

马尔科夫链蒙特卡洛方法

Dezeming Family

2022 年 12 月 7 日

DezemingFamily 系列文章和电子书**全部都有免费公开的电子版**，可以很方便地进行修改和重新发布。如果您获得了 DezemingFamily 的系列电子书，可以从我们的网站 [<https://dezeming.top/>] 找到最新的版本。对文章的内容建议和出现的错误也欢迎在网站留言。

本文是对 [1] 论文的详细解读，该论文可以说是渲染必读论文，但有些符号表示和描述可能对初学者并不友好，由于里面介绍的几种重要技术，例如双向路径追踪、多重重要性采样以及 Metropolis 方法都是非常重要的，因此我打算写一下本论文的解读，作为构建“高端图形渲染学习体系”的一个重要组成部分。

由于论文 [1] 篇幅过长，为了减少 Latex 编译的时间以及更好把控不同部分的内容，我将整个论文划分为了多本小册子来进行讲解。

本文的预备知识：**蒙特卡洛方法、蒙特卡洛光线追踪**（可以看 Peter Shirley 的光线追踪三本小书）、**BSDF 模型、路径追踪、向量空间**。

目录

一 基本介绍	1
二 马尔科夫链	1
2.1 直接采样的接受-拒绝法	1
2.2 马尔科夫链	1
2.3 平稳分布	2
三 Metropolis 算法	3
3.1 Metropolis 采样方法	3
3.2 变异策略	4
3.3 启动偏差	5
3.4 一个简单的实例	5
3.5 使用 Metropolis 方法估计积分	6
四 MLT 算法总览	6
五 Metropolis 采样算法	7
5.1 平稳分布	7
5.2 细致平衡	8
六 MLT 的理论基础	8
参考文献	9

一 基本介绍

Metropolis 采样方法来自于计算物理，主要解决高维空间中采样困难的问题。

Metropolis 方法中路径根据它们对理想图像的贡献进行采样，在路径空间进行随机行走。从单个种子路径开始，通过应用随机突变（例如向当前路径添加新顶点）来生成一系列光传输路径。每个突变都以专门计算得到的概率被接受或拒绝，以确保根据路径对理想图像的贡献对路径进行采样。然后，通过对许多路径进行采样并记录它们在图像平面上的位置来估计该图像。Metropolis 光传输算法是无偏的，可以处理一般的几何和散射模型，使用很少的存储空间，并且可以比以前的无偏方法效率高几个数量级。它在通常被认为困难的问题上表现特别好，例如涉及明亮的间接光、小的几何孔或光滑表面的问题。此外，即使对于具有相对简单照明的场景，它也不输于以前的无偏算法。

我们先介绍基于马尔科夫链的 Metropolis 算法，然后再引入 Metropolis 光传输的应用。

二 马尔科夫链

2.1 直接采样的接受-拒绝法

当我们想采样某种分布的样本时，如果只要概率密度分布函数，根据一些方法（比如逆函数法、接受-拒绝法）就能采样到该分布的样本。

但有些时候我们不知道分布的显式表示，因此无法使用逆函数法；或者有些时候被积函数非常复杂，维度非常高，如果用接受-拒绝法效率会非常低（每个样本在每个维度的值都有被拒绝的概率，总概率就会非常大），因为很多样本都会被拒绝。

Metropolis 方法能够从任意的概率分布中采样，获得正比于被积函数 $f(x)$ 的概率密度为 $p(x)$ 的样本分布。

先简单描述一下接受拒绝法（冯诺依曼提出）。对于有限支撑（在有限区间内函数值不为 0） $[a, b]$ 内的函数，函数值属于 $[0, f_{max}]$ ，想要找一堆符合该函数分布的样本：

- 产生 $[a, b]$ 内均匀的随机数： $X \sim U(a, b)$ 。
- 产生一个 $[0, f_{max}]$ 上独立于 X 的均匀分布的随机数： $Y \sim U(0, f_{max})$ 。
- 如果 $Y \leq f(X)$ ，就接受该样本 X ；否则就拒绝样本。
- 然后再循环上述过程，去获取下一个样本。

当然，也可以做一些优化，比如找一个近似于 $f(x)$ 的函数 $g(x)$ 满足在 $[a, b]$ 中 $g(x) \geq f(x)$ 。然后上述过程就改为：

