# 无限面光源

Dezeming Family

2022 年 3 月 27 日

DezemingFamily 系列书和小册子因为是电子书，所以可以很方便地进行修改和重新发布。如果您获得了 DezemingFamily 的系列书，可以从我们的网站 [https://dezeming.top/] 找到最新版。对书的内容建议和出现的错误欢迎在网站留言。

## 目录

# 一　PBRT 无限面光源

由于最近确实又需要再实现一下无限面光源，以前的实战篇里没有讲解原理，只是告诉了怎么移植，在这里我就将 PBRT 里的无限面光源进行一下原理上的详细介绍。

这里，基于重要性采样策略的无限面光源采样步骤如下：

- 在图像 $(u, v)$ 中定义一个分段常量 2D 概率密度函数，对应概率密度函数 $p(u, v)$。
- 应用对 2D 样本采样的方式来采样。
- 对 $(u, v)$ 相应地定义一个球面概率密度函数（因为无线面光源是一个大的球面）。

这三个步骤的结合，使得可以在方向球上生成与辐射度函数（无限环境光的辐射度）非常接近的分布的样本，从而大幅减少方差。

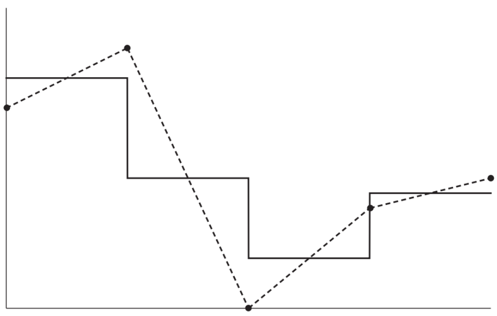
整个过程分为两步：

- 从环境光 map 中计算标量值图像 img。
- 计算图像行列的采样分布。

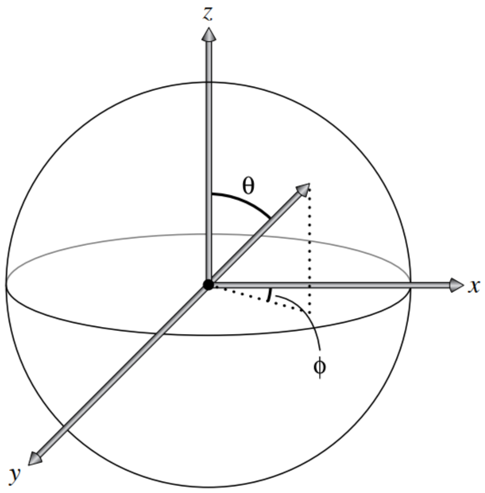
第一步会把环境光 map 定义为分段常量标量函数。首先，会生成一个与 map 同样长宽的 img 图，通过 y() 函数计算亮度 (luminance)；这个 img 图的像素值是相邻的两个像素插值得到的（见下面的程序，对于 $(u, v)$，其坐标为 $(u + .5, v + .5)$，用 Lookup 函数来访问获得）：

```
    for (int v = 0; v < height; v++) {
        float vp = (v + .5f) / (float)height;
        for (int u = 0; u < width; ++u) {
            float up = (u + .5f) / (float)width;
            img[u + v * width] = ......
        }
    }
```

这是为了插值得到更平滑的值（当某个像素为纯黑时，周边像素可能不是黑的），1D 情况类似于下图：



其次，img 映射到球面时，需要乘以 $\sin\theta$ 值（$\theta$ 值在区间 $[0, \pi]$ 之间），这种相乘的意义在于减少“扭曲”。，因为球面映射到 2D 全景图时，纬度越高的地方单个微元映射的区域就越大。纬度为 0 度的线设长度为 $l_0$；纬度为 30 度的线就是 $\cos(\pi/6)$。注意纬度的表示和 $\theta$ 是有区别的：




