Welcome!
Today's Agenda:

- Introduction: Forward Path Tracing
- Multiple Importance Sampling
- Virtual Point Lights
- Photon Mapping
- All Together Now
- Exam Questions Sampler
Forward Path Tracing

Images: Simon Brown, sjbrown.co.uk/2011/01/03/two-way-path-tracing
Forward

Forward Path Tracing

Tracing paths from the light helps when:
- the light is hard to reach
- the light cannot be *importance sampled*.

Tracing paths from the eye is better when:
- the camera is hard to reach.

Many scenes would benefit from both approaches. Now what?
- decide on a per-pixel basis?
- do both and average? (would that even work?)
- something smarter?
**Next Event Estimation:**

- A ray that is aimed at a light, but hits a diffuse surface is discarded.
- A ray that is supposed to sample indirect illumination, but hits a light, is discarded.
When Next Event Estimation Fails

**Light sampling**: paths to random points on the light yield high variance.

**Hemisphere sampling** (with importance): random rays yield low variance.
When Next Event Estimation Fails

**Light sampling**: paths to random points on the light yield low variance.

**Hemisphere sampling** (with importance): random rays yield very high variance.
The Cause of Variance

Sampling the function with a constant pdf: correct result, but potentially a lot of variance.
The Cause of Variance

Sampling the function with a pdf proportional to the function itself: correct result, minimal variance (but: this pdf is generally impossible to obtain).
The Cause of Variance

We can also use *two* pdfs, by taking two samples:

- if we keep both samples, we should average them;
- otherwise, we need to reject one of the samples.
The Cause of Variance

We can also use *two* pdfs, by taking two samples:

- if we keep both samples, we should average them: \( w_1 = 0.5, w_2 = 0.5 \)
- otherwise, we need to reject one of the samples: \( w_1 = 0, w_2 = 1 \) (or vice versa).

Other blends are also possible.

The ideal blend *takes the pdfs themselves into account.*
Today's Agenda:

- Introduction: Forward Path Tracing
- Multiple Importance Sampling
- Virtual Point Lights
- Photon Mapping
- All Together Now
- Exam Questions Sampler
Ray one (red), samples the hemisphere, pdf is $\frac{\cos \theta}{\pi}$.

Ray two (green-ish), samples the lights, pdf is constant (1/SA).
Multiple Importance Sampling

We now have two samples that may return direct light:

- the red ray, which is supposed to sample the hemisphere, so we set its weight to 0;
- the green ray, which has a weight of 1.

A better blend considers both pdfs.

Ray one (red), samples the hemisphere, pdf is \( \frac{\cos \theta}{\pi} \).

Ray two (green-ish), samples the lights, pdf is constant (1/SA).
Multiple Importance Sampling

We now have two samples that may return direct light:

- the red ray, which is supposed to sample the hemisphere, so we set its weight to 0;
- the green ray, which has a weight of 1.

A better blend considers both pdfs.

1.Pdf for hemisphere sampling: $\cos \theta \pi$ (where $\theta$ depends on the generated random bounce);
2. Pdf for next event estimation: $\frac{1}{SA}$

We can simply average these pdfs: $pdf_{\text{avergae}} = w_1 pdf_1 + w_2 pdf_2$

(which is valid if $w_1 + w_2 = 1$).

Or, we can use the balance heuristic* to calculate the weights: $w_i = \frac{p_i(x)}{p_1(x) + p_2(x)}$.

**MIS**

Multiple Importance Sampling

**Practical MIS:**

We sample irradiance using two samples, each chosen according to its own pdf.

\[
pdf_{brdf}(\omega_{i, brdf}) = \frac{\cos \theta_{i, brdf}}{\pi}, \quad pdf_{nee}(\omega_{i, nee}) = \frac{1}{SA}
\]