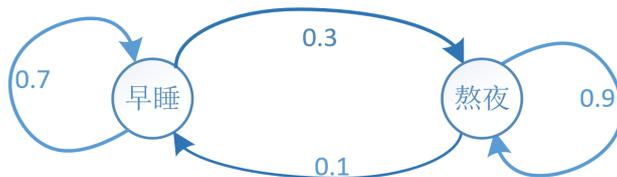
- 产生 $[a, b]$ 内符合 $g(x)$ 分布的样本： $X \sim g(x)$ 。
- 产生一个 $[0, g(X)]$ 上的独立于 X 的均匀随机数 Y ： $Y \sim U(0, g(X))$ 。
- 如果 $Y \leq f(X)$ ，就接受该样本 X ；否则就拒绝样本。
- 然后再循环上述过程，去获取下一个样本。

我们可以看到，接受拒绝法不要求出归一化概率密度函数，只需要知道能够计算目标函数 $f(x)$ 的值，就可以产生符合目标函数分布的样本。Metropolis 方法也同样可以，它也不需要求解归一化的概率密度函数。

2.2 马尔科夫链

虽然完整学习马尔科夫过程需要一些时间和精力，但所幸光传输中我们并不需要过多精力放在这上面，想要入门学习马尔科夫过程可以参考中科院孙应飞教授的随机过程课程。

假如我们有一堆随机状态，构成一个状态序列 x_0, \dots, x_n ，其中每一个状态都与上一个状态有关，即一个状态会有一些的概率转移到下一个状态。比如离散情况下，考虑一个人睡觉，有两种情况，一是早睡，二是熬夜；当一个人第一天早睡时，第二天也倾向于早睡，而当他第一天熬夜时，第二天也更容易熬夜。假如 x_0 是早睡，那么 x_1 是早睡的概率是 0.7，熬夜的概率是 0.3；假如 x_0 是熬夜，那么 x_1 是熬夜的概率是 0.9，早睡的概率是 0.1。之后我们就可以画一个概率转移图：



所以，给定 x_0 ，按照转移概率，下个状态会不断生成，也就是说马尔科夫链是一个状态链，链中系统状态较高的概率就会出现的次数越多。考虑在采样符合某分布的样本时应用马尔科夫方法，可以在目标函数值越高的地方获得更多的样本，这与其他的标准蒙特卡洛方法结果是类似的。

一般我们要保证马尔科夫链中的状态是连通的，也就是说，从当前状态转移到其他任何状态（不论经过多少步）的概率都要大于 0。连通也叫作不可约性 (irreducible)。

另外马尔科夫链还可能具有周期性，比如对于从某个状态出发，必须要经过一定周期（转移至少两次）才能回到初始状态，那么就是周期态。在我们的例子中，当前状态是早睡，下一个状态还有可能是早睡，因此转移一次就还是初始状态，所以并不具有周期性。有的时候有的马尔科夫链中，下一次状态不可能为当前状态，所以呈现一定的周期性（要保证能够返回到原状态）。

如果经历有限步能回返到原状态，就说明该马尔科夫链具有正回返性 (positive recurrent)。

如果马尔科夫链是正回返、连通，且是非周期态，那么就称它为遍历态。对于遍历态的马尔科夫链而言，不管它初始状态是什么样的，该链所有样本的分布最终都能达到平稳的分布（目标分布）。我们再解释一下平稳分布，以离散情况为例。

2.3 平稳分布

假设一共有 n 种状态，设各个状态 j 的概率为 π_j ， $p_{i \rightarrow j}$ 表示从状态 i 一步到状态 j 的概率：

$$p_{i \rightarrow j} = p(x_t = j | x_{t-1} = i) \quad (二.1)$$

构成状态转移矩阵（注意矩阵中的转移概率都是一步转移概率）：

$$P = \begin{bmatrix} p_{1 \rightarrow 1} & p_{1 \rightarrow 2} & \dots & p_{1 \rightarrow n} \\ p_{2 \rightarrow 1} & p_{2 \rightarrow 2} & \dots & p_{2 \rightarrow n} \\ \cdot & \cdot & \dots & \cdot \\ p_{n \rightarrow 1} & p_{n \rightarrow 2} & \dots & p_{n \rightarrow n} \end{bmatrix} \quad (二.2)$$