所以要除以 $\sin\theta$ 我们后面也会详细介绍关于 sin 和采样概率的问题。

之后，就是生成一个二维概率密度分布。我们下一节先介绍如何计算和生成二维分布，然后下下一节我们将介绍如何采样。

# 二 生成二维 Pdf 分布

一般来说，我们所使用的概率密度 $p(x,y)$ 中的 $x$ 和 $y$ 都不是独立的，所以无法转换为 $p(x)\cdot p(y)$。定义边缘密度函数（$p(x)$）和条件密度函数（$p(y|x)$）。

$$p(x) = \int p(x,y)dy$$
$$p(y|x) = \frac{p(x,y)}{p(x)} \tag{二.1}$$

从联合概率密度分布进行 2D 采样的方法，就是首先计算单个变量边缘概率密度，然后根据边缘概率密度函数抽取一个样本；一旦提取了样本，就可以计算给定该值的条件密度函数，并从该分布中再次使用 1D 采样来抽取一个样本。

我们考虑自变量定义在 $(u,v)\in[0,1]^2$ 范围内的 2D 数组，给定一个函数 $f(u,v)$，该函数的离散形式 $f[u_i,v_j]$ 自变量一共有 $n_u\times n_v$ 个取值。也就是说，$u_i\in[0,1,...,n_u-1]$，$v_j\in[0,1,...,n_v-1]$。由于是二维的函数，因此相当于 $f[u_i,v_j]$ 表示 $f(u,v)$ 在区间 $[\frac{i}{n_u},\frac{i+1}{n_u}]\times[\frac{j}{n_v},\frac{j+1}{n_v}]$。

为了方便表述，给定连续的 $(u,v)$，我们使用 $(\tilde{u},\tilde{v})$ 来表示对应的 $f[u_i,v_j]$ 值。

积分值为：

$$I_f = \int\int f(u,v)\mathbf{d}u\mathbf{d}v = \frac{1}{n_u n_v}\sum_{i=0}^{n_u-1}\sum_{j=0}^{n_v-1} f[u_i,v_j] \tag{二.2}$$

上式中，之所以右边要除以 $n_u n_v$，是因为定义区间在 $[0,1]$ 之间，二维离散函数可以看做一个一个的小格子，每个格子的长度是 $\frac{1}{n_u}$，宽度是 $\frac{1}{n_v}$。

因此 pdf 就这么求：

$$p(u,v) = \frac{f(u,v)}{\int\int f(u,v)dudv} = \frac{f[\tilde{u},\tilde{v}]}{\frac{1}{n_u n_v}\sum_{i=0}^{n_u-1}\sum_{j=0}^{n_v-1} f[u_i,v_j]}$$
$$p(v) = \int p(u,v)\mathbf{d}u = \frac{\frac{1}{n_u}\sum_{i=0}^{n_u-1} f[u_i,\tilde{v}]}{I_f}$$
$$p(u|v) = \frac{p(u,v)}{p(v)} = \frac{f[\tilde{u},\tilde{v}]/I_f}{p[\tilde{v}]} \tag{二.3}$$

因此，$p(\tilde{u}|\tilde{v})$ 是 $\tilde{u}$ 的分段常量 1 维函数。

class Distribution2D 的构造函数中，（1）首先会计算 1 维 pdf 函数 $p(\tilde{u}|\tilde{v})$（就是对每一行，对该行的每个元素做采样分布）；（2）然后会计算边缘概率密度分布 $p[\tilde{v}]$。

（1）过程的代码是：

```
pConditionalV.reserve(nv);
    for (int v = 0; v < nv; ++v) {
        // Compute conditional sampling distribution for $\tilde{v}$
        pConditionalV.emplace_back(new Distribution1D(&func[v * nu],
            nu));
    }
```

构建 1D 分布也很简单：

```
cdf[0] = 0;
//计算生成cdf
    for (int i = 1; i < n + 1; ++i) cdf[i] = cdf[i - 1] + func[i - 1] /
        n;
funcInt = cdf[n];
//cdf归一化
for (int i = 1; i < n + 1; ++i) cdf[i] /= funcInt;
```

