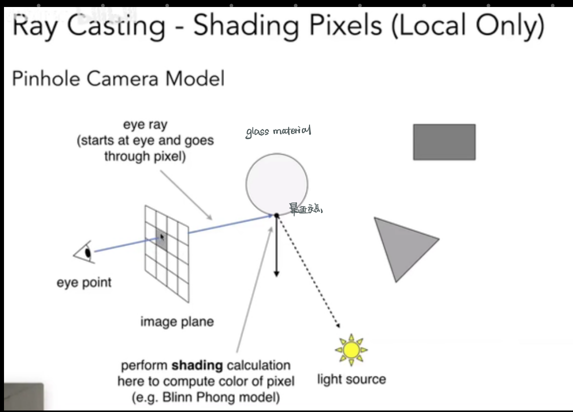
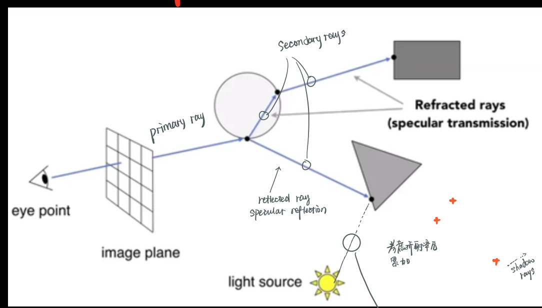
Why ray tracing? 光栅化缺点：无法解决全局效果 ① soft shadows ② bouncing lights <glossy surface> 间接光照

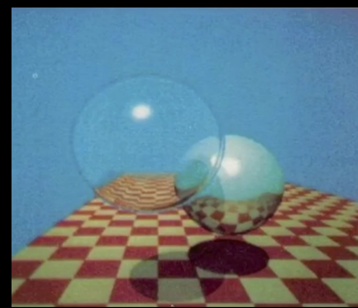
Basic ideas ① Light travels in straight line ② Light rays don't collide each other ③ 光源→场景→眼 Resiprocity 可逆性



Recursive Ray Tracing

Whitted-style

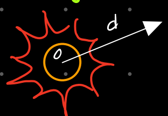




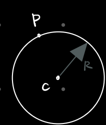
- 在每一个弹射点计算着色值 <将反射点与光源连线>
- 多次反射路径上的每个反射点均累加进对应屏幕像素

How to calculate

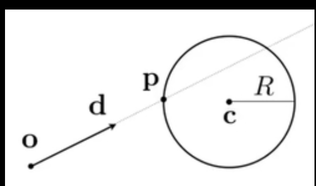
ray-surface intersection



ray equation $r(t) = o + td$, $t \in [0, \infty)$



$(p - c)^2 - R^2 = 0$



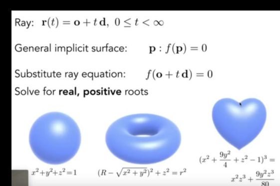
交点同时在二者上

$(o + td - c)^2 - R^2 = 0$

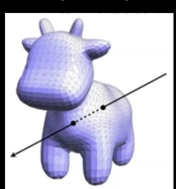
$t > 0$, $b^2 \geq 4ac$ 非虚数解

$$at^2 + bt + c = 0, \text{ where}$$
$$a = \mathbf{d} \cdot \mathbf{d}$$
$$b = 2(\mathbf{o} - \mathbf{c}) \cdot \mathbf{d}$$
$$c = (\mathbf{o} - \mathbf{c}) \cdot (\mathbf{o} - \mathbf{c}) - R^2$$
$$t = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

针对隐式表面求t值

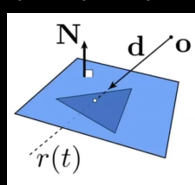


与Triangle Mesh求交？



0 or 1 intersection?
图形学不考虑相切
寻找t最小 <primary rays>
但需依次判断，慢

Ray Intersects with Triangle



① Ray-Plane Intersection
② Is the point inside triangle

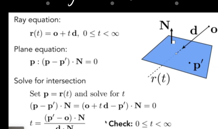
点+法线定义平面

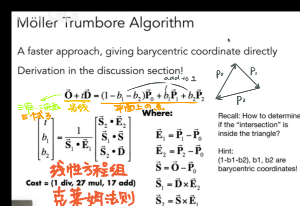
垂直

<x,y,z> <x_0,y_0,z_0> <N_x,N_y,N_z>

$(p - p') \cdot N = 0$

$ax + by + cz + d = 0$





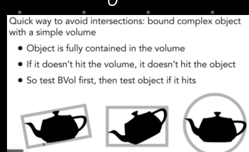
Accelerating ray-surface intersection

problem to solve

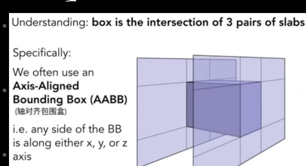
Simple ray-scene intersection
- Exhaustively test ray-intersection with every object
- Find the closest hit (with minimum t)

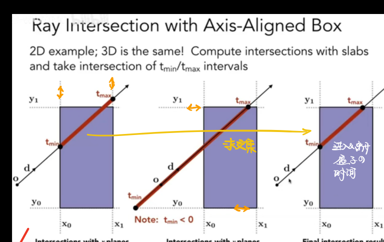
Problem:
- Naive algorithm = #pixels x #objects (x #bounces)
- Very slow!