If the first sample (i.e., \(\omega_{i, brdf}\)) hits a light, we calculate its MIS pdf using the two MIS weights:

\[
w_1 = \frac{pdf_{brdf}(\omega_{i, brdf})}{pdf_{brdf}(\omega_{i, brdf}) + pdf_{nee}(\omega_{i, brdf})}, \quad w_2 = \frac{pdf_{nee}(\omega_{i, brdf})}{pdf_{brdf}(\omega_{i, brdf}) + pdf_{nee}(\omega_{i, brdf})}
\]

\[
pdf_{MIS}(\omega_{i, brdf}) = w_1 \cdot pdf_{brdf}(\omega_{i, brdf}) + w_2 \cdot pdf_{nee}(\omega_{i, brdf})
\]

The first sample is now scaled by \(w_1\) instead of 0.
**MIS**

Multiple Importance Sampling

Practical MIS:

We sample irradiance using two samples, each chosen according to its own pdf.

\[
\text{pdf}_{\text{brdf}}(\omega_{i,\text{brdf}}) = \frac{\cos \theta_{i,\text{brdf}}}{\pi}, \quad \text{pdf}_{\text{nee}}(\omega_{i,\text{nee}}) = \frac{1}{SA}
\]

If the second sample (i.e. \(\omega_{i,\text{nee}}\)) hits a light, we calculate its MIS pdf using the two MIS weights:

\[
w_1 = \frac{\text{pdf}_{\text{brdf}}(\omega_{i,\text{nee}})}{\text{pdf}_{\text{brdf}}(\omega_{i,\text{nee}}) + \text{pdf}_{\text{nee}}(\omega_{i,\text{nee}})} , \quad w_2 = \frac{\text{pdf}_{\text{nee}}(\omega_{i,\text{nee}})}{\text{pdf}_{\text{brdf}}(\omega_{i,\text{nee}}) + \text{pdf}_{\text{nee}}(\omega_{i,\text{nee}})}
\]

\[
\text{pdf}_{\text{MIS}}(\omega_{i,\text{nee}}) = w_1 \text{pdf}_{\text{brdf}}(\omega_{i,\text{nee}}) + w_2 \text{pdf}_{\text{nee}}(\omega_{i,\text{nee}})
\]

Note: we use \(\omega_{i,\text{nee}}\) now; on the previous slide we evaluated the pdfs for \(\omega_{i,\text{brdf}}\).
Advanced Graphics – Bidirectional

MIS

```
float l = (depth < MAXDEPTH)

if (inside) {
    if (nt > 0.5) {
        n.x = 1.0 - n.x;
        n.y = 1.0 - n.y;
        n.z = 1.0 - n.z;
    }
    r = (r + (0.9 * n.x * n.x));
    E' = diffuse;
    E = true;
}

if (r < depth) {
    I = true;
    light = SampleLight(brand, I, N, B2, B2);
    I = light ;
    I = int (I > 0.0);
    E' = true;
    E = false;
    if (E') {
        E = true;
        pdf = SampleDiffuse( diffuse, N, r, r, 0.0, 0.0);
        pdf = E' * (dot(N, r) / pdf);
        E = true;
    }
```
Advanced Graphics – Bidirectional
Advanced Graphics – Bidirectional

MIS

```
if (depth < PRIMARY) {
    if (inside) {
        I = NT / (df + 1);  // Transmission
    } else {
        I = 1.0F - NT / Nd;
    }
    if (NT > I) {
        I = NT / (df + 1);
        I = 1.0F - NT / Nd;
    }
}

if (I * diffuse) {
    E = true;
}

if (I * refr) && (depth < PRIMARY) {
    E = true;
}

// WADEPTH)
survive = $SurviveProbability; diffuse;

estimtion = doing it properly, like:

result = SampleLight: Brand, I, AI, Blight,
          .x + radius.y + radius.z > 0)  && (theta <

E = true;

hit BREATH = EvaluateDistance( L, N ) * $Survive;
att factor = diffuse * trans;
weight = M2i( direct, breather );
att coefficient = dot( N, L );
E "((weight + cosTheta)(1 / direct))" (radius

if (shadow walk = done properly, closely following breather)

1 = SampleDiffuse( diffuse, N, r1, r2, &N, &pdf );

wrangle;

pdf = E = bwdf * ( dot(N, R ) / pdf );
```
Multiple Importance Sampling

- energy returned by k=1 paths
  - EL - LE
- energy returned by k=2 paths
  - EDL - LDE
- energy returned by k=3 paths
  - E(D|S)DL
  - L(D|S)DE
- energy returned by k=4 paths
- energy returned by k=5 paths
Today's Agenda:

- Introduction: Forward Path Tracing
- Multiple Importance Sampling
- Virtual Point Lights
- Photon Mapping
- All Together Now
- Exam Questions Sampler
Instant Radiosity*

Idea:

Trace from the light sources, record the first hit. Do this for $N$ particles (where $N$ is $\sim 10^3..10^5$). Each particle becomes a virtual point light.