(2) 过程的代码是：

```
// Compute marginal sampling distribution $p[\tilde{v}]$
std::vector<float> marginalFunc;
marginalFunc.reserve(nv);
for (int v = 0; v < nv; ++v)
    marginalFunc.push_back(pConditionalV[v]->funcInt);
pMarginal.reset(new Distribution1D(&marginalFunc[0], nv));
```

注意此时并不需要归一化的 cdf（funcInt 是没有归一化前的 cdf[n]），因为 1D 的 pdf 函数生成时不需要归一化的值。

# 三　采样二维 Pdf 分布

采样方法很直观，见 SampleContinuous 函数，该函数会先采样 1 维的边缘密度，然后再从边缘密度中采样条件概率密度，之后相乘得到：

$$p(u,v) = p(v)p(u|v) \tag{三.1}$$

采样 1 维时，当给定一个随机数 $\xi$，其实就是找到其所在的 CDF 间隔，也就是 $CDF[i] \leq \xi \leq CDF[i+1]$，查找方式用对半搜索即可。之后会计算 pdf：func[offset] / funcInt。最后计算在 $[0,1]$ 内的偏移量。

# 四　采样无限面光源

采样无限面光源的过程可以分为下面四步：

(1) - 在无限面光源纹理上采样到 (u,v)

(2) - 把无限面光源样本点转到球面方向

(3) - 计算无限面光源采样的 PDF

(4) - 得到球面无限面光源方向的 radiance

第一步得到 $[0,1)^2$ 范围内的采样点：

```
Point2f uv = distribution->SampleContinuous(u, &mapPdf);
```

第二步，将该点映射到球面上（就是把全景图 $[\pi, 2\pi]$ 映射到球面方向）：

```
float theta = uv[1] * Pi, phi = uv[0] * 2 * Pi;
......
Vector3f(sinTheta * cosPhi, sinTheta * sinPhi, cosTheta);
```

第三步，计算 Pdf（会在下面进行介绍）：

```
*pdf = mapPdf / (2 * Pi * Pi * sinTheta);
```

第四步，生成可见性，返回 radiance：

```
*vis = VisibilityTester(ref, Interaction(ref.p + *wi * (2 * worldRadius), ref.time, mediumInterface));
 return Spectrum(Lmap->Lookup(uv), SpectrumType::Illuminant);
```

这里只有 Pdf 的计算是没有讲过的。我们再提一下为什么前面需要乘以 $sin\theta$。如果所有的 $(\theta, \phi)$ 的抽样概率相同（且 radiance 均匀分布），那么越靠近极点的位置就越容易被抽样到，因此乘以 $\sin\theta$，相当于越靠近极点的位置纬线越短，纬线上的点也不容易被采样到。

平面上的采样需要转换到球面上，而对于多维概率密度来说，转换的方法相当于除以变换的雅克比行列式（我们暂时不讲解它的原理，而是放在 Dezeming Family 的“随机过程”专栏里《概率密度分布的转换——从一维到多维》），因此得到：

$$
\begin{aligned}
g(u, v) &= (\pi v, 2\pi u) \\
p(\theta, \phi) &= \frac{p(u, v)}{2\pi^2}
\end{aligned}
$$

对于球面坐标立体角的采样：$d\omega = \sin\theta \mathbf{d}\theta \mathbf{d}\phi$（该内容也需要涉及一些 pdf 分布变换的知识，可见 [1] 第三版的 773 页）。因此，立体角的概率密度为：

$$
p(\omega) = \frac{p(\theta, \phi)}{\sin\theta} = \frac{p(u, v)}{2\pi^2 \sin\theta} \tag{四.1}
$$

## 参考文献


[1] Pharr M , Humphreys G . Physically Based Rendering: From Theory To Implementation. 3nd Edition. Morgan Kaufmann Publishers Inc. 2016.