假如一开始有 m 个样本，这 m 个样本属于各个状态的概率各不相同。第 i 个状态的样本占比是 m_i ，构成一个样本数组：

$$\begin{bmatrix} m_1 & m_2 & \dots & m_n \end{bmatrix} \quad (二.3)$$

经过一次状态转移以后：

$$\begin{bmatrix} m_1 & m_2 & \dots & m_n \end{bmatrix} \begin{bmatrix} p_{1 \rightarrow 1} & p_{1 \rightarrow 2} & \dots & p_{1 \rightarrow n} \\ p_{2 \rightarrow 1} & p_{2 \rightarrow 2} & \dots & p_{2 \rightarrow n} \\ \cdot & \cdot & \dots & \cdot \\ p_{n \rightarrow 1} & p_{n \rightarrow 2} & \dots & p_{n \rightarrow n} \end{bmatrix} \Rightarrow \begin{bmatrix} m_1^{new} & m_2^{new} & \dots & m_n^{new} \end{bmatrix} \quad (二.4)$$

简写为 $mP \Rightarrow m$ 。各个状态的样本分布会被更新。当经过一定时间后，各个状态 j 的样本分布概率都会保持稳定，也就是说，当 $m = mP$ 时，表示 m 已经是稳态了。

如果条件更苛刻一点，比如：

$$m_j p_{i \rightarrow j} = m_i p_{j \rightarrow i} \quad (二.5)$$

则称为细致平衡 (detailed balance、local balance)。

三 Metropolis 算法

我们本节的内容使用的符号与 PBRT 书一致，描述的内容也是 PBRT 上的，因为 PBRT 的描述更有利于大家对 Metropolis 算法有一个初步的印象（我也懒得把符号统一了）。

Metropolis 算法用于从非负函数中产生一系列的样本，使得样本分布正比于函数值。这种方法不需要能对 f 积分，也不需要对被积函数进行归一化，每次迭代都能够产生可用样本。

根据马尔可夫链相关原理，首先把目标概率分布看做是由一个个状态构成的状态空间，然后从任意一个状态出发，通过随机游走建立马尔科夫链。随机游走中每一步都要遵循状态空间转移概率分布。

马尔可夫链解决积分问题一般有两种思路：一是已知目标概率分布 p ，然后使用蒙特卡洛方法对目标概率分布进行采样；第二种是已知转移概率分布，则给定任意初始概率分布，马尔科夫链会逐渐趋向于稳定分布。Metropolis 方法属于第一种。

Metropolis 方法会逐渐产生一系列的随机数，下一个随机数的产生仅依赖于上一个随机数，而随着随机数数量越多，随机序列的所有样本的分布会越来越逼近于目标概率分布。它也会使用接受-拒绝的方式，即下一个随机数产生时会有候选随机数，这个候选随机数有一定概率被接受或者被拒绝，如果被接受了，就作为新的样本进入下一次迭代，否则当前的随机数会继续被用在下一次迭代中（这意味着每次迭代都能够产生可用的样本，有可能是上一个样本同时在下次迭代中使用）。

Metropolis 方法有一些缺点：产生的随机序列中的连续样本在统计上是相关的，因此不可能确保 Metropolis 方法产生的一小组样本在整个域中分布良好。只有样本量足够的情况下，样本才能覆盖整个域。

本节我们先描述方法，部分证明会在后面或者参考文献给出。

3.1 Metropolis 采样方法

对于函数 $f: \mathbb{R}^n \rightarrow \mathbb{R}$ ，在选定第一个样本 X_0 以后，后续样本通过对 X_0 变异来得到，而变异的结果 X' 有一定概率被接受或者拒绝。如果接受， $X_1 = X'$ ，如果拒绝， $X_1 = X_0$ 。当从一个状态到另一个状态的转换要按照一些要求（稍后描述）时，得到的 X_i 的分布就会到达一种均衡分布（平稳分布）。在极限状态下，样本集 X_i 的分布就完全正比于 $f(x)$ 的概率密度函数 $p(x)$ ：

$$p(x) = \frac{f(x)}{\int_{\Omega} f(x) d\Omega} \quad (三.1)$$

于是可知，如何生成变异样本，以及如何根据一些限制来接受或拒绝样本都是比较重要的。假设我们有一个变异方法从 X 变到 X' ，我们需要能够计算出一个试探性的转移函数 (tentative transition function) $T(X \rightarrow X')$ 给出从 X 变异到 X' 的概率密度。同时，给定转移函数以后，还需要定义一个接受概率 $a(X \rightarrow X')$ 给定从 X 到 X' 的接受概率，这使得可以保证样本分布正比于 $f(x)$ 。如果分布已经均衡了，那么两个状态之间的转移密度应该是相同的：

$$f(X)T(X \rightarrow X')a(X \rightarrow X') = f(X')T(X' \rightarrow X)a(X' \rightarrow X) \quad (三.2)$$

