., 2015] with balance heuristic, 0.91 ms

Our projected solid angle and LTC sampling with clamped optimal MIS, 1.47 ms

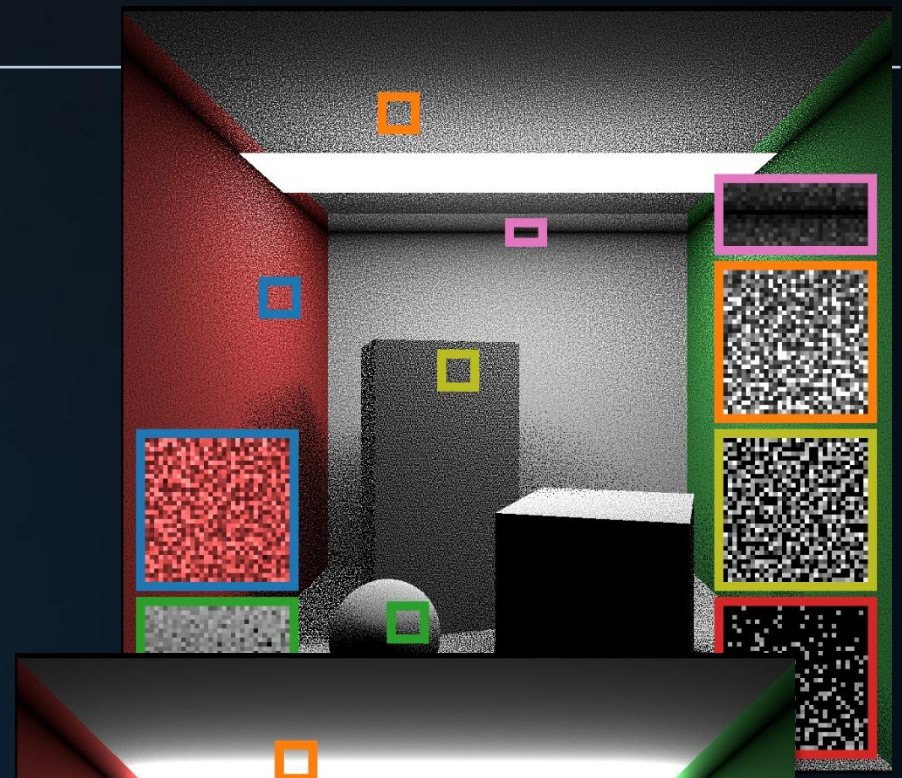
Reference

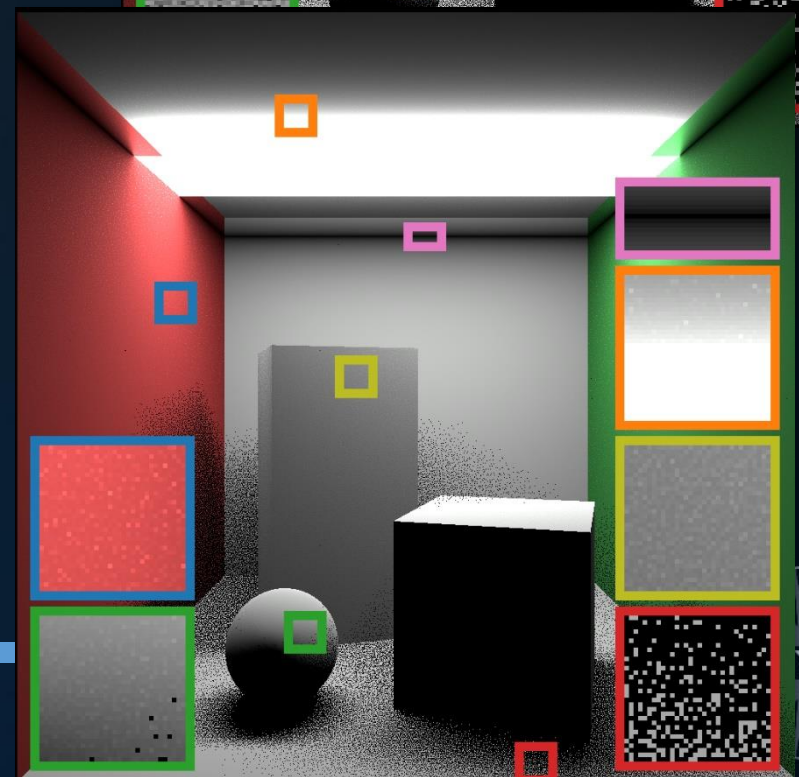
Fig. 1. An attic lit by a light probe, which is masked by a pentagon. Our projected solid angle sampling of this polygon provides better importance sampling for diffuse surfaces than solid angle sampling (orange inset). We also use it to sample the polygon proportional to an LTC [Heitz et al., 2016], thus reducing variance for specular shading. Timings are full frame times at a resolution of 1440^2 with two samples per pixel on an NVIDIA RTX 2080 Ti. Numbers are RMSEs.

With the advent of real-time ray tracing, there is an increasing interest in GPU-friendly importance sampling techniques. We present such methods to sample convex polygonal lights approximately proportional to diffuse BRDFs times the cosine term. For diffuse surfaces, we sample projected solid angle. Our algorithm parti-

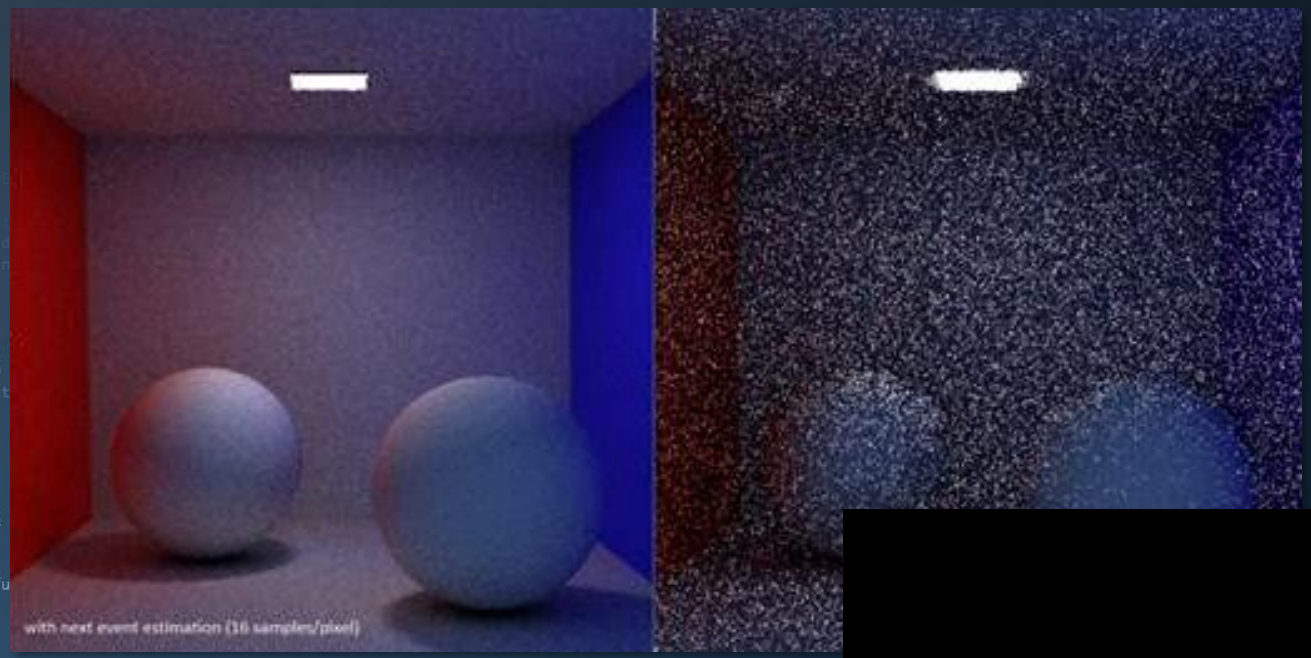
ACM Reference Format:
Christoph Peters. 2021. BRDF Importance Sampling for Polygonal Lights. *ACM Trans. Graph.* 40, 4, Article 140 (August 2021), 14 pages. https://doi.org/10.1145/3450626.3459672

