

第三本书: PDF method

PDF 全称为 probability density function 概率密度函数, 表示一个连续随机变量在某点附近出现的概率。对于一个区间 $[l, r]$, 变量落在这个区间内的概率为 $\int_l^r p(x)dx$ 。这是一个在高中统计学就接触过的概念。现在我们将这个概念用于计算积分值和光线追踪中。

基本性质

设概率密度函数为 $p(x)$, 它的反导数为 $P(x)$, 则

$$\int_{-\infty}^{\infty} p(x)dx = P(x)|_{-\infty}^{\infty} = 1$$

因为所有变量出现概率相加一定是 1。

运用均匀随机数生成符合特定概率密度函数的随机变量

假设 `rand()` 函数能够均匀随机产生 $[0, 1]$ 区间内的实数, 再给定一个概率密度函数 $p(x)$, 如何产生满足 $p(x)$ 的随机变量?

注意到 `rand()` 函数的值域是 $[0, 1]$, 我们可以把这个函数的意义理解成: 如果生成了 $x \in [0, 1]$, 这意味着生成比 x 小的数的概率是 x , 那么我们应该建立一个映射 $g: x \mapsto y$, 使得生成的满足该函数的随机变量小于 y 的概率是 x 。也就是 $P(x)|_{-\infty}^y = x$ 。假设这样的 x 有界, 那么 $P(x)|_{-\infty} = 0$, 即 $P(y) = x$ 。则 $y = g(x) = P^{-1}(x)$ 。

用 PDF 估算积分值

我们可以运用它来估算积分值。 $\int_a^b f(x)dx$ 可以看成 $(b-a) \cdot \text{average}(f(x))$, 其中 `average` 可以通过采很多次样, 对结果求和后除以采样次数的方法求出。但是有一个问题: 在采非常多次样之后, 和一定会收敛, 但是如果我们想用尽量少的采样次数让和收敛更快呢? 我们希望函数值 (的绝对值) 大的时候采样次数多一些, 函数值小的时候可以少采样, 这样收敛就会更快。这时 pdf 就要派上用场了。然而之前的平均数需要改成加权平均。假设我们按照原来的方法均匀采 n 次样, 那么积分可以表示成

$$\sum \frac{f(x_i)}{\frac{n}{b-a}}$$

联想到均匀采样时 $p(x) \equiv \frac{1}{b-a}$, 那么积分值可以表示成

$$\frac{1}{n} \sum \frac{f(x_i)}{p(x_i)}$$

PDF 在光线追踪中的应用

说了这么多数学方法, 现在我们来考察它在光线追踪的应用。

本书中经常用到曲面上的积分, 一般默认为第一类曲面积分 (后面跟一个 $dA = \sin \theta d\theta d\varphi$) (书上的 θ 和 φ 跟平常用法是反过来的)

我们考察前面提到的得出散射光线颜色的方法。传统的方法会用到随机向量来表示散射光线。现在我们希望用一个概率密度函数来表征各个方向上散射光线出现的概率, 我们用 $s(\text{direction}): \mathbb{R}^3 \rightarrow \mathbb{R}$ 来表示它。然后, 每次散射都会使得 R, G, B 的光强按一定比例减弱, 这就是之前的 `attenuation`。对于每个方向, 我们也要考察这个方向的 `ray_color`, 相当于递归调用。将连续求和表达成积分形式, 得到

$$Color = A \int s(\text{direction}) \cdot \text{color}(\text{direction})$$

联系到上述估算积分的方法，得到

$$Color = A \sum \frac{s(\text{direction}) \cdot \text{color}(\text{direction})}{p(\text{direction})}$$

Lambertian 的 scattering pdf

lambertian 的散射结果是由 `normal + random_unit_vector` 生成的。我们给出 lambertian 材料的散射概率密度函数 $s(\theta, \varphi) \propto \cos \theta$ 。

我们对它做第一类曲面积分：

$$\int \cos \theta dA = \int_{[0, \pi] \times [0, 2\pi]} \cos \theta \sin \theta d\theta d\varphi = \pi$$

$$\text{则 } s(\theta, \varphi) = \frac{1}{\pi} \cos \theta.$$

如果我们取 $p(\theta) = s(\theta)$ ，那么 $Color = A \sum \text{color}(\text{direction})$ ，跟未采用 pdf 的方法无异。但是我们还是希望能够给更重要的方向（例如，指向光源）更多权重，从而减小噪点。

反射光的分布可以用 BRDF (bidirectional reflectance direction function) $\frac{A \cdot s(\text{direction})}{\cos(\theta)}$ 表示，则对于 lambertian 材料， $BRDF = \frac{A}{\pi}$ 。