这就是前面说过的细致平衡 (detailed balance)。特别地，使达到平衡的速率最大化的 a 的定义是：

$$a(X \rightarrow X') = \min \left(1, \frac{f(X')T(X' \rightarrow X)}{f(X)T(X \rightarrow X')} \right) \quad (三.3)$$

如果转移概率在两个方向相同，那么：

$$a(X \rightarrow X') = \min \left(1, \frac{f(X')}{f(X)} \right) \quad (三.4)$$

我们得到基本的 Metropolis 采样算法：

```
1 X=X0
2 for (i = 1; i <= n; i++){
3     X'=mutate(X)
4     a=accept(X,X')
```

```

5     if (random() < a)
6         X=X'
7     record(X)
8 }

```

由于 Metropolis 算法避免在 $f(x)$ 值较小的区域采样，这些区域的样本会非常少。为了得到 $f(x)$ 在这些区域的一些信息，通常会使用期望的值 (expected values) 技术来提高基本的 Metropolis 算法。在这种情况下，我们会记录每个 X 和 X' (不管哪个被接受)，每个这些记录的样本都有一个关联权重，对于 X 是 $(1-a)$ ；对于 X' 是 a 。期望的值并不影响我们决定的当前状态：

```

1 X=X0
2 for (i = 1; i <= n; i++){
3     X'=mutate(X)
4     a=accept(X,X')
5     record(X,1-a)
6     record(X',a)
7     if (random() < a)
8         X=X'
9 }

```

我们可以看到，在极限中，两个伪代码中的 X 和 X' 的权重分布相同。与基本算法相比，期望的值技术能更快地给出更平滑的结果和更多关于 $f(x)$ 较低区域的信息。

3.2 变异策略

选择变异策略要求能够计算试探性的转移函数 (tentative transition function)。前面说过，如果转移函数是对称的，那么甚至不需要计算它们。同时使用多种变异策略也是一种有效手段，比如有的变异策略在某些条件下更有效，而在另外的条件下不够有效，那么同时使用多种策略是很好的选择。

在变异中，一次变异幅度很大可以使样本更快地遍历状态空间，而不是让样本被挤在更小的区域中。但当函数值在某个样本 X 处较大时，很容易造成很多次变异都被拒绝 ($f(X) \gg f(X')$ ，导致 $a(X \rightarrow X')$ 值非常小)。我们希望避免一次性出现一大堆相同的样本，以及更好地去让样本分布在整个状态域上——许多样本都在同一个值上会增加方差，直觉上我们移动的越多，结果就会越好。在这种情况下，提出样本 X' 尽量是小突变，其中 f 仍然相对较大，导致更高的接受率，避免了同样的样本值出现过多次。

一个有用的方式是对样本 X 应用随机扰动，每个样本可能是多维向量 (x_0, x_1, \dots) ，那么有些或者所有维度都可以被扰动。通过加上或减去一个被 s 放缩的随机数，但要注意这里假设每个维度的数都不能超过 1：

$$x'_i = (x_i \pm s\xi) \bmod 1 \quad (三.5)$$

这种随机扰动的方式是对称的，所以不必计算 $T(X \rightarrow X')$ 。或者，直接对 x_i 赋值一个新的随机数： $x_i = \xi$ ，这也是对称的方式。偶尔以这种方式生成一个全新的样本是很重要的，因为它可以确保我们不会陷入状态空间的一部分而不会对其余部分进行采样。能够达到所有状态 $X \in \Omega$ (其中 $f(X) > 0$) 是必要的，为了保证遍历性，需要使得对于全部 X 和 X' 而言 $T(X \rightarrow X') > 0$ 。