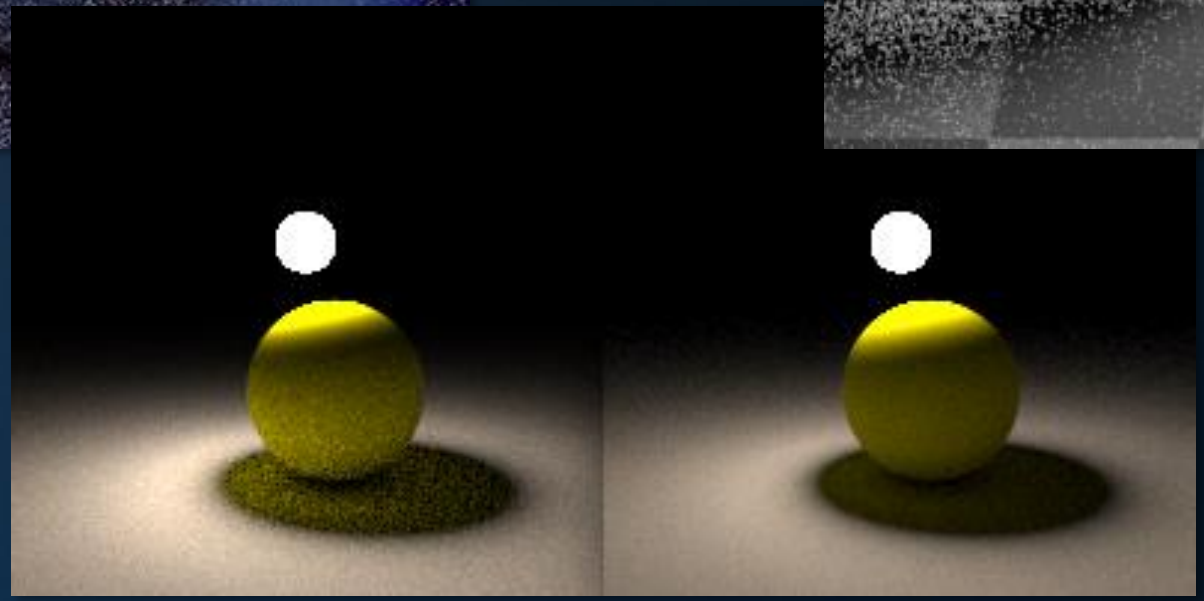
1 INTRODUCTION





NEE

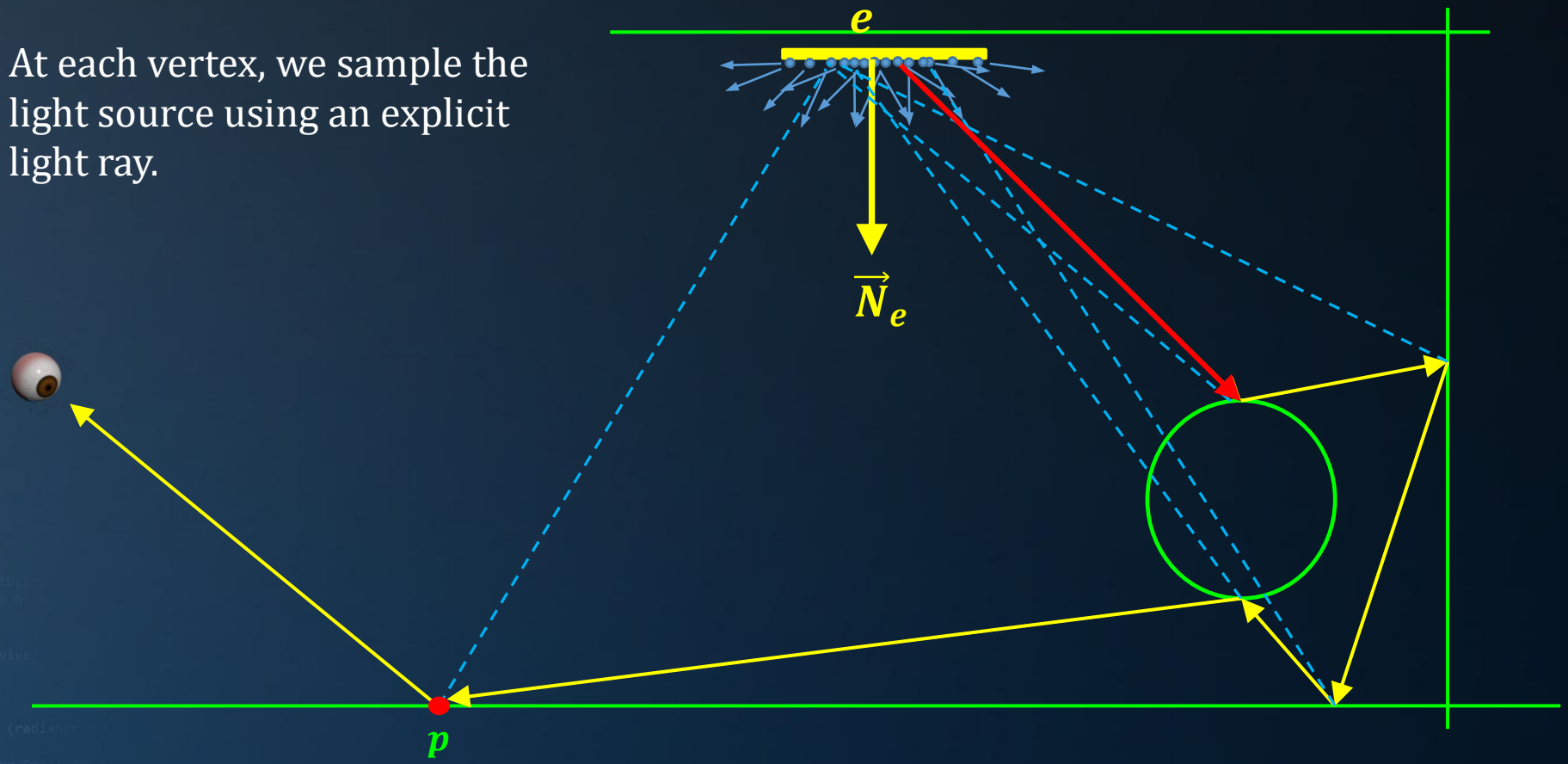




NEE

Next Event Estimation

At each vertex, we sample the light source using an explicit light ray.



NEE

Next Event Estimation

Why does this work?

“The light arriving via point p is the light reflected by point p, plus the light emitted by point p.”

And thus:

The light reflected by point p is the light arriving at p originating from light sources (1), plus the light reflected towards p (2).

1: Direct light at point p.

2: Indirect light at point p.

NEE

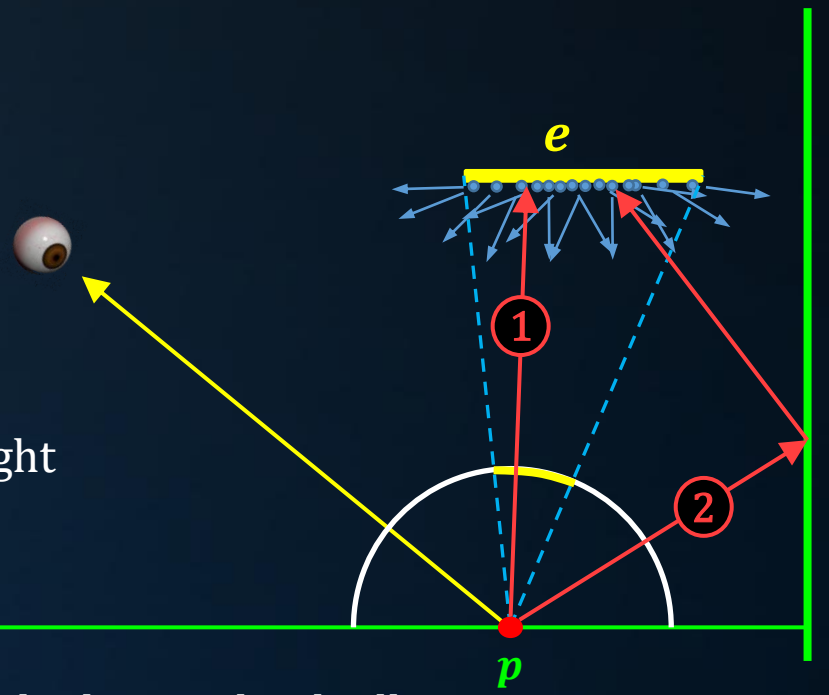
Next Event Estimation

If we send out a ray in a random direction over the hemisphere of ***p***, this ray may return two types of illumination:

1: Direct: the ray hit the light source;
2: Indirect: the ray missed the light source.

If we ignore all random rays that hit a light source (as in: terminate them, return 0), we remove the direct light arriving at ***p***.

If we sample just the lights, we remove the indirect light arriving at ***p***.



Since the contributions show no overlap, we can sample them individually, using two rays, and sum the result.

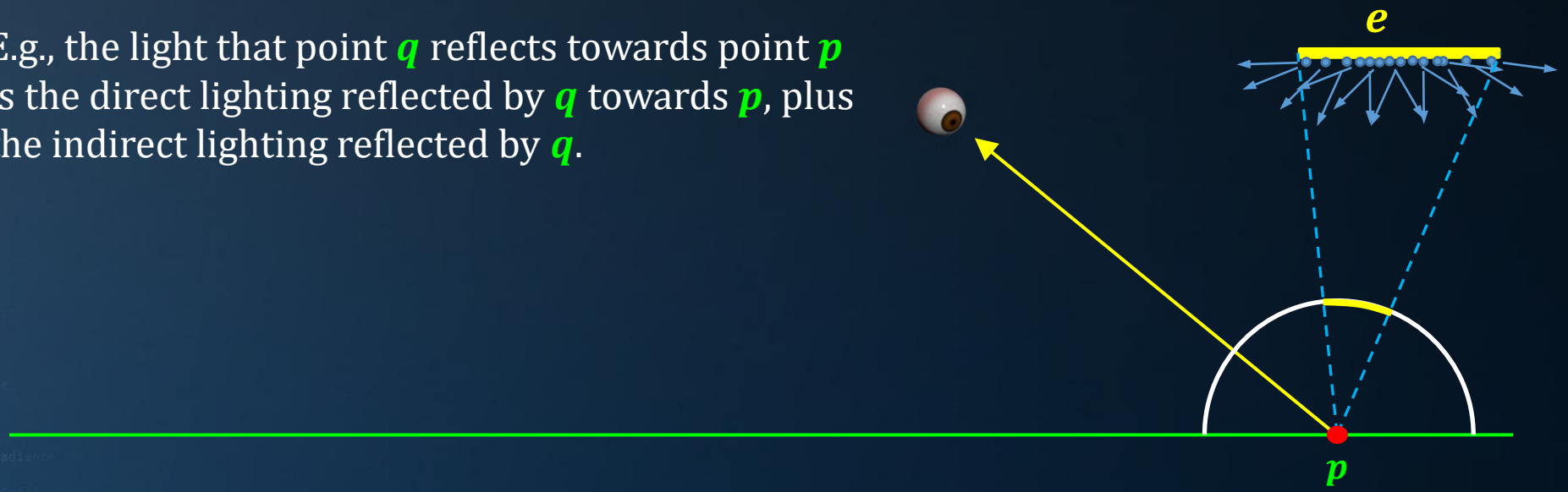
NEE

Next Event Estimation

Direct and indirect illumination can be sampled separately, *as long as we guarantee that there will not be overlap*.

This works for any point, not just the primary hit:

E.g., the light that point ***q*** reflects towards point ***p*** is the direct lighting reflected by ***q*** towards ***p***, plus the indirect lighting reflected by ***q***.