Code:

```
// in func. ray_color
let mut pdf = 0.0;
let mut albedo: color3 = color3::new();
if !rec
    .mat_ptr
    .as_ref()
    .unwrap()
    .scatter(r, &rec, &mut albedo, &mut scattered, &mut pdf)
{
    return emitted;
}
emitted
    + albedo
      * rec
        .mat_ptr
        .as_ref()
        .unwrap()
        .scattering_pdf(r, &rec, &scattered)
      * ray_color(&scattered, background, world, depth - 1)
    / pdf

// in impl Material for Lambertian
fn scatter(
    &self,
    r_in: &Ray,
    rec: &HitRecord,
    alb: &mut color3,
    scattered: &mut Ray,
```

```

pdf: &mut f64,
) -> bool {
    let direction = random_in_hemisphere(&rec.normal);
    *scattered = Ray::construct(&rec.p, &direction.unit(), r_in.time());
    *alb = self.albedo.value(rec.u, rec.v, &rec.p);
    *pdf = 0.5 / PI;
    true
}

fn scattering_pdf(&self, _r_in: &Ray, rec: &HitRecord, scattered: &Ray) ->
f64 {
    let cosine: f64 = dot(&rec.normal, &scattered.direction().unit());
    // 即为求 pdf, 上文已提到 cos theta / pi
    if cosine < 0.0 {
        0.0
    } else {
        cosine / PI
    }
}

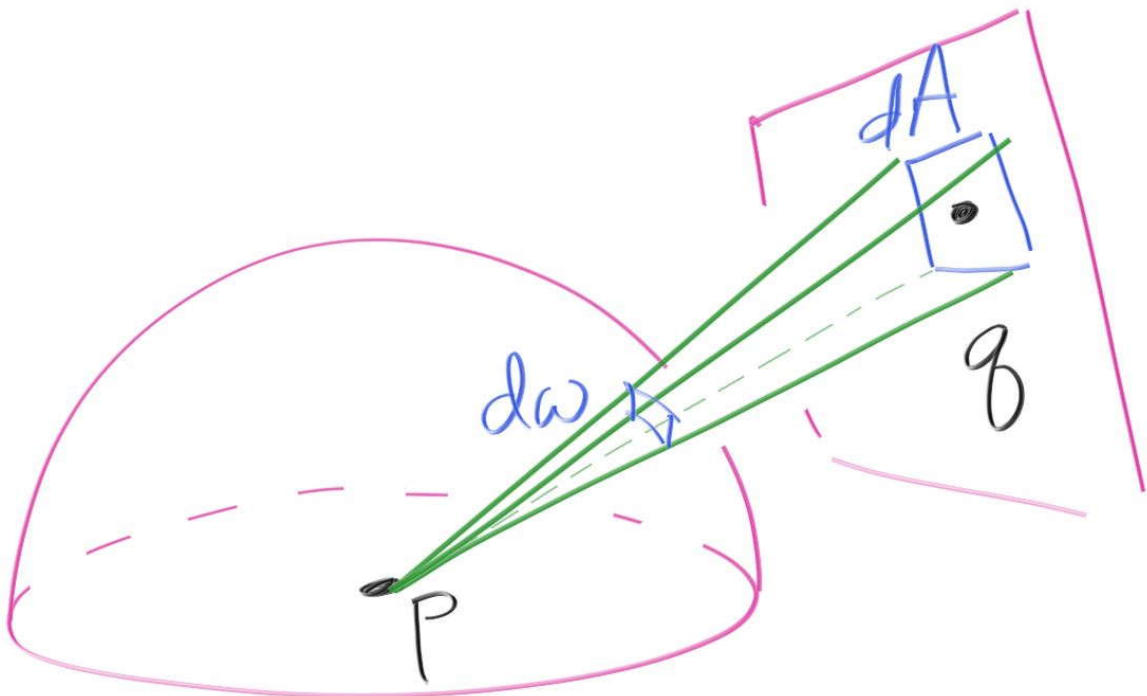
```

对光源采样以及混合 PDF

注意到将多个概率密度函数线性组合后，它依然是一个合法的 pdf，所以我们可以采用以下策略：

总的 pdf 由 $\frac{1}{2}$ 的对向光源的 pdf 与 $\frac{1}{2}$ 的与上文的 $s(\text{direction})$ pdf 相加得到。下面我们依然以 lambertian 为例求出对光源采样的 pdf，从而得到混合 pdf。

这里的光源是一个矩形，但是我们要生成一个与 θ, φ 相关的 pdf，因此我们需要转化一下。



在这个矩形中打到 dA 和在（半）球上打到 dw 的概率是一样的，又

$$dw = \frac{dA \cdot \cos(\alpha)}{|PQ|}$$

其中 α 为光线方向与发光面法向量（这里是 y 轴方向）的夹角。

又

$$p(\text{direction})dw = p_q(q)dA = \frac{dA}{A}$$

得

$$p(\text{direction}) = \frac{|PQ|}{A \cos \alpha}$$

参考资料

[*Ray Tracing: The Next Week*](#)

[*Ray Tracing: The Rest of Your Life*](#)