另一种方法是使用匹配被采样函数某部分的概率密度来采样：如果我们有一个概率密度函数 $p(x)$ 与 f 的某些部分很相似，那么就可以使用 $p(x)$ 来导出一个变异策略，使得新样本 $X \sim p$ ，此时转移函数就直接是：

$$T(X \rightarrow X') = p(X') \quad (三.6)$$

换句话说，当前状态 X 对于计算转移密度并不重要，只跟要转移到的状态有关。

3.3 启动偏差

我们至今有一个问题没有讨论过，就是如何选择初始样本 X_0 。我们已经假设了所有样本都是基于 f 采样到的，但如果一个样本不是从 f 采样到的，那么就会引起一个叫做启动偏差 (start-up bias) 的问题。

一个一般的解决方案是运行 Metropolis 算法迭代很多轮，然后丢弃已经产生的样本，用最后一个样本重新开始执行 Metropolis 过程。但是这样在一开始花费的时间太长，另一方面，我们只能猜测需要丢弃多少初始样本才能移除启动偏差。

如果另一种采样技术可以被使用，那么就可以用一个替代的方案：首先使用任意概率分布 $X_0 \sim p(x)$ 来采样到第一个样本，然后从状态 X_0 来开始马尔可夫链，但是所有样本的贡献都被下面的权重加权：

$$w = \frac{f(X_0)}{p(X_0)} \quad (三.7)$$

该方法完全消除了启动偏差，并以可预测的方式实现。

但是，如果 $f(X_0) = 0$ ，所有的样本的权重都将是 0，然而这并不意味着算法是有偏的，结果的期望值仍然会收敛到正确分布 (Veitch 论文里有证明)。为了降低方差以及避免这种问题，可以采样 N 个候选样本值 Y_1, \dots, Y_N ，并定义权重 $w_i = \frac{f(Y_i)}{p(Y_i)}$ ，之后从 Y_i 中为 Metropolis 算法抽取初始 X_0 ，抽取概率正比于他们的相关权重。并把所有的权重 w_i 取平均得到 w 。之后所有的 Metropolis 样本的权重都用 w 来加权。

3.4 一个简单的实例

我们来展示 Metropolis 如何采样一个一维函数。设某个函数 $f(x)$ ：

$$f(x) = \begin{cases} (x - 0.5)^2 & 0 \leq x \leq 1 \\ 0 & otherwise \end{cases} \quad (三.8)$$

但此时假设我们并不知道实际 $f(x)$ 的形状，我们企图用 Metropolis 方法采样。

定义两个变异策略。策略一是直接在 $[0, 1]$ 区间均匀采样（由于区间范围恰好是 1，所以此时转移概率密度就是 1）：

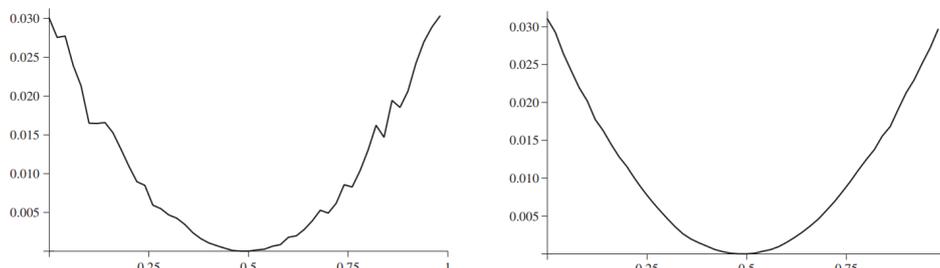
$$\begin{aligned} mutate_1(X) &\rightarrow \xi \\ T_1(X \rightarrow X') &= 1 \end{aligned}$$

策略二是在原来的值 X 上加 ± 0.05 范围内的随机数。只要偏移后的 X' 与 X 的距离小于 0.05，它的转移概率就是一个常量——因为偏移后的值的范围是 0.1 ($X - 0.05$ 到 $X + 0.05$)，要保证在这个区间的概率密度积分为 1，所以概率密度就是 $1/0.1$ 。