NEE

Next Event Estimation

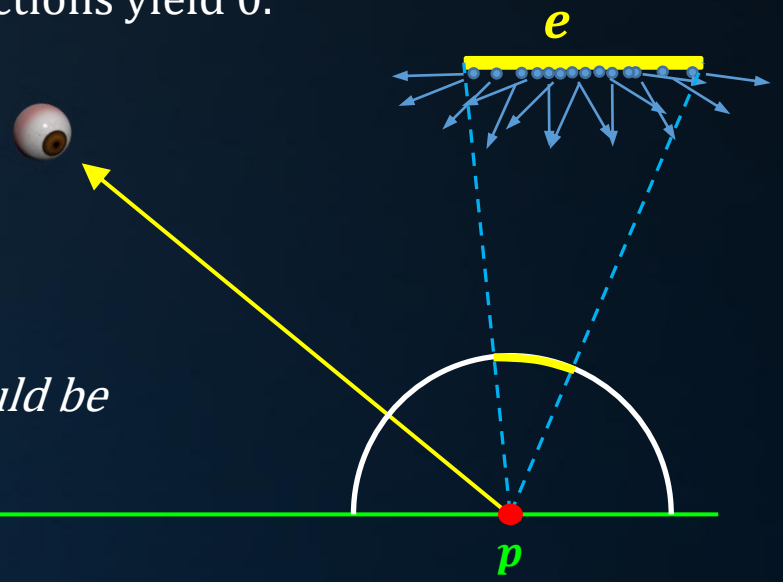
1. Why don't we use next event estimation for a specular surface?

Explicit light sampling still requires evaluation of the BRDF. For a specular surface, the BRDF for θ_o is ∞ for a single θ_i, which is why we continue the (not so) random walk in that direction. All other directions yield 0.

Consequence:

Since we do not send out an explicit light ray in this case, the random walk may now return direct illumination: there is no overlap.

In fact, if we didn't accept direct illumination, we would be missing energy.



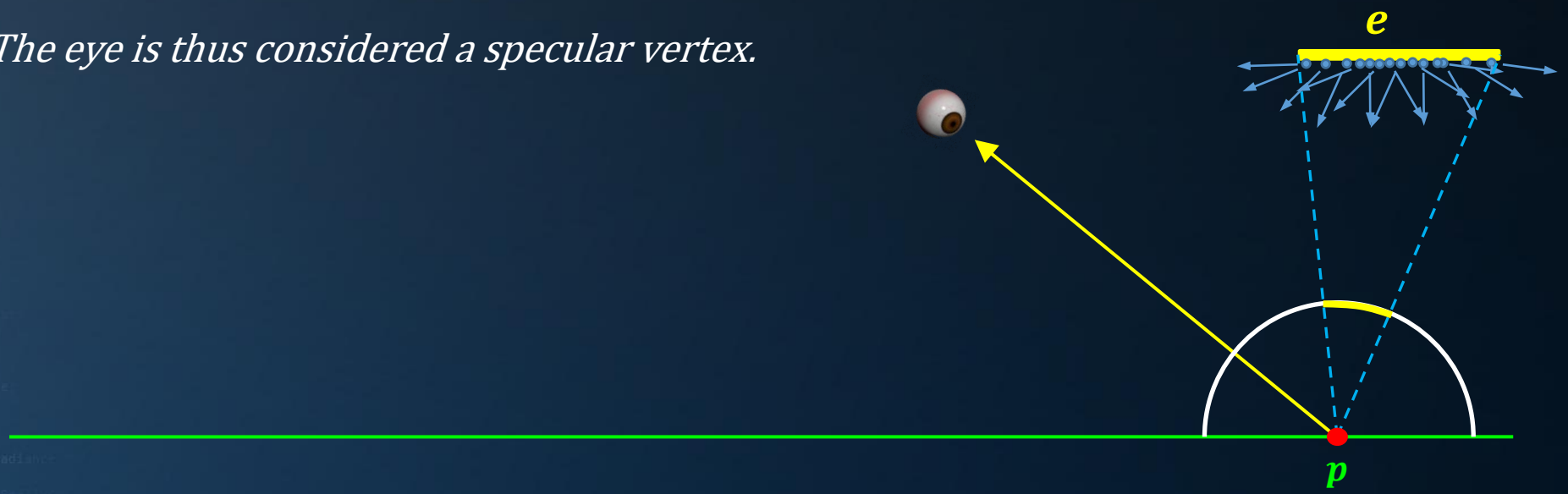
NEE

Next Event Estimation

2. Why should we return direct illumination for the primary ray?

The eye vertex did not send out an explicit light ray. Since direct illumination is not sampled separately, the random walk may return this illumination.

The eye is thus considered a specular vertex.



NEE

Next Event Estimation

Also think about it like this:

The eye looks directly at a light source.

If we terminate those paths, the light will look black.

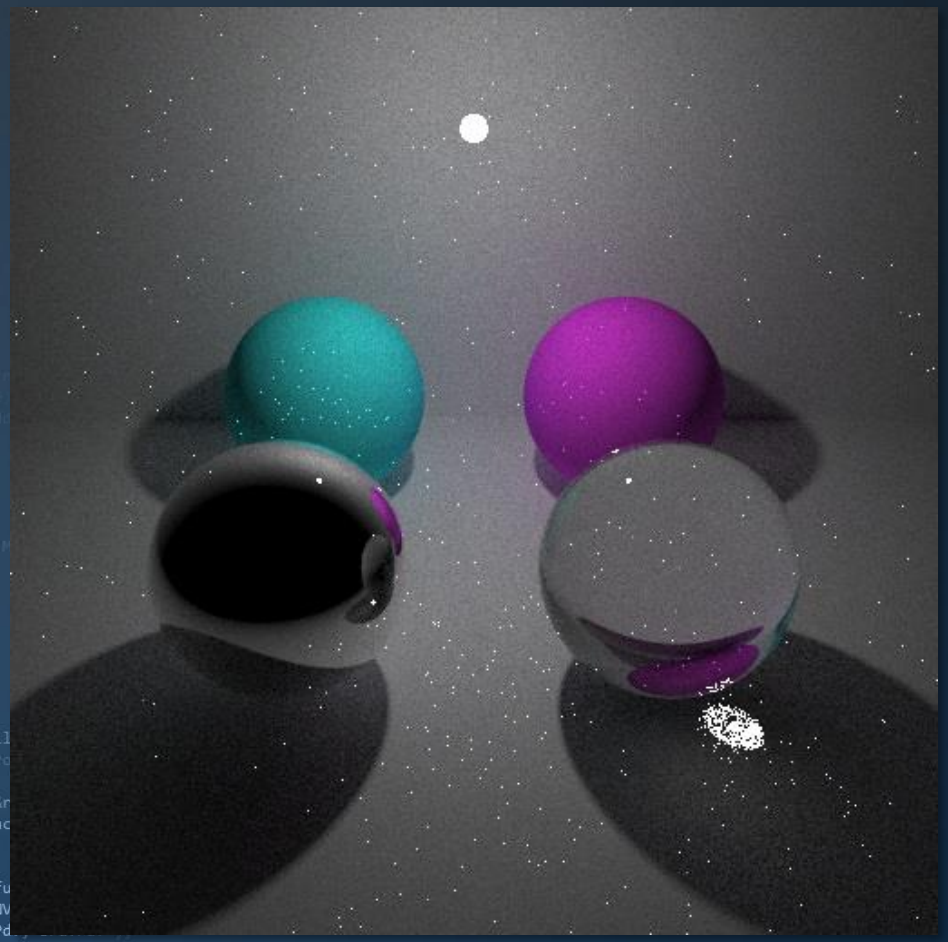
The eye looks at a light in a mirror.

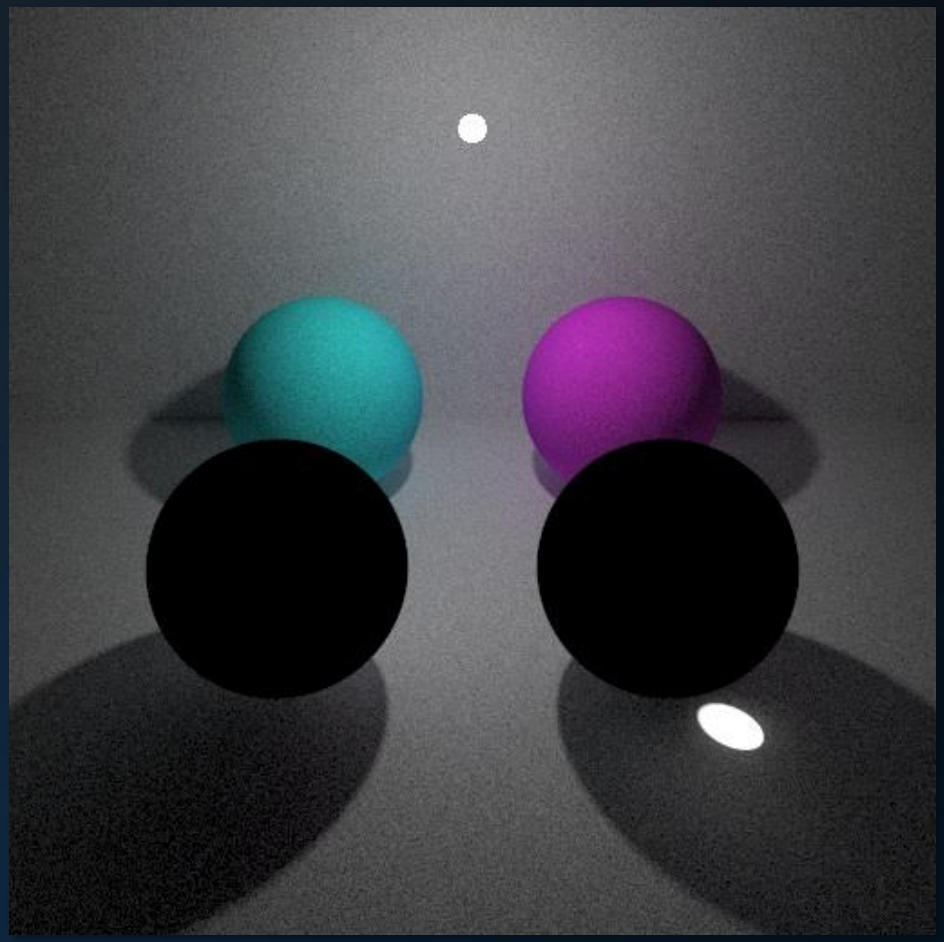
If we terminate those paths, we see a black light in the mirror.

