Lec 16 - Monte Carlo Path Tracing

一种近似积分方法：Monte Carlo Integral

- 有时候定积分很难精确计算（解析式求不出），使用数值方法
- 黎曼积分：分解成很多个长方形来积分
- Monte Carlo积分：随机采样
- 用任意一个PDF去采样，都可以用下面的式子求出积分的近似数值

$$F_N=\frac{1}{N}\sum_{i=1}^{N}\frac{f\left(X_i\right)}{p\left(X_i\right)}$$

- Uniform: p(x) = 1/(b-a)

$$\int f(x)\mathrm{d}x=\frac{1}{N}\sum_{i=1}^{N}\frac{f\left(X_i\right)}{p\left(X_i\right)}\quad X_i\sim p(x)$$

- N: 采样数
- 样本越多，结果越准
- 在x上采样的样本就要在x上积分

路径追踪 Path Tracing

Motivation: Whitted-Style Ray Tracing

- Always perform specular reflections / refractions
- Stop bouncing at diffuse surfaces
- 这些简化不一定正确
  - Where should the ray be reflected for glossy materials? 反射到镜面对应的方向附近一圈，而非单单镜面反射
  - No reflections between diffuse materials? 漫反射不应停下，否则少了很多
    - Color bleeding: 由于漫反射，颜色流到了其他面上
- Whitted-Style Ray Tracing is Wrong
- the rendering equation is correct (部分简化光线的性质)

$$L_o\left(p,\omega_o\right)=L_e\left(p,\omega_o\right)+\int_{\Omega^+}L_i\left(p,\omega_i\right)f_r\left(p,\omega_i,\omega_o\right)\left(n\cdot\omega_i\right)\mathrm{d}\omega_i$$

  - 但是它包括了对半球面的积分，还有递归
  - 但是我们只需要solve this integral numerically → Monte Carlo

We want to compute the radiance at p towards the camera

$$L_o(p,\omega_o) = \int_{\Omega^+} L_i(p,\omega_i) f_r(p,\omega_i,\omega_o)(n\cdot\omega_i)\,\mathrm{d}\omega_i$$

Monte Carlo integration: $\int_a^b f(x)\,\mathrm{d}x \approx \frac{1}{N}\sum_{k=1}^{N}\frac{f(X_k)}{p(X_k)} \qquad X_k \sim p(x)$

What's our "f(x)"? $\qquad L_i(p,\omega_i) f_r(p,\omega_i,\omega_o)(n\cdot\omega_i)$

What's our pdf? $\qquad p(\omega_i) = 1/2\pi$

(assume uniformly sampling the hemisphere)

最简化：只考虑直接光照，只考虑非光源

$$\begin{aligned} L_o(p,\omega_o) &= \int_{\Omega^+} L_i(p,\omega_i) f_r(p,\omega_i,\omega_o)(n\cdot\omega_i)\mathrm{d}\omega_i \\ &\approx \frac{1}{N}\sum_{i=1}^{N}\frac{L_i(p,\omega_i) f_r(p,\omega_i,\omega_o)(n\cdot\omega_i)}{p(\omega_i)} \end{aligned}$$

也考虑间接光照：递归计算

```
shade(p, wo)
    Randomly choose N directions wi~pdf
    Lo = 0.0
    For each wi
        Trace a ray r(p, wi)
        If ray r hit the light
            Lo += (1 / N) * L_i * f_r * cosine / pdf(wi)
        Else If ray r hit an object at q
            Lo += (1 / N) * shade(q, -wi) * f_r * cosine
            / pdf(wi)
    Return Lo
```

问题：

- 光线数量爆炸

解决方式：每次只打出一条光线

From now on, we always assume that
only 1 ray is traced at each shading point:

```
shade(p, wo)
    Randomly choose ONE direction wi~pdf(w)
    Trace a ray r(p, wi)
    If ray r hit the light
        Return L_i * f_r * cosine / pdf(wi)
    Else If ray r hit an object at q
        Return shade(q, -wi) * f_r * cosine / pdf(wi)
```

