

# The PDF Method In Book3

李紫燕

2023 年 7 月 20 日

我们在前两本书中编写的随机程序存在一个问题，即小光源会产生过多的噪声。这是因为我们的均匀采样无法充分采样这些小光源。只有当光线向它们散射时，才对光源进行采样，但对于小光源或远离的光源来说，这种情况可能不太常见。如果我们向这些光源发送更多的随机样本，可以减轻这个问题，但这会导致场景过亮。为了消除这种不准确性，我们可以通过调整采样样本的权重来解决。那么如何进行这种调整呢？为了做到这一点，我们需要引入概率密度函数 (PDF) 和蒙特卡洛积分估计 (MC) 的概念。

## 1 数学原理

PDF：指概率密度函数，是用来描述连续型随机变量的输出值，在某个确定的取值点附近的可能性的函数。

CDF：指累积分布函数，又叫分布函数，是概率密度函数 (PDF) 的积分，用来表示离散型随机变量  $x$  的概率分布。

$$P(x) = \text{area}(p, -\infty, x)$$

$f(x)$ ：通过  $P(x)$  可以算出不同  $x$  对应的概率，现在我们需要一个函数  $f(x) = P^{-1}(x)$  也就是  $P(x)$  的逆函数，据此可以求出不同概率下对应的数值。

MC：它是一种数值计算方法，利用随机抽样和统计分析来解决问题，而不依赖于解析的公式或确定性的算法。通过在问题的定义域上进行随机抽样，可以将问题转化为统计推断或分布分析的过程。

在 MC 中，PDF 在估计统计量时起着重要作用。PDF 描述了随机变量在不同取值上的概率分布情况。通过使用 PDF，我们可以根据概率对样本进行加权，从而实现用更少的样本准确地估计积分、求解概率或其他统计量。

### 1.1 一维下的 PDF 与 MC

举例说明：已知被积函数为  $y = x^2$ ，有线性 PDF  $p(r) = r/2, 0 < x < 2$ ，对 PDF 进行积分则有 CDF：

$$P(x) = \frac{x^2}{4} : 0 < x < 2$$

由定义可以计算出：

$$f(x) = \sqrt{4x}, 0 < x < 1$$

于是，我们随机生成  $N=1000$  个

$$r = \sqrt{4 * random\_double(0, 1)}$$

运用 MC 积分方法：对  $\frac{f(r)}{p(r)}$  取平均值，得到积分值

$$I = \frac{1}{N} \sum_{i=1}^N \frac{f(x_i)}{p(x_i)}$$

现在来做总结：

1. 有一个对于  $f(x)$  在  $[a, b]$  范围内的积分
2. 选一个 PDF  $p$ ，其值在  $[a, b]$  内不为 0
3. 对于一堆  $\frac{f(r)}{p(r)}$  求平均， $r$  是符合 PDF  $p$  的随机数。

对于任意 PDF  $p$  最后都会收敛到正确答案，但是  $p$  越近似  $f$  则收敛的越快。

## 1.2 在方向球面上的 PDF 与 MC

举例说明： $\int \cos^2(\theta)$ ，运用 MC，我们需要计算

$$\frac{\cos^2 \theta}{p(direction)}$$

此时可以用  $random\_in\_unit\_sphere().z()$  计算出  $\cos(\theta)$ ，并且在均匀球体上，可以令  $p(direction) = 1/area = 1/4 * \pi$

## 2 RayTracing 中的实际应用

### 2.1 Light Scattering

如果光发生散射，它将具有方向分布，我们可以将其描述为对固体角度的概率密度函数，这里我们将其称为散射概率密度函数： $s(direction)$ ，于是物体表面的颜色：

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

其中  $A$  和  $s()$  与视线方向和散射位置有关。应用 MC 方法后有：

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

其中  $p(direction)$  是我们随机生成的任意方向的 pdf.

已知  $dA = \sin(\theta)d\theta d\phi$ ， $s()$  正比于  $\cos(\theta)$ ， $\int s()dA = 1$ ，于是有

$$Area = \int_0^{2\pi} \int_0^{\pi/2} \cos(\theta) \sin(\theta) d\theta d\phi = 2\pi \frac{1}{2} = \pi$$

所以：

$$s(direction) = \frac{\cos(\theta)}{\pi}$$

## 2.2 Generating Random Directions

现在我们需要随机生成任意方向的 pdf, 即  $p(direction)$ .