Bounding Volume 包围盒



若未碰撞包围盒，则不碰内部之物体





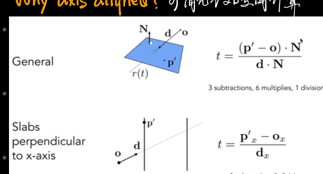
- Recall: a box (3D) = three pairs of infinitely large slabs
- Key ideas
  - The ray enters the box **only when** it enters all pairs of slabs
  - The ray exits the box **as long as** it exits any pair of slabs
- For each pair, calculate the t_{min} and t_{max} (negative is fine)
- For the 3D box, $t_{enter} = \max\{t_{min}\}$, $t_{exit} = \min\{t_{max}\}$
- If $t_{enter} < t_{exit}$, we know the ray **stays a while** in the box (so they must intersect!) (not done yet, see the next slide)

- However, ray is not a line
  - Should check whether t is negative for physical correctness!
- What if $t_{exit} < 0$?
  - The box is "behind" the ray — no intersection!
- What if $t_{exit} >= 0$ and $t_{enter} < 0$?
  - The ray's origin is inside the box — have intersection!
- In summary, ray and AABB intersect iff
  - $t_{enter} < t_{exit}$ && $t_{exit} >= 0$

本篇目录

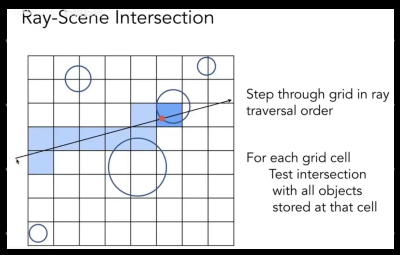
- Basic ray tracing
  - Ray generation
  - Ray object intersection
- Acceleration
  - Ray AABB intersection
  - Spatial partitions vs object partitions
  - BVH traversal

Why axis aligned? 可简化为2D空间计算



Using AABB to accelerate ray tracing

○ Uniform grids <不常用>



前提：光线→盒子比较容易

Grid resolution ? 1 grid ×

Too Many ×

Heuristic - # cells = C * #objs

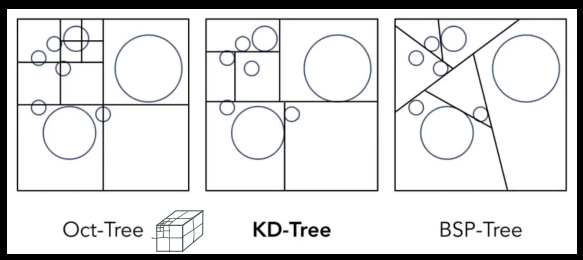
C ≈ 27 <3D>

✓

×

Too many traversal in empty cells

○ Spatial partitions <改进版>



When to stop division?

真到当前区内物体数量太多

× 随着维度提高，方量增大

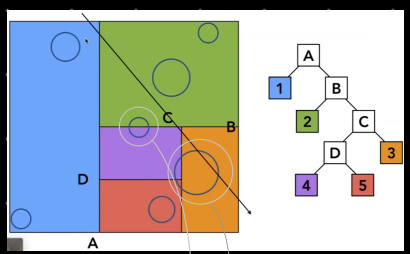
每次只沿一条轴切一次

<尽量交替 x→y→z>

保持二叉性质

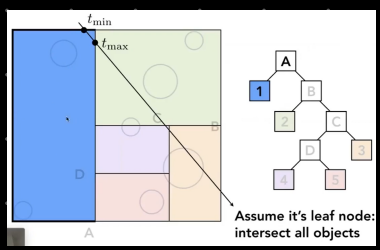
维度越高 越难计算

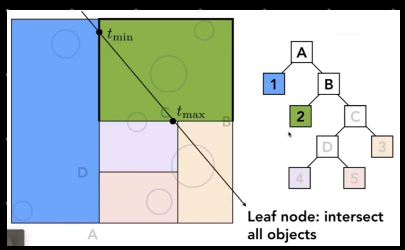
KD Tree <近期不常用>



Leaf nodes stores the geometries intersecting with the current grid

△如何定义到个中点？视情况决定





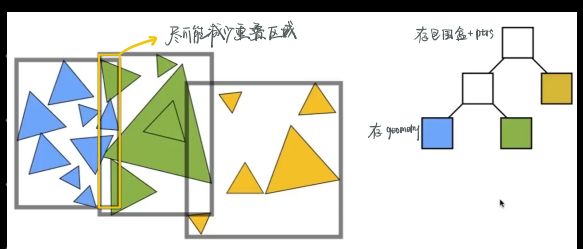
类二分查找，直到找到真 t_{min} →

Problems

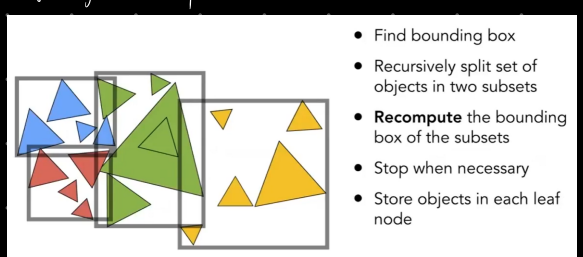
× 如何判断三角形 <mesh triangle> 与 BB 交集？ 顶点判断并不通用

× 同一 Geometry 存储于不同 leaf，并不直观

○ Object Partitions & Bounding Volume Hierarchy BVH <流行>



划分 geometries 并划出 BB



✡ 划分方式是算法重点、

均匀分布 + 避免重合

Heuristic 1 Always split the longest axis

Heuristic 2 Split node at the median object

沿目标轴 → 判定各三角形重心

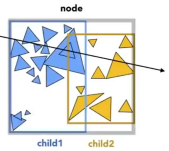
Topk 问题 → partition 减治算法

$O(n) = n + n/2 + n/4 + \cdots < 2n$

Heuristic ends when node contains too few elems

BVH Traversal

```
Intersect(Ray ray, BVH node) {
  if (ray misses node.bbox) return;

  if (node is a leaf node)
     test intersection with all objs;
     return closest intersection;

  hit1 = Intersect(ray, node.child1);
  hit2 = Intersect(ray, node.child2);

  return the closer of hit1, hit2;
}
```