Now, render the scene as usual (rasterization, or Whitted-style ray tracing). At the first diffuse surface, use the VPLs to estimate indirect light, and the lights themselves for direct illumination.

VPLs

Instant Radiosity

Images: M. Hasan, SIGGRAPH Asia ‘09.
Instant Radiosity

Illumination from $N$ point lights:

$$L(s \leftarrow x) = \frac{k_d(x)}{\pi} \sum_{y \in VPLS} V(x \leftrightarrow y) \frac{\cos \theta_o \cos \theta_i}{\|x - y\|^2} L(y \leftarrow y)$$

Where $L(y \leftarrow)$ is the radiance arriving at the diffuse surface from light $z$:

$$L(y \leftarrow z) = \frac{k_d(y)}{\pi} L_e(z) \cos \theta_{i,y}$$

Or, after $n$ bounces:

$$L(y \leftarrow) = L_e(y) \prod_{j=0}^{n} \frac{k_d(p_j)}{\pi} \cos \theta_{i,p_j}$$

Calculating $L_e(z)$:

$$x, pdf(x) = \text{RandomPointOnLight}()$$

$$\text{rad} = L(x)/pdf(x)$$
VPLs

Instant Radiosity

Using VPLs has some interesting characteristics:

- No noise! Those splotches though...
- VPLs can bounce: they can represent all indirect light
- VPLs cannot represent direct light
- #VPLs < #pixels
- Evaluating VPLs can be done with or without occlusion
- VPL visibility can also be evaluated using shadow maps
- Instant Radiosity is a bidirectional technique: we propagate flux when placing the VPLs, and we propagate importance when connecting to them.
Today's Agenda:

- Introduction: Forward Path Tracing
- Multiple Importance Sampling
- Virtual Point Lights
- Photon Mapping
- All Together Now
- Exam Questions Sampler
Photon Mapping*

Idea: what if we got rid of visibility queries altogether?

With the photon mapping algorithm, we split rendering in two phases:

- In phase 1 we deposit flux ($\Phi$) in the scene by tracing a large number of photons;
- In phase 2, we estimate illumination using the photon map.

Photon Mapping

Phase 1: propagating flux.

Photon emission:

- Point light: emitted in uniformly distributed random directions from the point.
- Area light: emitted from random positions on the square, with directions limited to a hemisphere. The emission directions are chosen from a cosine distribution.

All photons thus have the same power: their density is the only way to express varying brightness.
Photons

Photon Mapping

Phase 1: propagating flux.

Surface interaction:
A photon that hits a surface may get absorbed or reflected. We chose using random numbers / Russian roulette.

Note:
In a monochromatic simulation, this maintains the direct relation between photon density and local brightness. With RGB color this relation is lost.
Photons

Photon Mapping

Phase 1: propagating flux.

Photon storage:
At each non-specular path vertex we store the photon:

```c
struct photon {
    float3 position; // world space position of the photon hit
    float3 power; // current power level for the photon
    float3 L; // incident direction
};
```

A photon may be stored multiple times along its path before it gets absorbed. Since the total set of photons represents the illumination, we divide photon power by the total number of stored photons.
Photon Mapping

Phase 2: radiance estimation.

In the second pass, we render the scene using rasterization or Whitted-style ray tracing; the photon map is used to estimate illumination.