In all other cases, we send out an explicit light ray. To compensate for that extra ray:

- The extra ray may only sample direct illumination. It doesn't bounce.
- Any other way of sampling direct illumination is blocked.

NEE





http://sjbrown.co.uk/2011/01/03/two-way-path-tracing

Today's Agenda: *Monte Carlo*

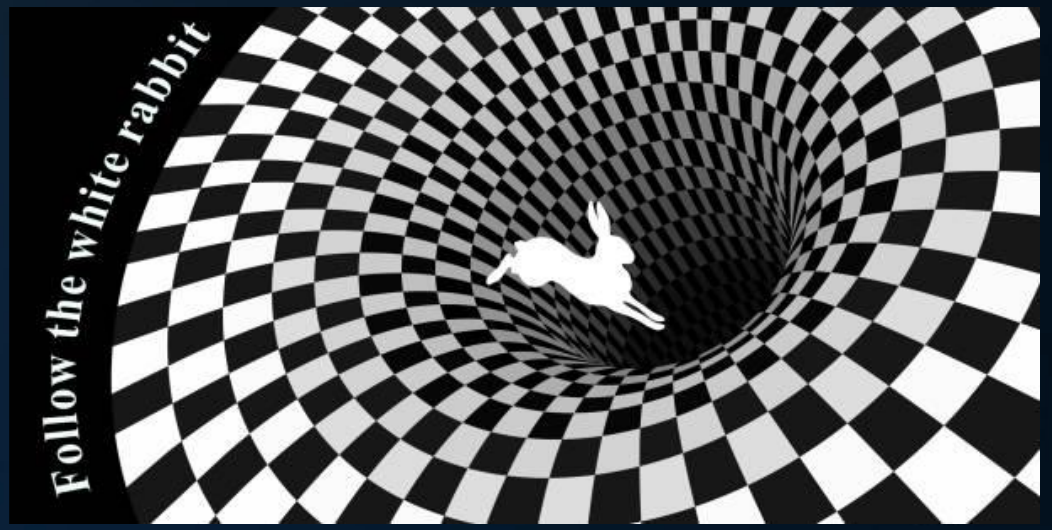
- Sampling an Area Light with One Ray
- Sampling Multiple Area Lights with One Ray
- Difficult Cases: Spherical Lights, Occluded Lights
- The Random Walk
- Random Walk with Next Event Estimation
- Digest
- P3 Topics

Efficiency

Efficiency in Monte-Carlo:

We must use importance sampling.

- We are not tracing photons backwards, we are establishing importance.
- A path that cannot be importance sampled is a noisy path.
- We can still trace from the light to the camera (‘forward path tracing’ aka light tracing).
- Consider using Multiple Importance Sampling.
- *(it works for NEE, but also when combining forward and backward path tracing)*
- Any pdf is valid, as long as … and …. .
- We can play with this stuff.
- We can *learn* importance.



Today's Agenda: *Monte Carlo*

- Sampling an Area Light with One Ray
- Sampling Multiple Area Lights with One Ray
- Difficult Cases: Spherical Lights, Occluded Lights
- The Random Walk
- Random Walk with Next Event Estimation
- Digest
- P3 Topics

Future Work

Materials

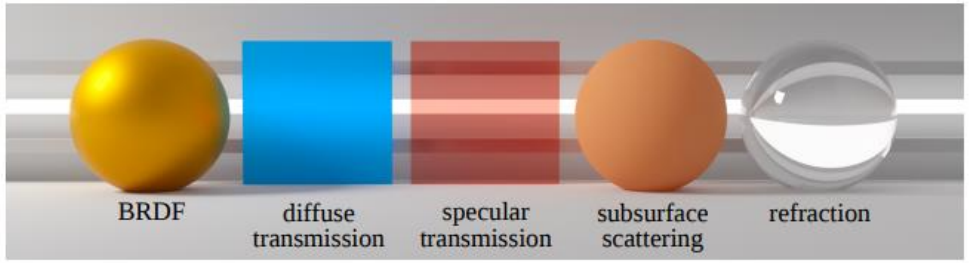


Figure 1: Rendered examples using our new BSDF.

Extending the Disney BRDF to a BSDF with Integrated Subsurface Scattering

by Brent Burley, Walt Disney Animation Studios

1 Introduction

We introduced a new physically based shading model on Wreck-It Ralph [Bur12], and we used this single, general-purpose BRDF on all materials (except hair). This model, which has come to be known as the Disney BRDF, is able to reproduce a wide range of materials with only a few parameters. For our next film, Frozen, we continued to use this BRDF unmodified, but effects like refraction and subsurface scattering were computed separately from the BRDF, and indirect illumination was approximated using point clouds. All of these effects were combined through ad hoc shading in an additive way.

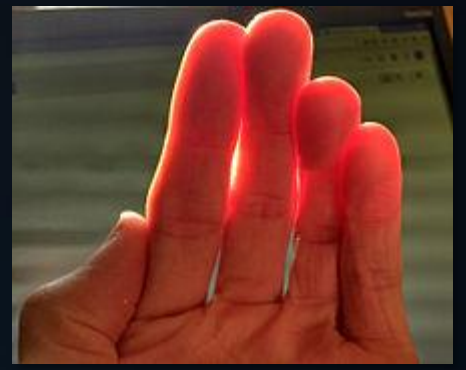
Starting with Big Hero 6 in 2014, we switched from ad hoc lighting and shading to path-traced

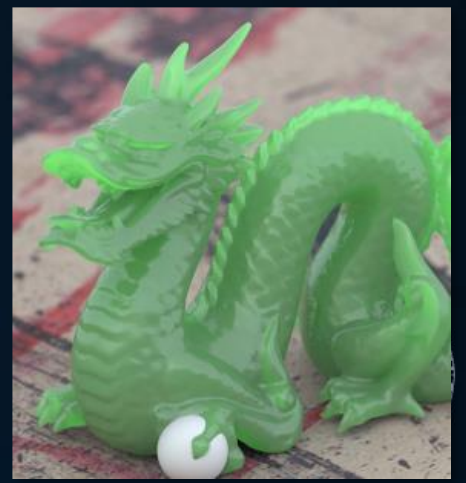
Other resources:

Implementing the Disney BSDF
schuttejoe.github.io/post/disneybsdf

Efficient Rendering of Layered Materials using an Atomic Decomposition with Statistical Operators. Belcour, 2018.

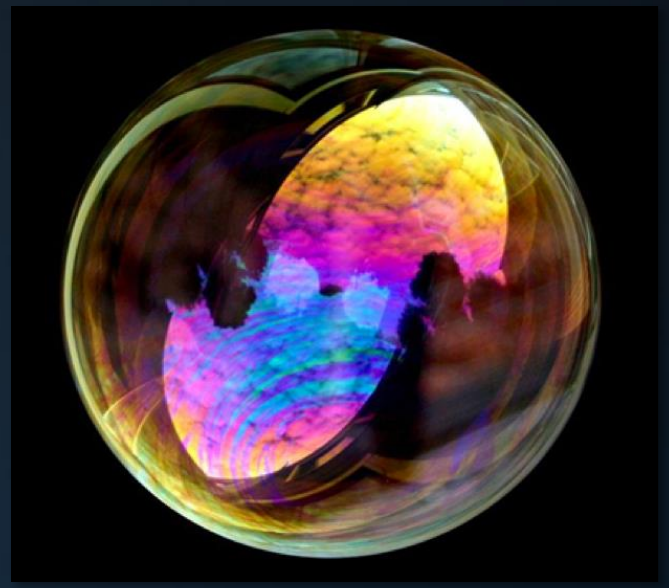
Eric Heitz's Research Page:
eheitzresearch.wordpress.com/research
(all his work is good)





Future Work

Materials

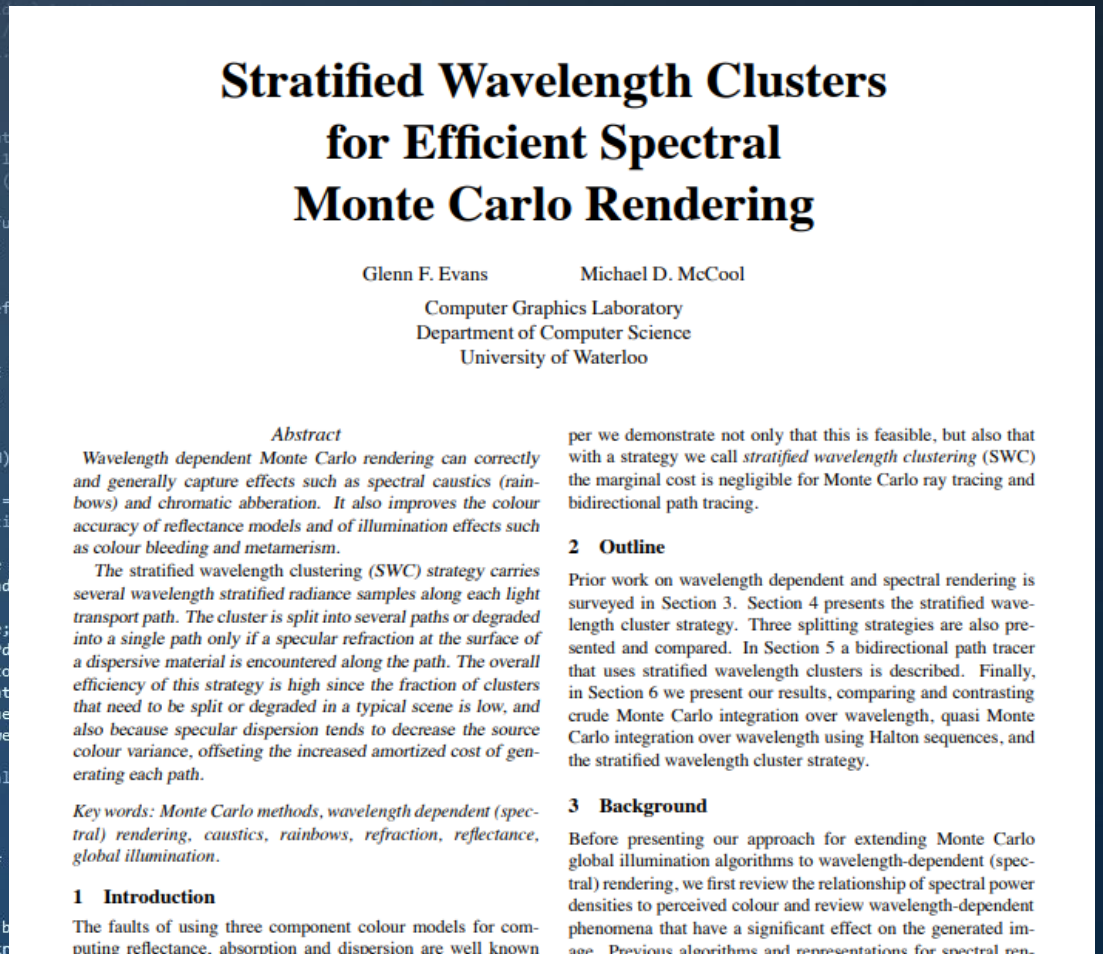


“thin film”

“consistent normal mapping”

Future Work

Spectral Rendering



Stratified Wavelength Clusters for Efficient Spectral Monte Carlo Rendering

Glenn F. Evans Michael D. McCool

Computer Graphics Laboratory
Department of Computer Science
University of Waterloo

Abstract

Wavelength dependent Monte Carlo rendering can correctly and generally capture effects such as spectral caustics (rainbows) and chromatic abberation. It also improves the colour accuracy of reflectance models and of illumination effects such as colour bleeding and metamerism.