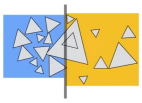


优缺点比较

Spatial vs Object Partitions

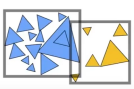
Spatial partition (e.g.KD-tree)

- Partition space into non-overlapping regions
- An object can be contained in multiple regions



Object partition (e.g. BVH)

- Partition set of objects into disjoint subsets
- Bounding boxes for each set may overlap in space



Radiometry 辐射度量学 — 精确描述光的属性/行为 <比如 light Intensity in Blinn-Phong>

- Measurement & units for illumination
- Accurately measure the spatial property of light
- perform light calculation in physically correct way

{ Radiant Flux 辐射通量
Intensity 辐射强度
Irradiance 辐射照度
Radiance 辐射亮度

Radiant Energy & Flux <Power>

Definition: Radiant energy is the energy of electromagnetic radiation. It is measured in units of joules, and denoted by the symbol:

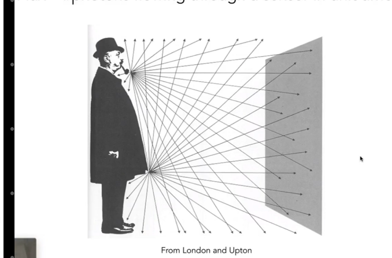
$$Q \ [\mathrm{J = Joule}]$$

Definition: Radiant flux (power) is the energy emitted, reflected, transmitted or received, per unit time.

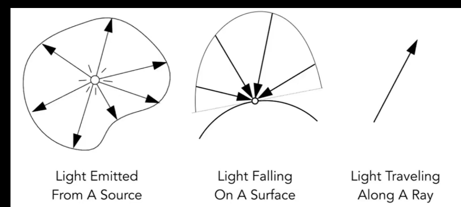
$$\Phi \equiv \frac{\mathrm{d}Q}{\mathrm{d}t} \ [\mathrm{W = Watt}] \ [\mathrm{lm = lumen}]^*$$

光学功率

Flux – #photons flowing through a sensor in unit time



Radiant Flux

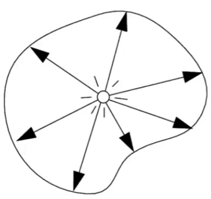


Radiant Intensity Irradiance Radiance

Radiant Intensity power per solid angle

什么是立体角？

Definition: The radiant (luminous) intensity is the power per unit **solid angle (?)** emitted by a point light source.
(立体角)



$$I(\omega) \equiv \frac{\mathrm{d}\Phi}{\mathrm{d}\omega}$$

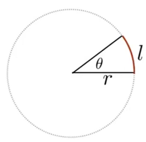
$$\left[\frac{\mathrm{W}}{\mathrm{sr}}\right]\left[\frac{\mathrm{lm}}{\mathrm{sr}} = \mathrm{cd} = \mathrm{candela}\right]$$

The candela is one of the seven SI base units.

Angles and Solid Angles

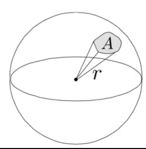
Angle: ratio of subtended arc length on circle to radius

- $\theta = \frac{l}{r}$
- Circle has 2π radians



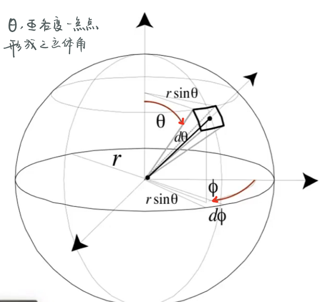
Solid angle: ratio of subtended area on sphere to radius squared

- $\Omega = \frac{A}{r^2}$
- Sphere has 4π steradians



微分立体角 <在单位球上>

θ、φ各有度一点点，形成之立体角



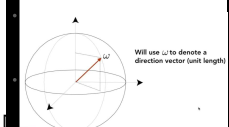
$$\mathrm{d}A = (r\,\mathrm{d}\theta)(r\sin\theta\,\mathrm{d}\phi)$$
面积 $= r^2\sin\theta\,\mathrm{d}\theta\,\mathrm{d}\phi$

$$\mathrm{d}\omega = \frac{\mathrm{d}A}{r^2} = \sin\theta\,\mathrm{d}\theta\,\mathrm{d}\phi$$
立体角

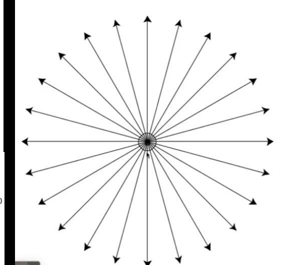
靠近极点/赤道此值不同

Sphere: S^2

$$\Omega = \int_{S^2} \mathrm{d}\omega = \int_0^{2\pi}\int_0^{\pi} \sin\theta\,\mathrm{d}\theta\,\mathrm{d}\phi = 4\pi$$

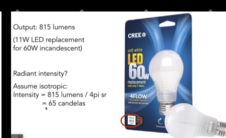


Isotropic Point Source



power
$$\Phi = \int_{S^2} I\,\mathrm{d}\omega = 4\pi I$$
所有方向上 Intensity之和

$$I = \frac{\Phi}{4\pi}$$
某一方向上的Intensity



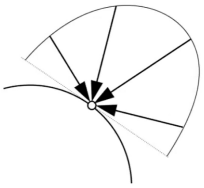
Irradiance power received per unit area 即为光线垂直/投影至垂直方向