为了简化问题, 我们假设  $z$  轴是表面法线,  $\theta$  是与法线的夹角。我们只需处理关于  $z$  轴旋转对称的分布, 且有  $p(direction) = f(\theta)$ ,  $dA = \sin(\theta) \cdot d\theta \cdot d\phi$ , 于是：

$$a(\phi) = \frac{1}{2\pi}, b(\theta) = 2\pi f(\theta) \sin(\theta)$$

$$r_1 = \int_0^\phi \frac{1}{2\pi} dt = \frac{\phi}{2\pi}, \phi = 2\pi \cdot r_1$$

$$r_2 = \int_0^\theta 2\pi f(t) \sin(t) dt$$

通过上面的 PDF 方法, 取  $p(directions) = \cos(\theta)/\pi$ ,  $r_1$ 、 $r_2$  为随机生成的 (0,1) 的实数, 我们可以得到：

$$z = \cos(\theta) = \sqrt{1 - r_2}$$

$$x = \cos(\phi) \sin(\theta) = \cos(2\pi r_1) \sqrt{1 - z^2} = \cos(2\pi r_1) \sqrt{r_2}$$

$$y = \sin(\phi) \sin(\theta) = \sin(2\pi r_1) \sqrt{1 - z^2} = \sin(2\pi r_1) \sqrt{r_2}$$

至此, 我们能够相对于  $Z$  轴生成随机方向的向量。现在希望能够相对于表面法线向量来生成这样的随机方向向量, 于是我们构建了 `onb` 类, 通过 `build_from_w()` 和 `local()` 两个函数实现上述功能。

## 2.3 Sampling Lights Directly

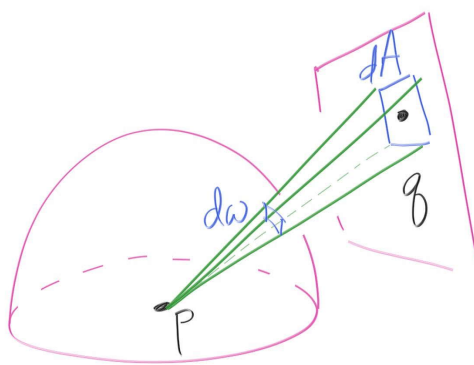


图 1: 方形光源

光源对图片噪声的影响是很大的，由于光源的形状不同，发光情况不同，所以对于每一个作为光源的 Hittable 物体都要实现 pdf\_value() 和 random() 两个函数，分别用来分配 pdf 权重和产生随机采样光线。例如，对于方形的光源：(如图 1)

$$dw = \frac{dA \cdot \cos(\alpha)}{\text{distance}^2(p, q)}$$

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

所以:

$$p(\text{direction}) \cdot \frac{dA \cdot \cos(\alpha)}{\text{distance}^2(p, q)} = p_q(q) \cdot dA = \frac{dA}{A}$$

于是:

$$p(\text{direction}) = \frac{\text{distance}^2(p, q)}{\cos(\alpha) \cdot A}$$

## 2.4 Mixture Densities

我们已经使用了一个与  $\cos(\theta)$  有关的 PDF 和一个与光源有关的 PDF，现在我们把两个 PDF 类结合起来。在我们的 MIXTURE\_PDF 类中，数据成员为两位的 PDF 数组，成员函数 value() 用于求出在当前位置的 value，成员函数 generate() 用于根据 PDF 生成任意方向的向量，这里贴出代码实现：

```
pub struct MixturePdf {
    pub p: [Arc<dyn Pdf>; 2],
}
impl MixturePdf {
    pub fn new(p0: Arc<dyn Pdf>, p1: Arc<dyn Pdf>) -> Self {
        Self { p: [p0, p1] }
    }
}
impl Pdf for MixturePdf {
    fn generate(&self) -> Vect3 {
        if random_double() < 0.5 {
            self.p[0].generate()
        } else {
            self.p[1].generate()
        }
    }
    fn value(&self, direction: Vect3) -> f64 {
        0.5 * self.p[0].value(direction) + 0.5 * self.p[1].value(direction)
    }
}
```

## 2.5 Conclusion

PDF 的作用，在于巧妙地分配采样权重和概率，在采样数相同的情况下实现了更好的渲染效果。Book3 实现了三种 pdf:

1.cosin\_pdf: 光线和物体相交时，材质的属性通常会影响光的传播方式。也就是说，物体材质不同，对应的 pdf 不同，这里的 cosin\_pdf 为材质 Lambertian 实现了完美的漫反射

2.hittable\_pdf: 对不同的光源实现 pdf, 有效采样重点光源，减少噪点。

3.mixture\_pdf: 前面两个 pdf 的结合，采样权重分配更加合理