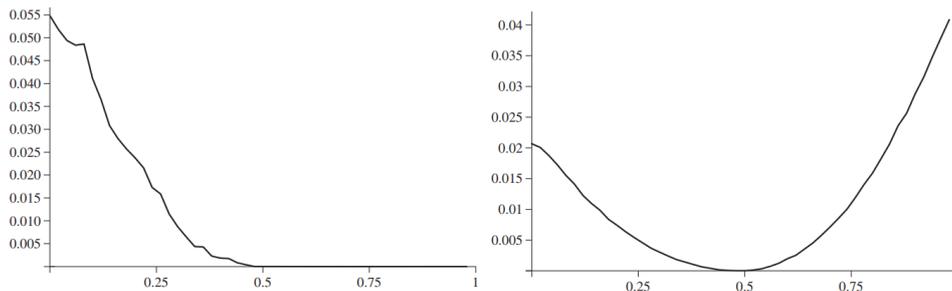
$$\begin{aligned} mutate_2(X) &\rightarrow X + 0.1(\xi - 0.5) \\ T_2(X \rightarrow X') &= \begin{cases} \frac{1}{0.1} & |X - X'| \leq 0.05 \\ 0 & otherwise \end{cases} \end{aligned}$$

为了得到初始样本，从 $[0, 1]$ 中以均匀概率密度采样一个样本， $X_0 = \xi$ 。除了 Ω 上的单个点 ($X = 0.5$)，都能保证 $f(X) > 0$ 。

执行 Metropolis 采样以后，下图左边是只用变异策略一得到的结果，右图是 10% 的概率用策略一，90% 的概率用策略二得到的结果（由于策略一接受率不高，导致方差较大）：



但是，当只用策略二时也有问题，比如下图：左边只采样了 10000 个样本（没有样本能够跨越到 $X_i > 0.5$ ），右边采样了 30000 个样本，确实能跨越到 $X_i > 0.5$ ，但是估计的分布仍然不是很对，我们知道该方法是正确的，但是它的效率非常低。因此，结合两种策略是很重要的。



3.5 使用 Metropolis 方法估计积分

考虑如何用 Metropolis 方法来估计例如 $\int f(x)g(x)d\Omega$ 的积分。考虑蒙特卡洛估计器（样本 X_i 是以概率密度 $p(x)$ 抽取的）：

$$\int_{\Omega} f(x)g(x)d\Omega \approx \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)g(X_i)}{p(X_i)} \quad (三.9)$$

如果我们产生了一系列的样本 X_1, \dots, X_N 其概率密度正比于 $f(x)$ ，那么积分就可以计算为：

$$\int_{\Omega} f(x)g(x)d\Omega \approx \frac{1}{N} \left[\sum_{i=1}^N \int_{\Omega} g(X_i)f(x)d\Omega \right] = \left[\frac{1}{N} \sum_{i=1}^N g(X_i) \right] \int_{\Omega} f(x)d\Omega \quad (三.10)$$

考虑最简单的情况，就是 $f(x) = 1$ ， $\Omega = [0, 1]$ ，那么：

$$\int_{\Omega} f(x)g(x)d\Omega = \int_{\Omega} g(x)d\Omega \approx \frac{1}{N} \left[\sum_{i=1}^N g(X_i) \right] \quad (三.11)$$

这样看起来可能会觉得 Metropolis 方法有些啰嗦，或者不明所以。我们后面会在光传输中描述如何应用 Metropolis 方法。但有一点大家应该知道，即生成正比于目标函数值的分布的样本集是很重要的。

四 MLT 算法总览

我们本节到后面的内容的符号描述都是使用 [1] 上

对于每条路径 $\bar{x} = \mathbf{x}_0 \mathbf{x}_1 \dots \mathbf{x}_k$ ，设对图像贡献函数 f 的积分式为：

$$\int_D f(\bar{x})d\mu(\bar{x}) \quad (四.1)$$

Metropolis 光传输 (MLT) 方法可以产生一系列正比于 f 的概率密度的样本 $\bar{X}_0, \bar{X}_1, \dots, \bar{X}_N$ ，其中 \bar{X}_i 是 \bar{X}_{i-1} 变异得到的。变异方法包括增加、删除或者替换一些当前路径的顶点。

然而，每次变异都有被拒绝的可能，取决于当前路径和新路径的贡献。比如如果一个新的路径穿墙了（两个顶点被阻挡）那么就会被拒绝。Metropolis 方法给出了每次变异被接受的概率，使得样本分布正比于函数 f 的值。在路径变异中，把新路径的贡献添加到像素值中（如果在滤波器范围有多个像素值，就贡献到这些像素中），除了滤波器的权重会根据距离像素中心不同而不同以外，样本值的权重都是一样的，因此获得样本多的像素可能会更亮一些，但这确实是属于正确的情况（这是基本方案的效果，以后还会介绍允许不同权重的改进方案）。

```

1 MLT算法实现：
2 path = intial_Path()
3 image = {arrays of zeros}
4 for (i=1; i<=N; i++){