Intensity 不会变：立体角不变，越远光能范围越大

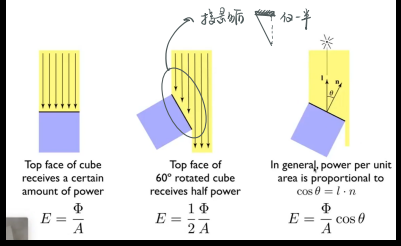
Definition: The irradiance is the power per (perpendicular/ projected) unit area incident on a surface point.

$$E(\mathbf{x}) \equiv \frac{\mathrm{d}\Phi(\mathbf{x})}{\mathrm{d}A}$$

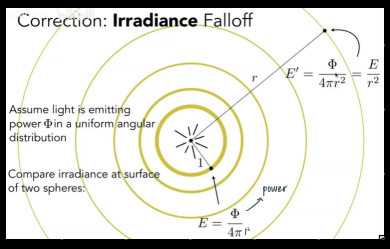
$$\left[\frac{\mathrm{W}}{\mathrm{m}^2}\right] \left[\frac{\mathrm{lm}}{\mathrm{m}^2} = \mathrm{lux}\right]$$



Lambert's Cosine Law



Correction: **Irradiance** Falloff



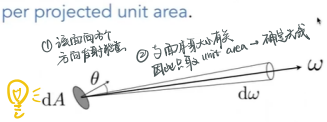
Radiance 描述光线传播中的能量变化

Radiance is the fundamental field quantity that describes the distribution of light in an environment

- Radiance is the quantity associated with a ray
- Rendering is all about computing radiance

Light Traveling Along A Ray

Definition: The radiance (luminance) is the power emitted, reflected, transmitted or received by a surface, per unit solid angle, per projected unit area.



$$L(\mathrm{p},\omega) \equiv \frac{\mathrm{d}^2\Phi(\mathrm{p},\omega)}{\mathrm{d}\omega\,\mathrm{d}A\cos\theta}$$

$\cos\theta$ accounts for projected surface area

$$\left[\frac{\mathrm{W}}{\mathrm{sr\,m}^2}\right] \left[\frac{\mathrm{cd}}{\mathrm{m}^2} = \frac{\mathrm{lm}}{\mathrm{sr\,m}^2} = \mathrm{nit}\right]$$

Radiance

Definition: power per unit solid angle per projected unit area.

$$L(\mathrm{p},\omega) \equiv \frac{\mathrm{d}^2\Phi(\mathrm{p},\omega)}{\mathrm{d}\omega\,\mathrm{d}A\cos\theta}$$

Recall

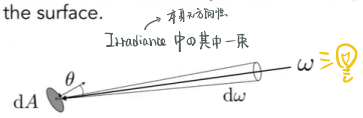
- Irradiance: power per projected unit area
- Intensity: power per solid angle

So

- Radiance: Irradiance per solid angle
- Radiance: Intensity per projected unit area

Incident Radiance

Incident radiance is the irradiance per unit solid angle arriving at the surface.



$$L(\mathrm{p},\omega) = \frac{\mathrm{d}E(\mathrm{p})}{\mathrm{d}\omega\cos\theta}$$

i.e. it is the light arriving at the surface along a given ray (point on surface and incident direction).

relationship between irradiance & radiance

Irradiance vs. Radiance

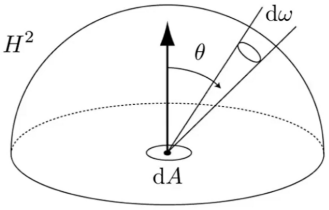
Irradiance: total power received by area dA

Radiance: power received by area dA from "direction" dω

$$\underset{\text{Irradiance}}{dE(\mathrm{p},\omega)} = \underset{\text{Radiance}}{L_i(\mathrm{p},\omega)}\cos\theta\,\mathrm{d}\omega$$

$$E(\mathrm{p}) = \int_{H^2} L_i(\mathrm{p},\omega)\cos\theta\,\mathrm{d}\omega$$

Unit Hemisphere: H^2

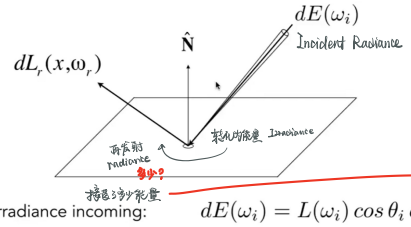


Bidirectional Reflectance Distribution Function BRDF

Reflection at a Point

Radiance from direction ω_i turns into the power E that dA receives

Then power E will become the radiance to any other direction ω_o



Differential irradiance incoming: $dE(\omega_i) = L(\omega_i)\cos\theta_i\, d\omega_i$

Differential radiance exiting (due to $dE(\omega_i)$): $dL_r(\omega_r)$

平面内某一区域受到光照后

向四面八方反射光

BRDF 尝试描述每个方向上

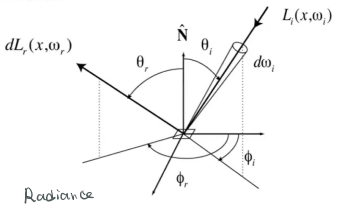
能量反射的不同

对任一出射方向算出 radiance

除以该小区域收到的 Irradiance

The Bidirectional Reflectance Distribution Function (BRDF)

represents how much light is reflected into each outgoing direction ω_r from each incoming direction



$$f_r(\omega_i \to \omega_r) = \frac{dL_r(\omega_r)}{dE_i(\omega_i)} = \frac{dL_r(\omega_r)}{L_i(\omega_i)\cos\theta_i\, d\omega_i} \quad \left[\frac{1}{\text{sr}}\right]$$