The stratified wavelength clustering (SWC) strategy carries several wavelength stratified radiance samples along each light transport path. The cluster is split into several paths or degraded into a single path only if a specular refraction at the surface of a dispersive material is encountered along the path. The overall efficiency of this strategy is high since the fraction of clusters that need to be split or degraded in a typical scene is low, and also because specular dispersion tends to decrease the source colour variance, offseting the increased amortized cost of generating each path.

Key words: Monte Carlo methods, wavelength dependent (spectral) rendering, caustics, rainbows, refraction, reflectance, global illumination.

1 Introduction

The faults of using three component colour models for computing reflectance, absorption and dispersion are well known per we demonstrate not only that this is feasible, but also that with a strategy we call *stratified wavelength clustering* (SWC) the marginal cost is negligible for Monte Carlo ray tracing and bidirectional path tracing.

2 Outline

Prior work on wavelength dependent and spectral rendering is surveyed in Section 3. Section 4 presents the stratified wavelength cluster strategy. Three splitting strategies are also presented and compared. In Section 5 a bidirectional path tracer that uses stratified wavelength clusters is described. Finally, in Section 6 we present our results, comparing and contrasting crude Monte Carlo integration over wavelength, quasi Monte Carlo integration over wavelength using Halton sequences, and the stratified wavelength cluster strategy.

3 Background

Before presenting our approach for extending Monte Carlo global illumination algorithms to wavelength-dependent (spectral) rendering, we first review the relationship of spectral power densities to perceived colour and review wavelength-dependent phenomena that have a significant effect on the generated image. Previous algorithms and representations for spectral ren-

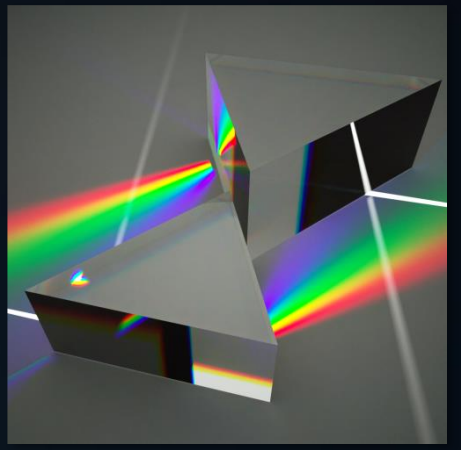
Other resources:

Hero Wavelength Spectral Sampling. Wilie et al., 2014.

Physically Meaningful Rendering using Tristimulus Colours. Meng et al., 2015.

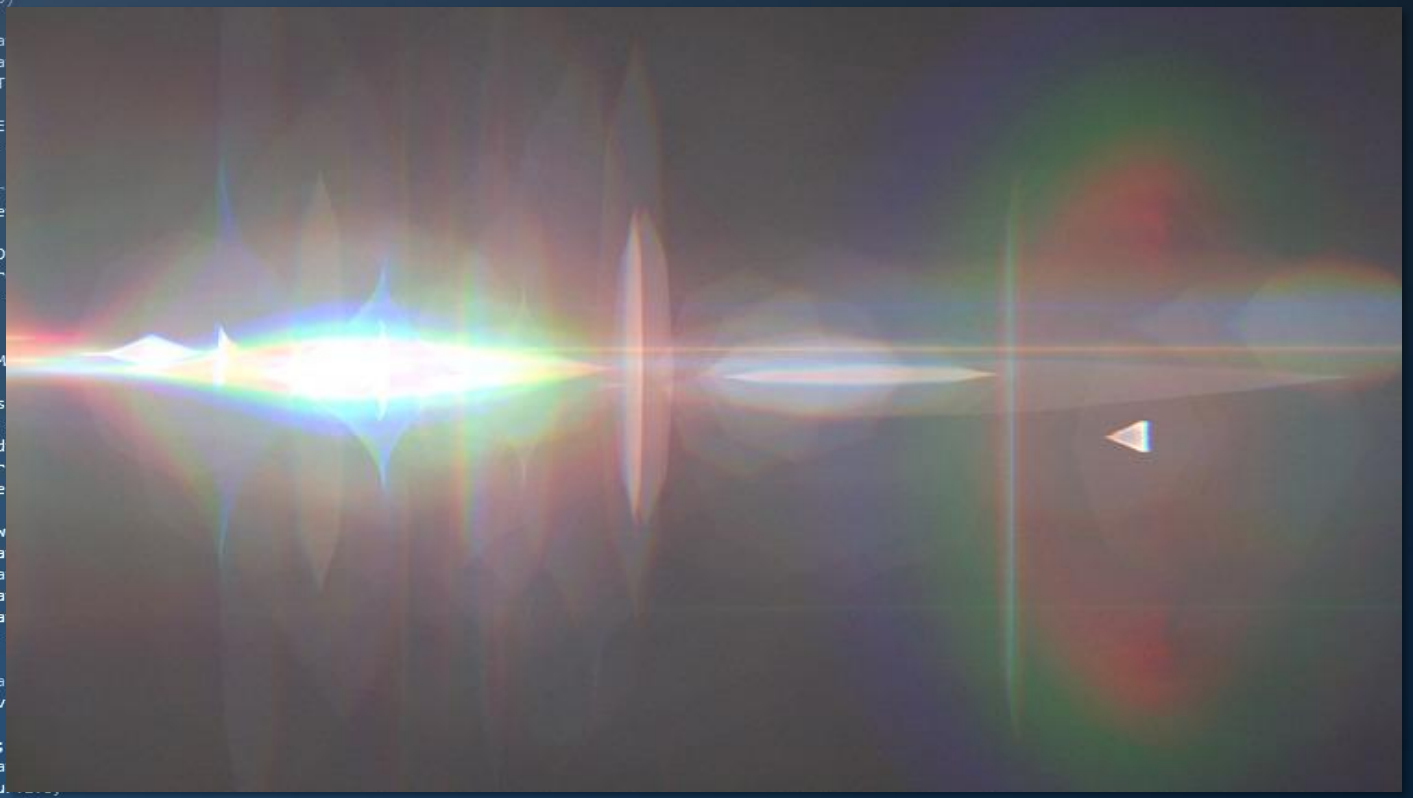
PBRT. Pharr et al., 2004-2018.
https://www.pbrt.org

Mitsuba Renderer. W. Jakob, 2014.
https://www.mitsuba-renderer.org



Future Work

Spectral Rendering



Future Work

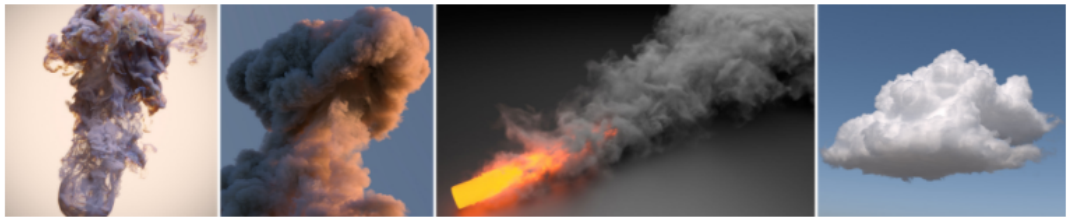
Participating Media

Monte Carlo methods for physically based volume rendering

Jan Novák[1] Iliyan Georgiev[2] Johannes Hanika[3] Jaroslav Křivánek[4] Wojciech Jarosz[5]

[1]Disney Research [2]Solid Angle [3]Karlsruhe Institute of Technology [4]Charles University, Prague [5]Dartmouth College

In *ACM SIGGRAPH Courses*, 2018



Various media rendered with Monte Carlo methods for physically based simulation of light transport in volumes. The three images on the left are courtesy of Lee Griggs.

Abstract | Downloads | Cite | Related

Abstract

We survey methods that utilize Monte Carlo (MC) integration to simulate light transport in scenes with participating media. The goal of this course is to complement a recent Eurographics 2018 state-of-the-art report providing a broad overview of most techniques developed to date, including a few methods from neutron transport, with a focus on concepts that are most relevant to CG practitioners.

The wide adoption of path-tracing algorithms in high-end realistic rendering has stimulated many diverse research initiatives aimed at efficiently rendering scenes with participating media. More computational power has enabled holistic approaches that tie volumetric effects and surface scattering together and simplify authoring workflows. Methods that were previously assumed to be incompatible have been unified to allow renderers to benefit from each method's respective strengths. Generally, investigations have shifted away from specialized solutions, e.g. for single- or multiple-scattering approximations or analytical methods, towards the more versatile Monte Carlo algorithms that are currently enjoying a widespread success in many production settings.