```

```

5   path' = Mutate(path)
6   a = acceptProbability(path->path')
7   if(random() < a)
8       path = path'
9   recordSample(image, path)
10 }

```

MLT 算法的关键好处在于路径可以局部改变，变异策略可以只对当前顶点做少量改变，使得仅仅多发出一到两条光线就可以构建一个新的路径。且当一条重要路径找到以后，它周边的重要路径也都可以被探索到，因此相当于分摊了找到重要路径的成本。第三，变异集合很容易扩展，构建保留一些路径特性的变异（比如保留光源位置，变异其他顶点），可以探索场景中的各种相关性，对于复杂的光照问题设计专门的变异策略也是可能的。

五 Metropolis 采样算法

给定一个状态空间 Ω 和一个非负函数 $f: \Omega \rightarrow \mathbb{R}^+$ ，给定一个初始状态 $\bar{X}_0 \in \Omega$ ，目标是生成一系列样本 $\bar{X}_0, \bar{X}_1, \dots$ 使得 \bar{X}_i 分布符合 f （无论我们是从什么状态 \bar{X}_0 开始）。这种随机游走（且下一个样本只会决定于当前样本）就是马尔可夫链。

转移函数 (transition function) 设为符号 K (PBRT 书中，试探性的转移函数 (tentative transition function) 用的符号是 T ，所以不要弄混了，我们目前定义的是转移函数，后面会介绍试探性的转移函数)，要保证对于全部的 $\bar{X} \in \Omega$ ：

$$\int_{\Omega} K(\bar{X} \rightarrow \bar{X}') d\mu(\bar{X}') = 1 \quad (五.1)$$

5.1 平稳分布

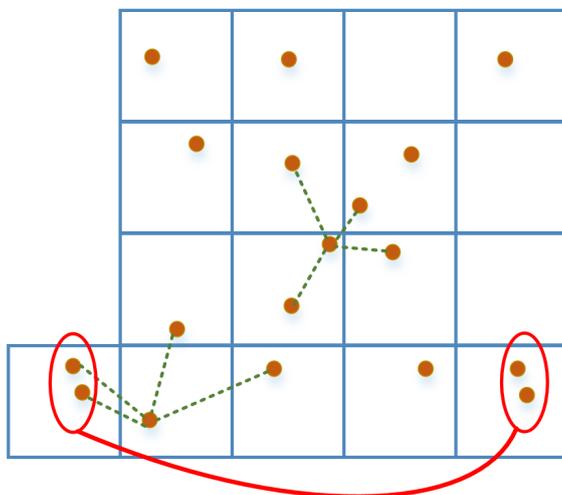
前面马尔可夫链中有平稳分布的描述，和下面其实是一样的，只是我们用 [1] 的语言再描述一遍。

考虑 \bar{X}_i 是以 p_i 概率密度分布的样本， p_i 又由 p_{i-1} 决定：

$$p_i(\bar{X}') = \int_{\Omega} K(\bar{X} \rightarrow \bar{X}') p_{i-1}(\bar{X}) d\mu(\bar{X}) \quad (五.2)$$

在适当条件下（后面会解释）， p_i 会收敛到一个独一无二的密度函数 p^* 。注意 p^* 并不取决于初始状态 \bar{X}_0 。

对于这种方式给一个简单的例子，考虑一个状态空间有 n^2 个顶点分布在 $n \times n$ 的网格中，每个顶点都与它的四邻域用边来连接，这些连接边可能会 warp 到另一边：



每个顶点在变异中要么转移到某个邻域点，要么保留在当前点，转移到每个点的概率相同，都是 $1/5$ 。以某个点 \bar{X}_0 为例，它转移一次以后， \bar{X}_1 在邻域四个点的概率相同，再转移一次以后，它会随机分布在

周边 13 个点（概率不一定相同， \bar{X}_0 邻域那四个点的概率密度肯定比其他点大）当这个过程持续一段时间以后，最后 p_i 就能收敛到一个固定的概率密度 p^* 满足：

$$p^*(\bar{x}') = \sum_{\bar{x}} K(\bar{x} \rightarrow \bar{x}') p^*(\bar{x}) \quad (五.3)$$