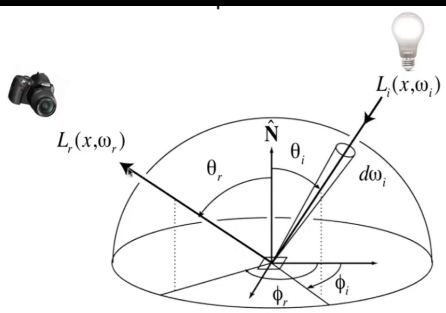
Radiance / Irradiance

定义材质

BRDF 定义该比例分配方式 <能量分布>

镜面 → 全部

漫反射 → 均分方向



$$L_r(\mathrm{p},\omega_r) = \int_{H^2} f_r(\mathrm{p},\omega_i \to \omega_r)\, L_i(\mathrm{p},\omega_i)\, \cos\theta_i\, d\omega_i$$

BRDF * 区域 Irradiance = 出射 radiance

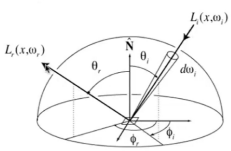
全部方向

积分 → 每个方向上的出射

Challenge: Recursive Equation

Reflected radiance depends on incoming radiance

$$\boxed{L_r(\mathrm{p},\omega_r)} = \int_{H^2} f_r(\mathrm{p},\omega_i \to \omega_r)\boxed{L_i(\mathrm{p},\omega_i)}\cos\theta_i\, d\omega_i$$



But incoming radiance depends on reflected radiance (at another point in the scene)

BRDF 渲染方程

Re-write the reflection equation:

$$L_r(\mathrm{p},\omega_r) = \int_{H^2} f_r(\mathrm{p},\omega_i \to \omega_r)\, L_i(\mathrm{p},\omega_i)\, \cos\theta_i\, d\omega_i$$

by adding an Emission term to make it general!

The Rendering Equation

若物体自身也发光

Input BRDF Weakening factor

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

自发光 半球 Hemisphere

Coming inward

Note: now, we assume that all directions are pointing outwards!

The rendering equation may be written in the form

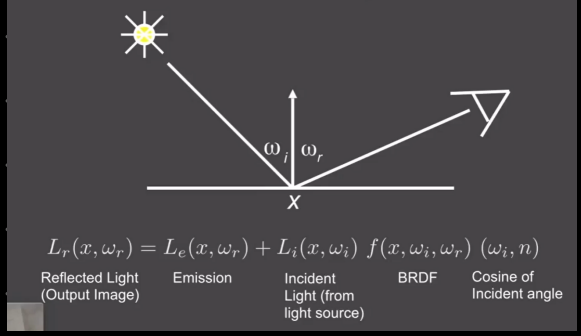
$$L_o(\mathbf{x},\omega_o,\lambda,t) = L_e(\mathbf{x},\omega_o,\lambda,t) + \int_\Omega f_r(\mathbf{x},\omega_i,\omega_o,\lambda,t) L_i(\mathbf{x},\omega_i,\lambda,t)(\omega_i\cdot\mathbf{n})\, d\omega_i$$

where

- $L_o(\mathbf{x},\omega_o,\lambda,t)$ is the total spectral radiance of wavelength λ directed outward along direction ω_o at time t, from a particular position $\mathbf{x}$
- $\mathbf{x}$ is the location in space
- ω_o is the direction of the outgoing light
- λ is a particular wavelength of light
- t is time
- $L_e(\mathbf{x},\omega_o,\lambda,t)$ is emitted spectral radiance 自发光
- $\int_\Omega \dots d\omega_i$ is an integral over Ω 半球范围
- Ω is the unit hemisphere centered around $\mathbf{n}$ containing all possible values for ω_i
- $f_r(\mathbf{x},\omega_i,\omega_o,\lambda,t)$ is the bidirectional reflectance distribution function, the proportion of light reflected from ω_i to ω_o at position $\mathbf{x}$, time t, and at wavelength λ
- ω_i is the negative direction of the incoming light
- $L_i(\mathbf{x},\omega_i,\lambda,t)$ is spectral radiance of wavelength λ coming inward toward $\mathbf{x}$ from direction ω_i at time t
- $\mathbf{n}$ is the surface normal at $\mathbf{x}$
- $\omega_i \cdot \mathbf{n}$ is the weakening factor of outward irradiance due to incident angle, as the light flux is smeared across a surface whose area is larger than the projected area perpendicular to the ray. This is often written as $\cos\theta_i$.

BRDF in single point light source

Reflection Equation

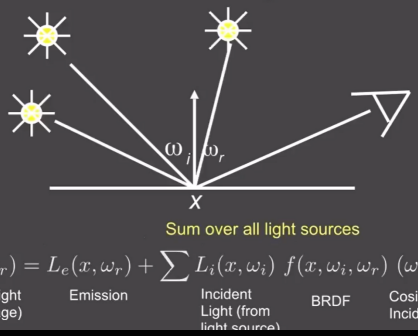


$$L_r(x,\omega_r) = L_e(x,\omega_r) + L_i(x,\omega_i)\ f(x,\omega_i,\omega_r)\ (\omega_i, n)$$