At each non-specular path vertex we estimate the reflected radiance:

$$L(x, \omega_o) = \int_{\Omega_x} f_r(x, \omega_i, \omega_o) L_i(x, \omega_i) \cos \theta_i \, d\omega_i$$

This requires information about the radiance $L_i(x, \ldots)$ arriving over the hemisphere $\Omega_x$. We estimate this radiance by looking at the photons that arrived near $x$:

$$L(x, \omega_o) \approx \frac{1}{\pi r^2} \sum_{p=1}^{N} f_r(x, \omega_p, \omega_o) \Delta\Phi(x, \omega_p)$$
Phottos

Photon Mapping

Phase 2: radiance estimation.

We estimate this radiance by looking at the photons that arrived near $x$:

$$L(x, \omega_o) \approx \frac{1}{\pi r^2} \sum_{p=1}^{N} f_r(x, \omega_p, \omega_o) \Delta \Phi(x, \omega_p)$$

Note:

- We assume that we gathered photons on a disc of radius $r$.
- We assume that the gathered photons belong to the same surface.
- Each photon within radius $r$ has the same influence on the estimate.
Photon Mapping

Phase 2: radiance estimation.

Instead of using the same weight for each photon we can use a filter:

\[ w_{pg} = \alpha \left[ 1 - \frac{1 - e^{-\beta d_p^2}}{1 - e^{-\beta}} \right] , \]

where \( \alpha = 0.918, \beta = 1.953^* \). Value \( d_p^2 \) is the squared distance between photon \( p \) and \( x \).

Now:

\[ L(x, \omega_o) \approx \sum_{p=1}^{N} f_r(x, \omega_p, \omega_o) \Delta \Phi(x, \omega_p) w_{pg} . \]

*: Mark J. Pavicic, Convenient Anti-Aliasing Filters that Minimize Bumpy Sampling. In Graphics Gems I.
Photons

Photon Mapping

Algorithm characteristics:

- Low-frequent noise
- Can be used in a rasterizer
- Can be used for direct + indirect
- Still a bidirectional technique
Photons

```c
bbox (depth < PREVIOUS)

if (Inside || ! 
  r < t / mg2) 
  return;

if (r0 - r1 - t) < 0) 
  return;

E = diffuse; 
  if (r1 + r2) && (depth < PREVIOUS)
    return;
  
  r0.;

E = diffuse; 
  if (true;

SampleDiffuse( diffuse, r1, r2, &E, &pdf )

E = E * pdf * (dot(N, R) / pdf);
```
Photons
Photons

Photon Mapping

Reducing low-frequency noise:

*Final gather:*

Consequence: we cannot sample EDL paths.
Today’s Agenda:

- Introduction: Forward Path Tracing
- Multiple Importance Sampling
- Virtual Point Lights
- Photon Mapping
- All Together Now (TODO)
- Exam Questions Sampler
An exam can be seen as a Monte-Carlo process. Explain why.
After reading the probability tutorial, answer these:

a) What is a definite integral?

b) What do we mean by an analytical solution?

c) How is the Riemann sum defined (mathematically)?

d) What is ‘univariate’?

e) What is ‘aliasing’?

f) Define, in your own words, ‘expected value’.

g) What is ‘deviation’ in the context of probability theory?

And, finally:

When using importance sampling, we assume that for \( N = \infty \),
\[
\frac{b-a}{N} \sum_{i=1}^{N} \frac{f(X)}{p(X)} = \frac{b-a}{N} \sum_{i=1}^{N} f(X), \text{ if } \int_a^b p(x)dx = 1.
\]
Provide one example for which this is not true.
A scene is illuminated by a single double-sided square light source.

Two algorithms are used to sample the light source: the first picks a random point on a random side of the light source, while the second algorithm only picks random points on the side of the light source facing point $p$.

a) Write down the Monte-Carlo integrator that estimates the illumination on point $p$ using the first algorithm.

b) Write down the Monte-Carlo integrator that estimates the illumination on point $p$ using the second algorithm.

Note: both methods should obviously produce the same answer, on average.
Today's Agenda:

- Introduction: Forward Path Tracing
- Multiple Importance Sampling
- Virtual Point Lights
- Photon Mapping
- All Together Now
- Exam Questions Sampler
INFOMAGR – Advanced Graphics
Jacco Bikker - November 2018 – February 2019

END of “Bidirectional”

next lecture: “Filtering”