This is **path tracing**! (FYI, Distributed Ray Tracing if N != **1)**

- noisy严重
  - 解决方案：每个像素取多个不同路径计算 → Ray Generation

```
ray_generation(camPos, pixel)
    Uniformly choose N sample positions within the pixel
    pixel_radiance = 0.0
    For each sample in the pixel
        Shoot a ray r(camPos, cam_to_sample)
        If ray r hit the scene at p
            pixel_radiance += 1 / N * shade(p, sample_to_cam)
    Return pixel_radiance
```

问题2: 停不下来

解决方案2-1: 层数

- 缺陷：结果能量会损失

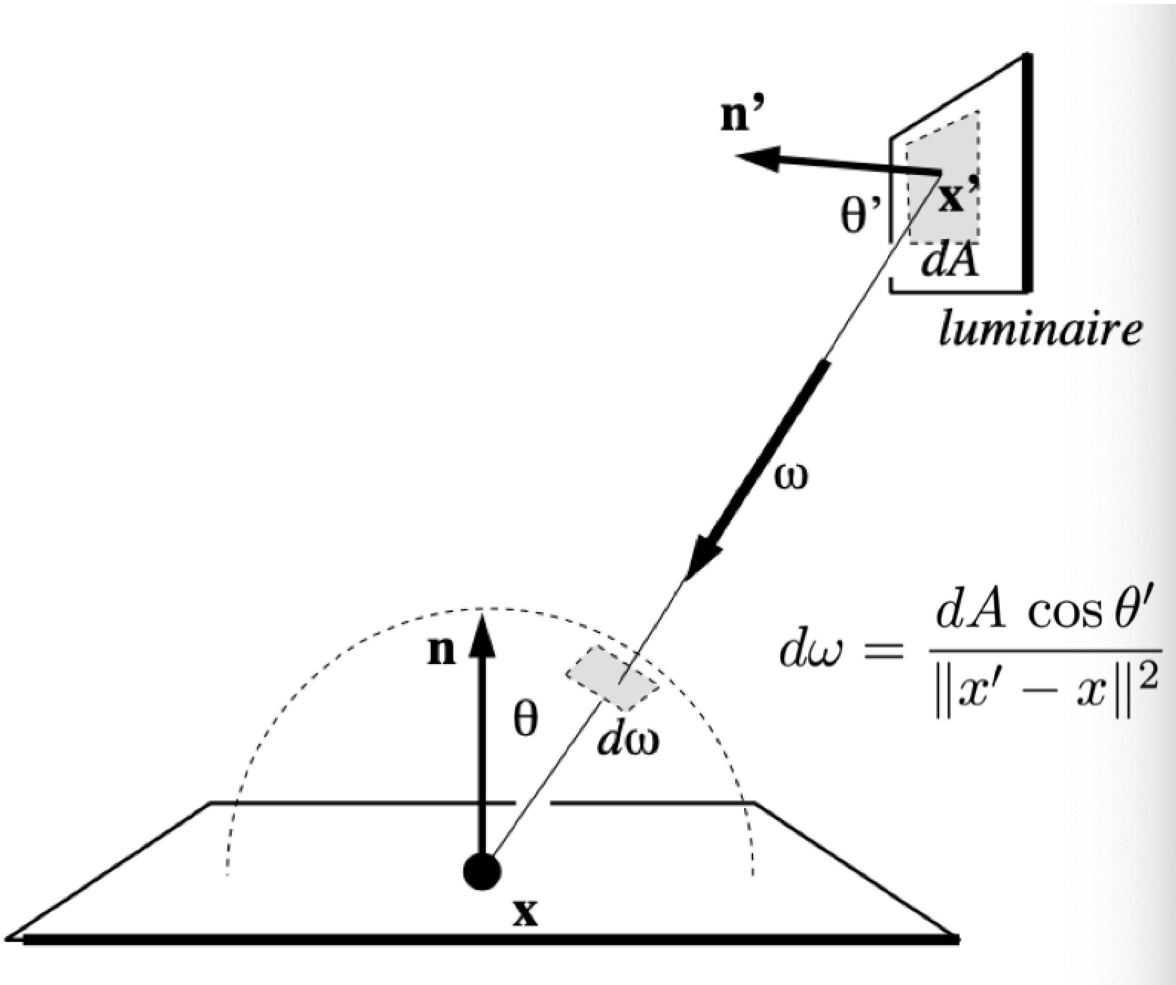
解决方案2-2：Russian Roulette (RR)