Reflected Light (Output Image) | Emission | Incident Light (from light source) | BRDF | Cosine of Incident angle

multiple light source

Reflection Equation



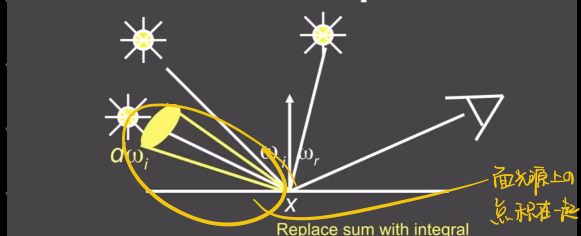
Sum over all light sources

$$L_r(x,\omega_r) = L_e(x,\omega_r) + \sum L_i(x,\omega_i)\ f(x,\omega_i,\omega_r)\ (\omega_i, n)$$

Reflected Light | Emission | Incident Light (from light source) | BRDF | Cosine of Incident angle

with surface light source

Reflection Equation



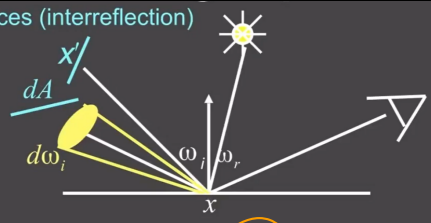
Replace sum with integral

$$L_r(x,\omega_r) = L_e(x,\omega_r) + \int_\Omega L_i(x,\omega_i)\ f(x,\omega_i,\omega_r)\ \cos\theta_i\, d\omega_i$$

Reflected Light | Emission | Incident Light (from light source) | BRDF | Cosine of Incident angle

considering inter-reflection 其实物体反射光来

Surfaces (interreflection)



$$L_r(x,\omega_r) = L_e(x,\omega_r) + \int_\Omega L_r(x',-\omega_i) f(x,\omega_i,\omega_r) \cos\theta_i d\omega_i$$

Reflected Light (Output Image)	Emission	Reflected Light	BRDF	Cosine of Incident angle
UNKNOWN	KNOWN	UNKNOWN	KNOWN	KNOWN

将反射面当作面光源

Rendering Equation as Integral Equation

$$L_r(x,\omega_r) = L_e(x,\omega_r) + \int_\Omega L_r(x',-\omega_i)\ f(x,\omega_i,\omega_r)\cos\theta_i d\omega_i$$

Reflected Light (Output Image)	Emission	Reflected Light	BRDF	Cosine of Incident angle
UNKNOWN	KNOWN	UNKNOWN	KNOWN	KNOWN

Is a Fredholm Integral Equation of second kind [extensively studied numerically] with canonical form

简写

过程<让BB>您啊

$$I(u) = e(u) + \int I(v)\ K(u,v)\,dv$$

Kernel of equation

$$I(u) = e(u) + \int I(v) K(u,v)dv$$

Kernel of equation
Light Transport Operator

$$L = E + KL$$

Can be discretized to a simple matrix equation [or system of simultaneous linear equations] (L, E are vectors, K is the light transport matrix)

物景中所有 radiance = 所有直发光 + 所有反射光

Rendering Equation (Kajiya 86)

- General class numerical Monte Carlo methods
- Approximate set of all paths of light in scene

$$L = E + KL \quad L = IL \quad I = \text{identity matrix}$$
$$IL - KL = E$$
$$(I-K)L = E$$
$$L = (I-K)^{-1}E$$

Binomial Theorem

$$L = (I + K + K^2 + K^3 + \ldots)E$$
$$L = E + KE + K^2E + K^3E + \ldots$$

算子拥有类泰勒展开之性质

$$L = E + KE + K^2E + K^3E + \ldots$$

- Emission directly From light sources
- Direct Illumination on surfaces
- Indirect Illumination (One bounce indirect) [Mirrors, Refraction]
- (Two bounce indirect illum.)

反射模作为 BRDF * cosθ

$$L = E + KE + K^2E + K^3E + \ldots$$

- Emission directly From light sources
- Direct Illumination on surfaces
- Indirect Illumination (One bounce indirect) [Mirrors, Refraction]
- (Two bounce indirect illum.)

Shading in Rasterization

光没有来过此

光没过此

效果收敛至真实亮度 → 能量守恒

概率论复习

Random Variables

X random variable. Represents a distribution of potential values

$X \sim p(x)$ probability density function (PDF). Describes relative probability of a random process choosing value x

Example: uniform PDF: all values over a domain are equally likely

e.g. A six-sided die

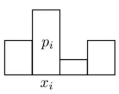
X takes on values 1, 2, 3, 4, 5, 6

$p(1) = p(2) = p(3) = p(4) = p(5) = p(6)$

Probabilities

n discrete values x_i

With probability p_i



Requirements of a probability distribution:

$p_i \geq 0$

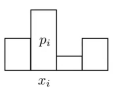
$\sum_{i=1}^{n} p_i = 1$

Six-sided die example: $p_i = \frac{1}{6}$

Expected Value of a Random Variable

The average value that one obtains if repeatedly drawing samples from the random distribution.

X drawn from distribution with
n discrete values x_i
with probabilities p_i



Expected value of X: $E[X] = \sum_{i=1}^{n} x_i p_i$

Die example: $E[X] = \sum_{i=1}^{n} \frac{i}{6}$

$= (1+2+3+4+5+6)/6 = 3.5$

Continuous Case: Probability Distribution Function (PDF)

概率密度函数

$X \sim p(x)$

A random variable X that can take any of a continuous set of values, where the relative probability of a particular value is given by a continuous probability density function p(x).