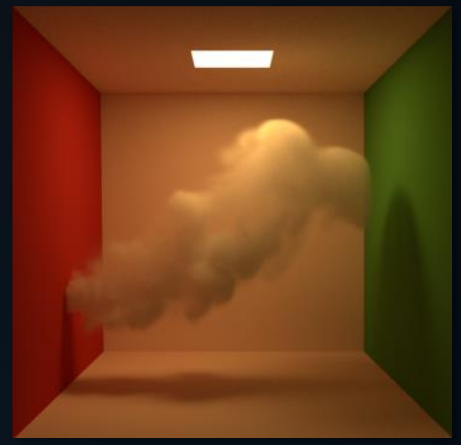
The goal of this course is to provide the audience with a deep, up-to-date understanding of key techniques for free-path sampling, transmittance estimation, and light-path construction in participating media, including those that are presently utilized in production rendering systems. We present a coherent overview of the fundamental building blocks and we contrast the various advanced methods that build on them, providing attendees with guidance for implementing existing solutions and developing new ones.

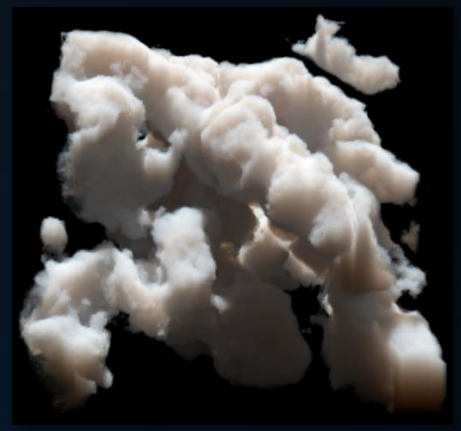
Downloads

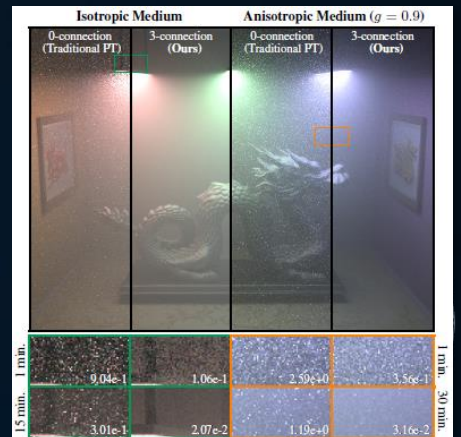
Other resources:

Importance Sampling Techniques for Path Tracing in Participating Media. Kulla and Fajardo, 2014.

Area Light Equi-Angular Sampling on ShaderToy: https://ww.shadertoy.com/view/ldXGzS

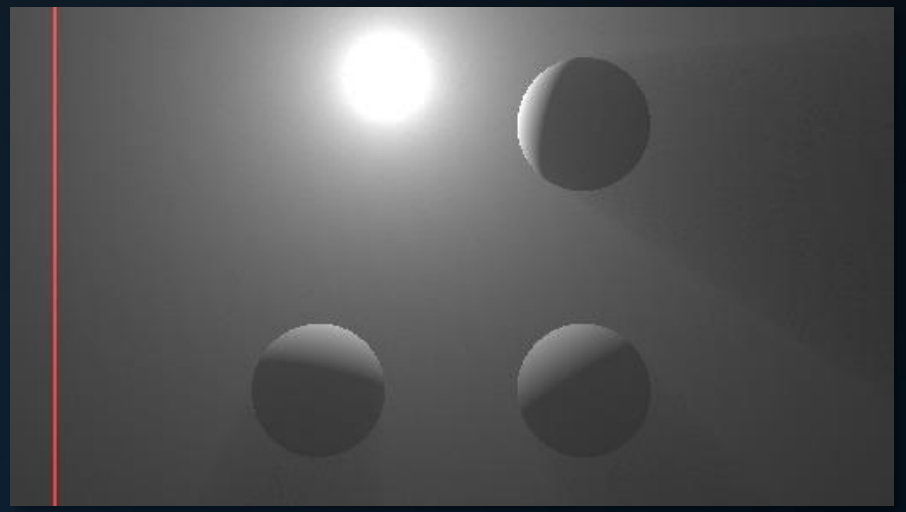






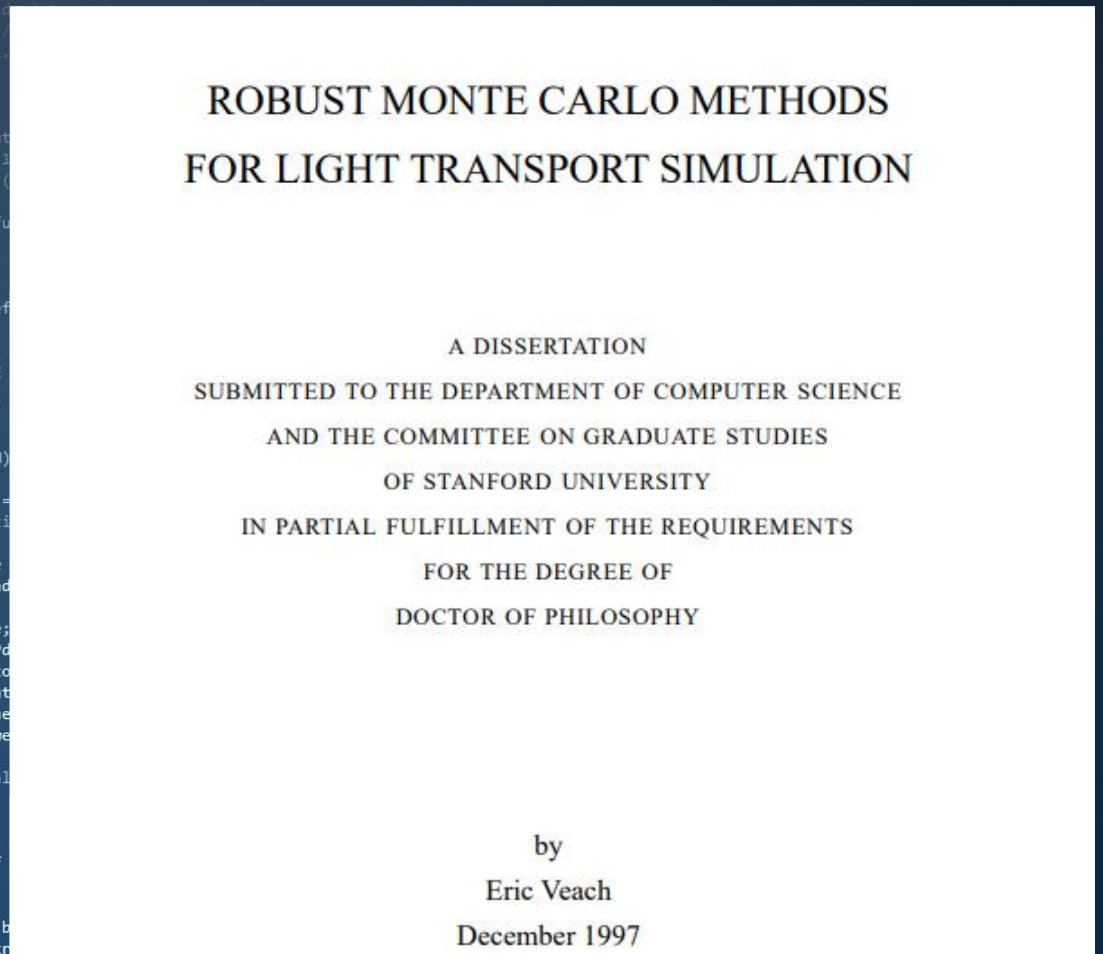
Future Work

Participating Media



Future Work

Light Transport



ROBUST MONTE CARLO METHODS
FOR LIGHT TRANSPORT SIMULATION

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

by
Eric Veach
December 1997

Other resources:

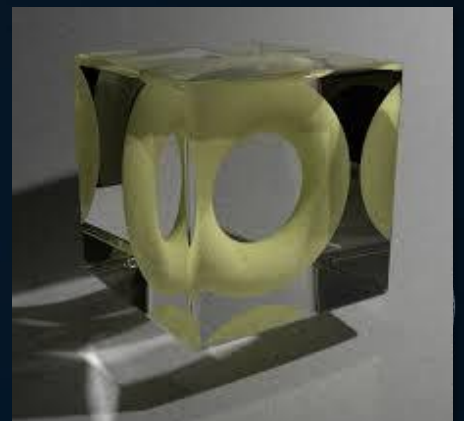
The Rendering Equation. James T. Kajiya, 1986.

Bidirectional Path Tracing. Michal Vlnas, 2018 (student paper).

Global Illumination using Photon Maps. Jensen, 1996.

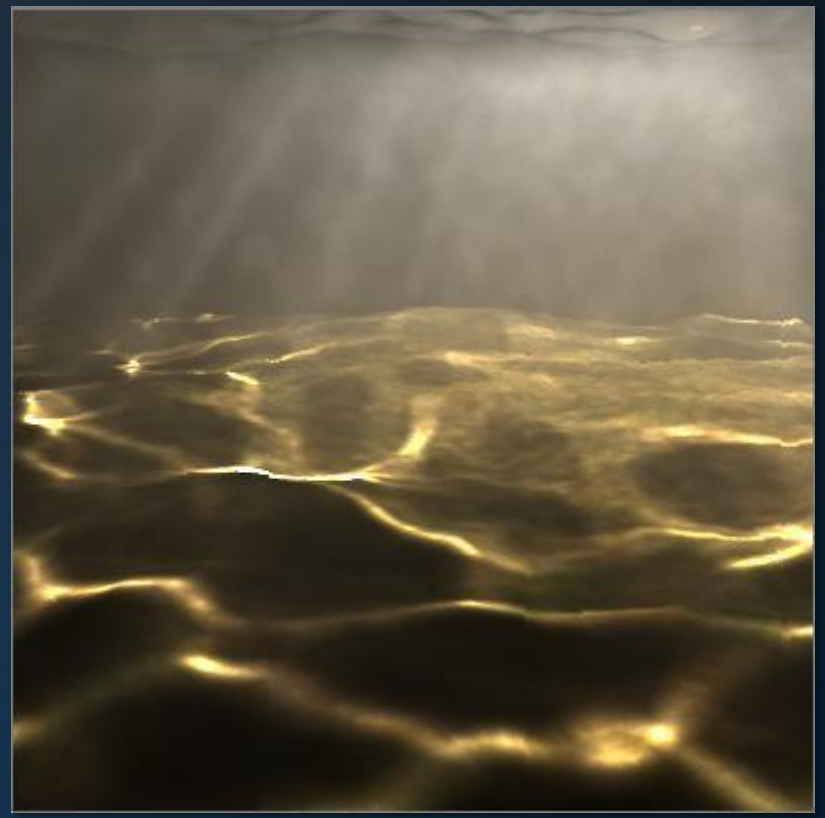
Progressive Photon Mapping. Hachisuka et al., 2008.