- With probability 0 < P < 1, you are fine，继续发出光线，**return the shading result divided by P: Lo / P** ; With probability 1 - P, otherwise 停止计算，返回0
- 结果正确

```
shade(p, wo)
    Manually specify a probability P_RR
    Randomly select ksi in a uniform dist. in [0, 1]
    If (ksi > P_RR) return 0.0;

    Randomly choose ONE direction wi~pdf(w)
    Trace a ray r(p, wi)
    If ray r hit the light
        Return L_i * f_r * cosine / pdf(wi) / P_RR
    Else If ray r hit an object at q
        Return shade(q, -wi) * f_r * cosine / pdf(wi) / P_RR
```

- 但是很不高效
  - 原因：均匀采样导致，环境当中光线的来源往往不是均匀的。a lot of rays are “wasted” if we uniformly sample the hemisphere at the shading point.
  - 解决方案1: Sampling the Light
    - 蒙特卡洛允许任何方式的采样，只要喂对应的x和p就行
    - 对光源积分是个很高效的想法，但是积分的对象和"Sample on x & integrate on x"的要求不匹配→只需要找到光源对应ω_i的关系就行→改变积分域



$$
\begin{aligned}
L_o(x,\omega_o) &= \int_{\Omega^+} L_i(x,\omega_i) f_r(x,\omega_i,\omega_o) \cos\theta \mathrm{d}\omega_i \\
&= \int_A L_i(x,\omega_i) f_r(x,\omega_i,\omega_o) \frac{\cos\theta\cos\theta'}{\|x'-x\|^2} \mathrm{d}A
\end{aligned}
$$

- 然后我们就可以consider the radiance coming from two parts:

    1. light source (direct, no need to have RR) 直接光照
    2. other reflectors (indirect, RR) 间接光照

```
shade(p, wo)
    # Contribution from the light source.
    Uniformly sample the light at x' (pdf_light = 1 / A)
    L_dir = L_i * f_r * cos θ * cos θ' / |x' - p|^2 / pdf_light

    # Contribution from other reflectors.
    L_indir = 0.0
    Test Russian Roulette with probability P_RR
    Uniformly sample the hemisphere toward wi (pdf_hemi = 1 / 2pi)
    Trace a ray r(p, wi)
    If ray r hit a non-emitting object at q
        L_indir = shade(q, -wi) * f_r * cos θ / pdf_hemi / P_RR

    Return L_dir + L_indir
```

- 另外还需检测光源和目标点中间有没有阻挡

Path Tracing确实很难：物理、概率、微积分、代码......

并不那么入门，但是是现代化图形学的一个根基，almost 100% correct, a.k.a. **PHOTO-REALISTIC**

Ray tracing 概念的区分：

- Previous：Ray tracing == Whitted-style ray tracing
- Modern
    - The general solution of light transport, including
        - (Unidirectional & bidirectional) path tracing
        - Photon mapping
        - Metropolis light transport
        - VCM / UPBP...

课上没讲的：

- Uniformly sampling the hemisphere
    - How? And in general, how to sample any function?(sampling)
- Monte Carlo integration allows arbitrary pdfs
    - What's the best choice? (importance sampling) **重要性采样理论**
- Do random numbers matter?

    - Yes! (low discrepancy sequences)比如蓝噪音
- I can sample the hemisphere and the light
    - Can I combine them? Yes! (multiple imp. sampling)
- The radiance of a pixel is the average of radiance on all paths passing through it
    - Why? (pixel reconstruction filter)
- Is the radiance of a pixel the color of a pixel?
    - No. (gamma correction（radiance到color的对应关系）, curves(HDR), color space)