Conditions on p(x): $p(x) \geq 0$ and $\int p(x)\,dx = 1$

Expected value of X: $E[X] = \int x\,p(x)\,dx$

Function of a Random Variable

A function Y of a random variable X is also a random variable:

$X \sim p(x)$

$Y = f(X)$

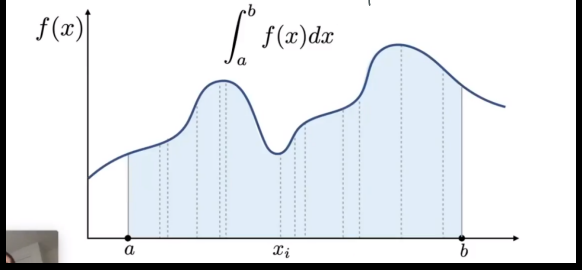
Expected value of a function of a random variable:

$E[Y] = E[f(X)] = \int f(x)\,p(x)\,dx$

后续用蒙特卡洛积分

Monte Carlo Integration

Why: we want to solve an integral, but it can be too difficult to solve analytically. 计算定积分 Averaging random samples of the function value.

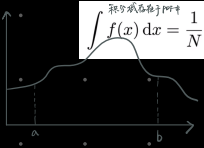


① Definite Integral $\int_a^b f(x)\,dx$

② Random Variable $X_i \sim p(x)$

③ Monte Carlo Estimator $F_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)}$

$$\int f(x)\,dx = \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)} \qquad X_i \sim p(x)$$



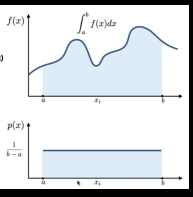
Uniform random variable

$X_i \sim p(x) = C$ (constant)

$\int_a^b p(x)\,dx = 1$

$\Longrightarrow \int_a^b C\,dx = 1$

$\Longrightarrow C = \frac{1}{b-a}$



Definite integral $\int_a^b f(x)dx$

Uniform random variable $X_i \sim p(x) = \frac{1}{b-a}$

Basic Monte Carlo estimator $F_N = \frac{b-a}{N}\sum_{i=1}^{N} f(X_i)$

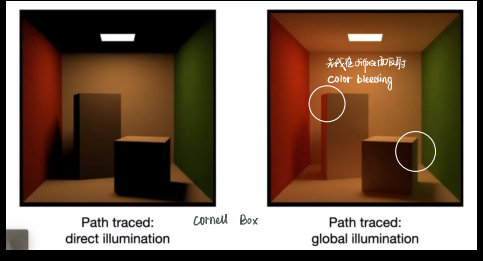
path Tracing

motivation: whitted-style ray-tracing

① always perform specular reflections / refractions

② stop bouncing at diffuse surfaces

Mirror reflection — **Glossy** reflection



Path traced: direct illumination — Cornell Box — Path traced: global illumination

But the rendering equation is correct

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

$$L_r(x,\omega_r) = L_e(x,\omega_r) + \int_{\Omega} L_r(x',-\omega_i) f(x,\omega_i,\omega_r)\cos\theta_i\,d\omega_i$$

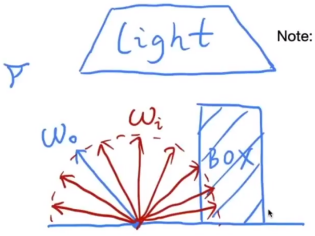
Reflected Light (Output Image)	Emission	Reflected Light	BRDF	Cosine of Incident angle
UNKNOWN	KNOWN	UNKNOWN	KNOWN	KNOWN

① Solving integral with hemisphere ② Recursive execution

approach 使用蒙特卡洛

① 仅考虑直接光照

Suppose we want to render **one pixel (point)** in the following scene for **direct illumination** only



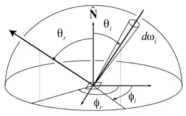
Abuse the concept of Reflection Equation a little bit

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

(again, we assume all directions are pointing outwards)

Fancy as it is, it's still just an integration over directions

So, of course we can solve it using Monte Carlo integration!



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

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

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

(assume uniformly sampling the hemisphere)

So, in general

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

$$\approx \frac{1}{N}\sum_{i=1}^{N}\frac{L_i(p,\omega_i) f_r(p,\omega_i,\omega_o)(n\cdot\omega_i)}{p(\omega_i)}$$

(note: abuse notation a little bit for i)

What does it mean?

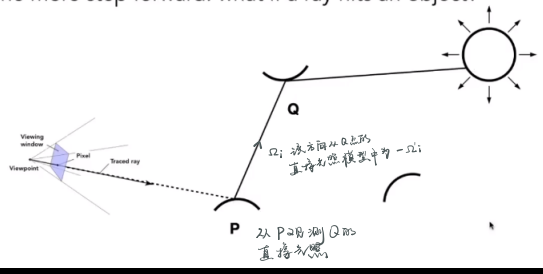
A correct shading algorithm for direct illumination!

$$L_o(p,\omega_o) \approx \frac{1}{N}\sum_{i=1}^{N}\frac{L_i(p,\omega_i) f_r(p,\omega_i,\omega_o)(n\cdot\omega_i)}{p(\omega_i)}$$

```
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)
    Return Lo
```

② 考虑间接光照 Global Illumination

One more step forward: what if a ray hits an object?