(article on) Vertex Connection and Merging, Georgev et al., 2012. https://schuttejoe.github.io/post/vertexconnectionandmerging



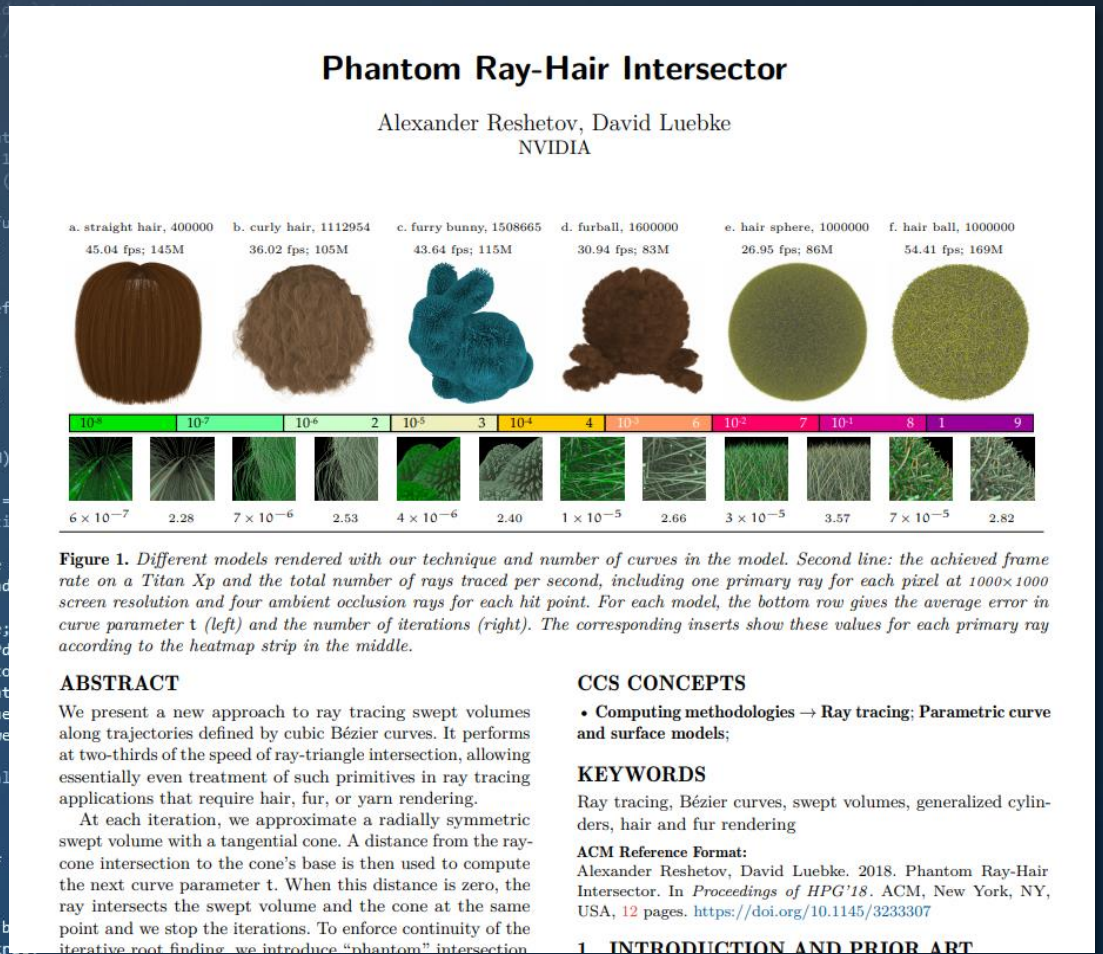
Future Work

Light Transport



Future Work

Primitives



Phantom Ray-Hair Intersector

Alexander Reshetov, David Luebke
NVIDIA

a. straight hair, 400000 — 45.04 fps; 145M
b. curly hair, 1112954 — 36.02 fps; 105M
c. furry bunny, 1508665 — 43.64 fps; 115M
d. furball, 1600000 — 30.94 fps; 83M
e. hair sphere, 1000000 — 26.95 fps; 86M
f. hair ball, 1000000 — 54.41 fps; 169M

10^{-8} 10^{-7} 10^{-6} 2 10^{-5} 3 10^{-4} 4 10^{-3} 6 10^{-2} 7 10^{-1} 8 1 9

6×10^{-7} 2.28 7×10^{-6} 2.53 4×10^{-6} 2.40 1×10^{-5} 2.66 3×10^{-5} 3.57 7×10^{-5} 2.82

Figure 1. *Different models rendered with our technique and number of curves in the model. Second line: the achieved frame rate on a Titan Xp and the total number of rays traced per second, including one primary ray for each pixel at 1000×1000 screen resolution and four ambient occlusion rays for each hit point. For each model, the bottom row gives the average error in curve parameter* t *(left) and the number of iterations (right). The corresponding inserts show these values for each primary ray according to the heatmap strip in the middle.*

ABSTRACT

We present a new approach to ray tracing swept volumes along trajectories defined by cubic Bézier curves. It performs at two-thirds of the speed of ray-triangle intersection, allowing essentially even treatment of such primitives in ray tracing applications that require hair, fur, or yarn rendering.

At each iteration, we approximate a radially symmetric swept volume with a tangential cone. A distance from the ray-cone intersection to the cone's base is then used to compute the next curve parameter t. When this distance is zero, the ray intersects the swept volume and the cone at the same point and we stop the iterations. To enforce continuity of the iterative root finding, we introduce "phantom" intersection

CCS CONCEPTS

• **Computing methodologies** → **Ray tracing**; **Parametric curve and surface models**;

KEYWORDS

Ray tracing, Bézier curves, swept volumes, generalized cylinders, hair and fur rendering

ACM Reference Format:
Alexander Reshetov, David Luebke. 2018. Phantom Ray-Hair Intersector. In *Proceedings of HPG'18*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3233307

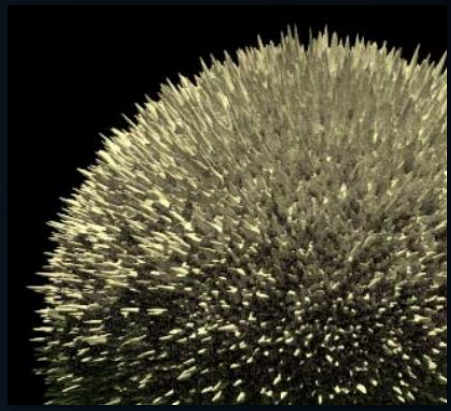
1 INTRODUCTION AND PRIOR ART

Other resources:

Direct Ray Tracing of Smoothed and Displacement Mapped Triangles. Smits et al., 2000.

Direct Ray Tracing of Phong Tessellation. Ogaki and Tokuyoshi, 2011.

Two-Level Ray Tracing with Reordering for Highly Complex Scenes. Hanika et al., 2010.



Future Work

Production

Arnold: A Brute-Force Production Path Tracer

ILIYAN GEORGIEV, THIAGO IZE, MIKE FARNSWORTH, RAMÓN MONTOYA-VOZMEDIANO, ALAN KING, BRECHT VAN LOMMEL, ANGEL JIMENEZ, OSCAR ANSON, SHINJI OGAKI, ERIC JOHNSTON, ADRIEN HERUBEL, DECLAN RUSSELL, FRÉDÉRIC SERVANT, and MARCOS FAJARDO, Solid Angle

Fig. 1. Arnold is a path-tracing renderer used for the production of photo-realistic and art-directed visual effects in feature films (left), commercials (middle), animated films (right), television series, music videos, game cinematics, motion graphics, and others. *Gravity ©2013 Warner Bros. Pictures, courtesy of Framestore; Racing Faces ©2016 Opel Motorsport, courtesy of The Mill; Captain Underpants ©2017 DreamWorks Animation.*

Arnold is a physically based renderer for feature-length animation and visual effects. Conceived in an era of complex multi-pass rasterization-based workflows struggling to keep up with growing demands for complexity and realism, Arnold was created to take on the challenge of making the simple and elegant approach of brute-force Monte Carlo path tracing practical for production rendering. Achieving this required building a robust piece of ray-tracing software that can ingest large amounts of geometry with detailed shading and lighting and produce images with high fidelity, while scaling well with the available memory and processing power.

Arnold's guiding principles are to expose as few controls as possible, provide rapid feedback to artists, and adapt to various production workflows. In this article, we describe its architecture with a focus on the design

ACM Reference format:
Iliyan Georgiev, Thiago Ize, Mike Farnsworth, Ramón Montoya-Vozmediano, Alan King, Brecht Van Lommel, Angel Jimenez, Oscar Anson, Shinji Ogaki, Eric Johnston, Adrien Herubel, Declan Russell, Frédéric Servant, and Marcos Fajardo. 2018. Arnold: A Brute-Force Production Path Tracer. *ACM Trans. Graph.* 37, 3, Article 32 (July 2018), 12 pages.
https://doi.org/10.1145/3182160

1 INTRODUCTION

At a purely technical level, visual realism in computer-generated imagery boils down to two areas: (1) amount of scene detail, e.g., number of geometric primitives, amount of textures, variety of ma-

Other resources:

The Iray Light Transport Simulation and Rendering System. Keller et al., 2017.