前面我们在马尔可夫链中关于平稳分布描述的与这里是一样的，只是前面用了一个矩阵来表示，这里把矩阵的一行给单独拿了出来。

前面早睡（标记为 e ）和晚睡（标记为 l ）的例子也是这样。比如状态 \bar{X}_0 是早睡，那么第二天早睡和晚睡的概率分别是 $[0.7, 0.3]$ 。第三天时，早睡和晚睡的概率又变为了：

$$\begin{bmatrix} 0.7 & 0.3 \end{bmatrix} \begin{bmatrix} 0.7 & 0.3 \\ 0.1 & 0.9 \end{bmatrix} = \begin{bmatrix} 0.52 & 0.48 \end{bmatrix} \quad (五.4)$$

到第四天开始，晚睡的概率超过了早睡，但是概率变化会逐渐变小，直到趋于平稳。

5.2 细致平衡

在一个典型的物理系统中，转移函数 K 由控制系统的物理定律决定。给定一些任意的初始状态，系统通过 K 控制的跃迁向平衡状态转移。

Metropolis 方法与之不同，其想法是构造一个转移函数 K ，其产生的平稳分布将与给定的 f 成比例，并且将尽可能快地收敛到 f 。这项技术很简单，并且有一种称为细致平衡的直观物理解释。此时，转移函数叫做试探性的转移函数 (tentative transition function) T ，因为转移不一定就能成功。

假设已经达到了平稳状态，即 p_{i-1} 正比于 f ，考虑我们有任意两个状态 \bar{X} 和 \bar{Y} ，在均衡状态下而言必须需要定义相应的转移密度 $K(\bar{X} \rightarrow \bar{Y})$ ，该转移密度正比于 $f(\bar{X})T(\bar{X} \rightarrow \bar{Y})a(\bar{X} \rightarrow \bar{Y})$ ；同理， $K(\bar{Y} \rightarrow \bar{X})$ 也要正比于 $f(\bar{Y})T(\bar{Y} \rightarrow \bar{X})a(\bar{Y} \rightarrow \bar{X})$ ，为了保持平稳，只需要保证：

$$f(\bar{X})T(\bar{X} \rightarrow \bar{Y})a(\bar{X} \rightarrow \bar{Y}) = f(\bar{Y})T(\bar{Y} \rightarrow \bar{X})a(\bar{Y} \rightarrow \bar{X}) \quad (五.5)$$

该条件即细致平衡。此时可以证明得到 $p_{i-1}(\bar{X}) = p_i(\bar{X})$ （见 [1]），即均衡分布一定是正比于 f 的。

a 作为接受概率，我们自然希望它越大越好（被拒绝的少，防止样本聚集增大方差），前面介绍过的 a 的方法：

$$a(\bar{X} \rightarrow \bar{Y}) = \min \left\{ 1, \frac{f(\bar{Y})T(\bar{Y} \rightarrow \bar{X})}{f(\bar{X})T(\bar{X} \rightarrow \bar{Y})} \right\} \quad (五.6)$$

这样设定以后，当 $f(\bar{Y})T(\bar{Y} \rightarrow \bar{X}) > f(\bar{X})T(\bar{X} \rightarrow \bar{Y})$ 时，则变异就一定会被接受。否则就有一定的概率被接受，一定的概率被拒绝。总的来看，每个方向的预期移动次数是相同的。

Metropolis 方法在几个方面不同于遗传算法。首先，它们有不同的目的：遗传算法用于优化问题，而 Metropolis 方法用于采样问题（没有搜索最佳值）。遗传算法适用于个体群体，而 Metropolis 只存储一个当前状态。最后，遗传算法在选择允许的突变时有更大的自由度，因为它们不需要计算其行为的条件概率。

参考文献

- [1] Veach E . Robust Monte Carlo Methods for Light Transport Simulation[J]. Ph.d.thesis Stanford University Department of Computer Science, 1998.
- [2] Arvo, J. [1995]. Analytic Methods for Simulated Light Transport, PhD thesis, Yale University.
- [3] Pharr M, Jakob W, Humphreys G. Physically based rendering: From theory to implementation[M]. Morgan Kaufmann, 2016.