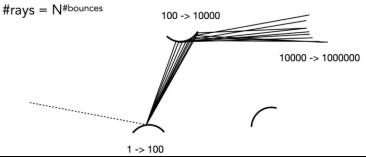



```
shade(p, wo)
    Randomly choose N directions wi~pdf
    Lo = 0.0
    For each wi
        Trace a ray r(p, wi)
        If ray r hit the light
            Lo += (1 / N) * L_i * f_r * cosine / pdf(wi)
        Else If ray r hit an object at q
            Lo += (1 / N) * shade(q, -wi) * f_r * cosine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

存在的问题

- 指数级增长的弹射

Problem 1: Explosion of #rays as #bounces go up:



key observation: #rays will not explode iff N = 1

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)
```

N=1 蒙特卡洛积分 = path tracing

⚠ N≠1 ⇨ distributed ray tracing

- 无限递归

但直接照到弹射次数是不对的 → 损失能量（弹射次数）

两难？

把真效果 ⟷ 算力

Solution: Russian Roulette (RR)
(俄罗斯轮盘赌)

Russian Roulette is all about probability
With probability 0 < P < 1, you are fine
With probability 1 - P, otherwise

Previously, we always shoot a ray at a shading point and get the shading result **Lo**

Suppose we manually set a probability P (0 < P < 1)
With probability P, shoot a ray and return the **shading result divided by P**: Lo / P
With probability 1-P, don't shoot a ray and you'll get 0

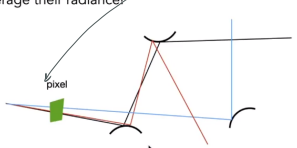
In this way, you can still **expect** to get Lo!:
E = P * (Lo / P) + (1 - P) * 0 = Lo

```
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
```

Ray Generation

But this will be noisy!

No problem, just trace more **paths** through each pixel and average their radiance!



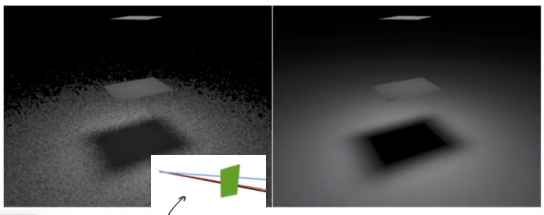
Very similar to ray casting in ray tracing

```
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
```

③

Now we already have a **correct** version of path tracing!

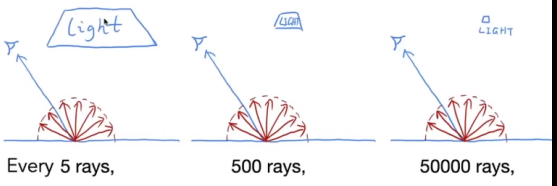
But it's **not really efficient**.



Low SPP (samples per pixel) noisy results

High SPP

Understanding the reason of being inefficient



能否打到光源 → 看运气

There will be 1 ray hitting the light, a lot of rays are "wasted" as uniform sampling

Sampling the Light (pure math)

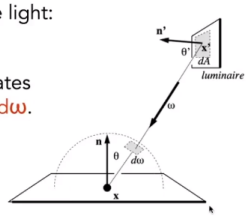
Monte Carlo methods allows any sampling methods, so we can sample the light (therefore no rays are "wasted")

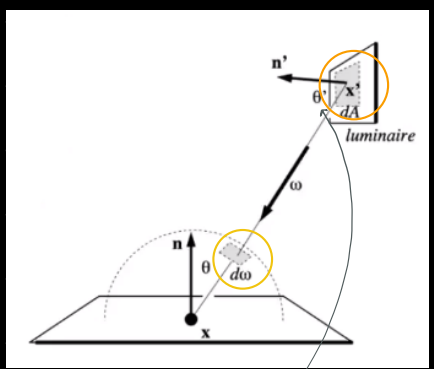
Assume uniformly sampling on the light:

pdf = 1 / A (because ∫pdf dA = 1)

But the rendering equation integrates on the solid angle: Lo = ∫Li fr cos dω.

Recall Monte Carlo Integration: Sample on x & integrate on x





① 在光源上采样 <面积>

矛盾，对蒙特卡洛方程的

② 渲染方程应用于半球，在立体角上积分

△ 积分渲染方程 → 在光源上积分，改变积分域

$$dw = \frac{dA \cos\theta'}{\|x' - x\|^2}$$

立体角定义 投影面积比

距离

$$L_o(x, \omega_o) = \int_{\Omega^+} L_i(x, \omega_i) f_r(x, \omega_i, \omega_o) \cos\theta \, d\omega_i$$

$$= \int_A L_i(x, \omega_i) f_r(x, \omega_i, \omega_o) \frac{\cos\theta \cos\theta'}{\|x' - x\|^2} \, dA$$

A

f(x)

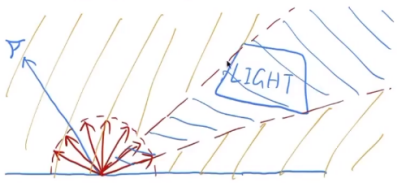
$p(x) = \frac{1}{A}$

Sampling the Light

Previously, we assume the light is "accidentally" shot by uniform hemisphere sampling

Now we consider the radiance coming from two parts:

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



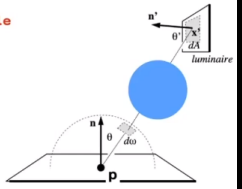
△ 点光源是一个难题，因此建议使用小面光源

```
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
```

One final thing: how do we know if the sample on the light is not blocked or not?

```
# Contribution from the light source.
L_dir = 0.0
Uniformly sample the light at x' (pdf_light = 1 / A)
Shoot a ray from p to x'
If the ray is not blocked in the middle
    L_dir = …
```

Now path tracing is finally done!



Topics we haven't cover

Ray tracing: Previous vs. Modern Concepts

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

- 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)
- Do random numbers matter?
  - Yes! (low discrepancy sequences)

采样 无向权重

- 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, curves, color space)

是在累加和

radience 换算 color 之问题