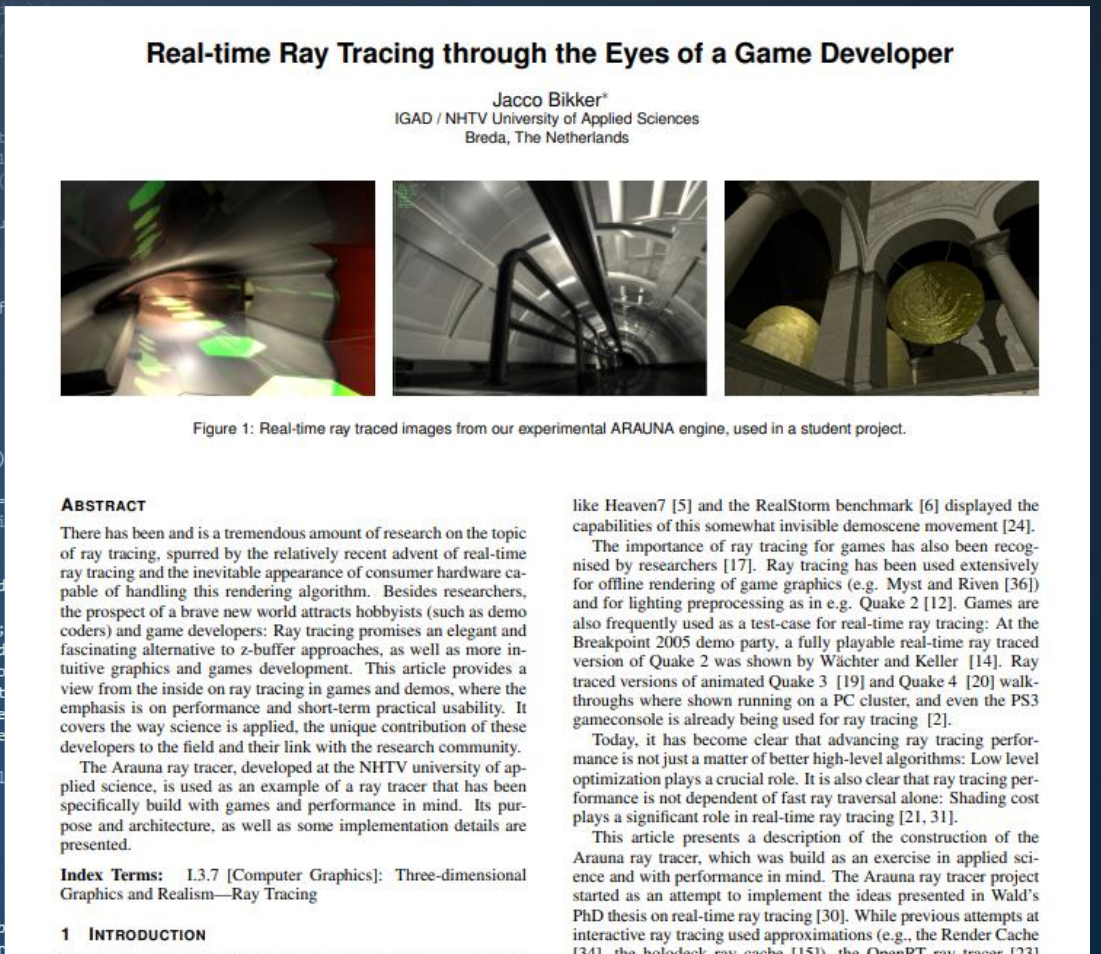
The Design and Evolution of Disney's Hyperion Renderer. Burley et al., 2018.

Manuka: A batch-shading architecture for spectral path tracing in movie production. Fascione et al., 2018.

Mitsuba 2: A Retargetable Forward and Inverse Renderer. Nimier-David, 2019.

Future Work

Real-time



Real-time Ray Tracing through the Eyes of a Game Developer

Jacco Bikker*
IGAD / NHTV University of Applied Sciences
Breda, The Netherlands

Figure 1: Real-time ray traced images from our experimental ARAUNA engine, used in a student project.

ABSTRACT

There has been and is a tremendous amount of research on the topic of ray tracing, spurred by the relatively recent advent of real-time ray tracing and the inevitable appearance of consumer hardware capable of handling this rendering algorithm. Besides researchers, the prospect of a brave new world attracts hobbyists (such as demo coders) and game developers: Ray tracing promises an elegant and fascinating alternative to z-buffer approaches, as well as more intuitive graphics and games development. This article provides a view from the inside on ray tracing in games and demos, where the emphasis is on performance and short-term practical usability. It covers the way science is applied, the unique contribution of these developers to the field and their link with the research community.

The Arauna ray tracer, developed at the NHTV university of applied science, is used as an example of a ray tracer that has been specifically build with games and performance in mind. Its purpose and architecture, as well as some implementation details are presented.

Index Terms: I.3.7 [Computer Graphics]: Three-dimensional Graphics and Realism—Ray Tracing

1 INTRODUCTION

like Heaven7 [5] and the RealStorm benchmark [6] displayed the capabilities of this somewhat invisible demoscene movement [24].

The importance of ray tracing for games has also been recognised by researchers [17]. Ray tracing has been used extensively for offline rendering of game graphics (e.g. Myst and Riven [36]) and for lighting preprocessing as in e.g. Quake 2 [12]. Games are also frequently used as a test-case for real-time ray tracing: At the Breakpoint 2005 demo party, a fully playable real-time ray traced version of Quake 2 was shown by Wächter and Keller [14]. Ray traced versions of animated Quake 3 [19] and Quake 4 [20] walkthroughs where shown running on a PC cluster, and even the PS3 gameconsole is already being used for ray tracing [2].

Today, it has become clear that advancing ray tracing performance is not just a matter of better high-level algorithms: Low level optimization plays a crucial role. It is also clear that ray tracing performance is not dependent of fast ray traversal alone: Shading cost plays a significant role in real-time ray tracing [21, 31].

This article presents a description of the construction of the Arauna ray tracer, which was build as an exercise in applied science and with performance in mind. The Arauna ray tracer project started as an attempt to implement the ideas presented in Wald's PhD thesis on real-time ray tracing [30]. While previous attempts at interactive ray tracing used approximations (e.g., the Render Cache

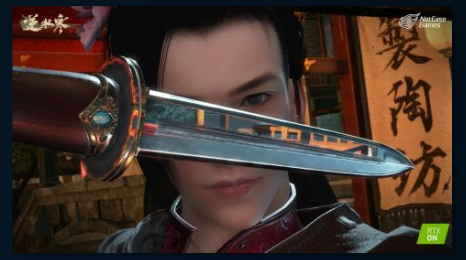
~~Other resources:~~

~~OptiX: A General Purpose Ray Tracing Engine. Parker et al., 2010.~~

~~REAL-TIME RAYTRACING WITH NVIDIA RTX. Stich, 2018.~~

~~(not a lot of research so far…)~~

Ray Tracing Gems 1 & 2,
https://research.nvidia.com/publication/2020-07_Spatiotemporal-reservoir-resampling,
https://research.nvidia.com/publication/2019-07_Temporally-Dense-Ray,
https://research.nvidia.com/publication/2019-03_Improving-Temporal-Antialiasing, …



Temporally Dense Ray Tracing
Semantic Image Synthesis with Spatially-Adaptive Normalization
A Style-Based Generator Architecture for Generative Adversarial Networ
Dynamic Diffuse Global Illumination with Ray-Traced Irradiance Fields
Near-Eye Display and Tracking Technologies for Virtual and Augmented R
NVGaze: An Anatomically-Informed Dataset for Low-Latency, Near-Eye G
Manufacturing Application-Driven Foveated Near-Eye Displays
Ray Tracing Gems
Improving Temporal Antialiasing with Adaptive Ray Tracing
Cool Patches: A Geometric Approach to Ray/Bilinear Patch Intersections
Precision Improvements for Ray/Sphere Intersection
A Fast and Robust Method for Avoiding Self-Intersection
Texture Level of Detail Strategies for Real-Time Ray Tracing
Simple Environment Map Filtering Using Ray Cones and Ray Differentials
Efficient Generation of Points that Satisfy Two-Dimensional Elementary
Massively Parallel Path Space Filtering
Massively Parallel Construction of Radix Tree Forests for the Efficient Sa
Fast, High Precision Ray/Fiber Intersection using Tight, Disjoint Boundin
Massively Parallel Stackless Ray Tracing of Catmull-Clark Subdivision Sur
A Ray-Box Intersection Algorithm and Efficient Dynamic Voxel Rendering
FocusAR: Auto-focus Augmented Reality Eyeglasses for both Real World a
Image Inpainting for Irregular Holes Using Partial Convolutions
Machine Learning and Rendering
Towards Virtual Reality Infinite Walking: Dynamic Saccadic Redirection
Correlation-Aware Semi-Analytic Visibility for Antialiased Rendering
Adaptive Temporal Antialiasing
Phantom Ray-Hair Intersector

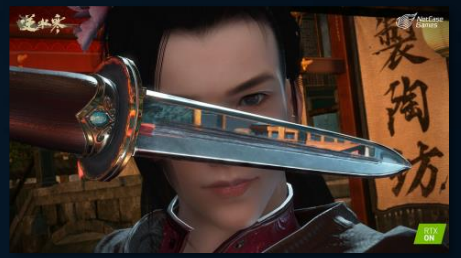
Other resources:

OptiX: A General Purpose Ray Tracing Engine. Parker et al., 2010.

REAL-TIME RAYTRACING WITH NVIDIA RTX. Stich, 2018.

(not a lot of research so far…)

Ray Tracing Gems, https://research.nvidia.com/publication/2020-07_Spatiotemporal-reservoir-resampling, https://research.nvidia.com/publication/2019-07_Temporally-Dense-Ray, https://research.nvidia.com/publication/2019-03_Improving-Temporal-Antialiasing, …



Future Work

Massive Scenes

https://pharr.org/matt/blog/2018/07/16/moana-island-pbrt-all.html

Moana Island scene. Heather Pritchett & Rasmus Tamstorf, 2016. *~15 billion primitives.*

Today's Agenda: *Monte Carlo*

- Sampling an Area Light with One Ray
- Sampling Multiple Area Lights with One Ray
- Difficult Cases: Spherical Lights, Occluded Lights
- The Random Walk
- Random Walk with Next Event Estimation
- Digest
- P3 Topics

INFOMAGR – Advanced Graphics

Jacco Bikker - November 2022 - February 2023

END of “Probability”

next up: “Bidirectional”