重要性重采样技术

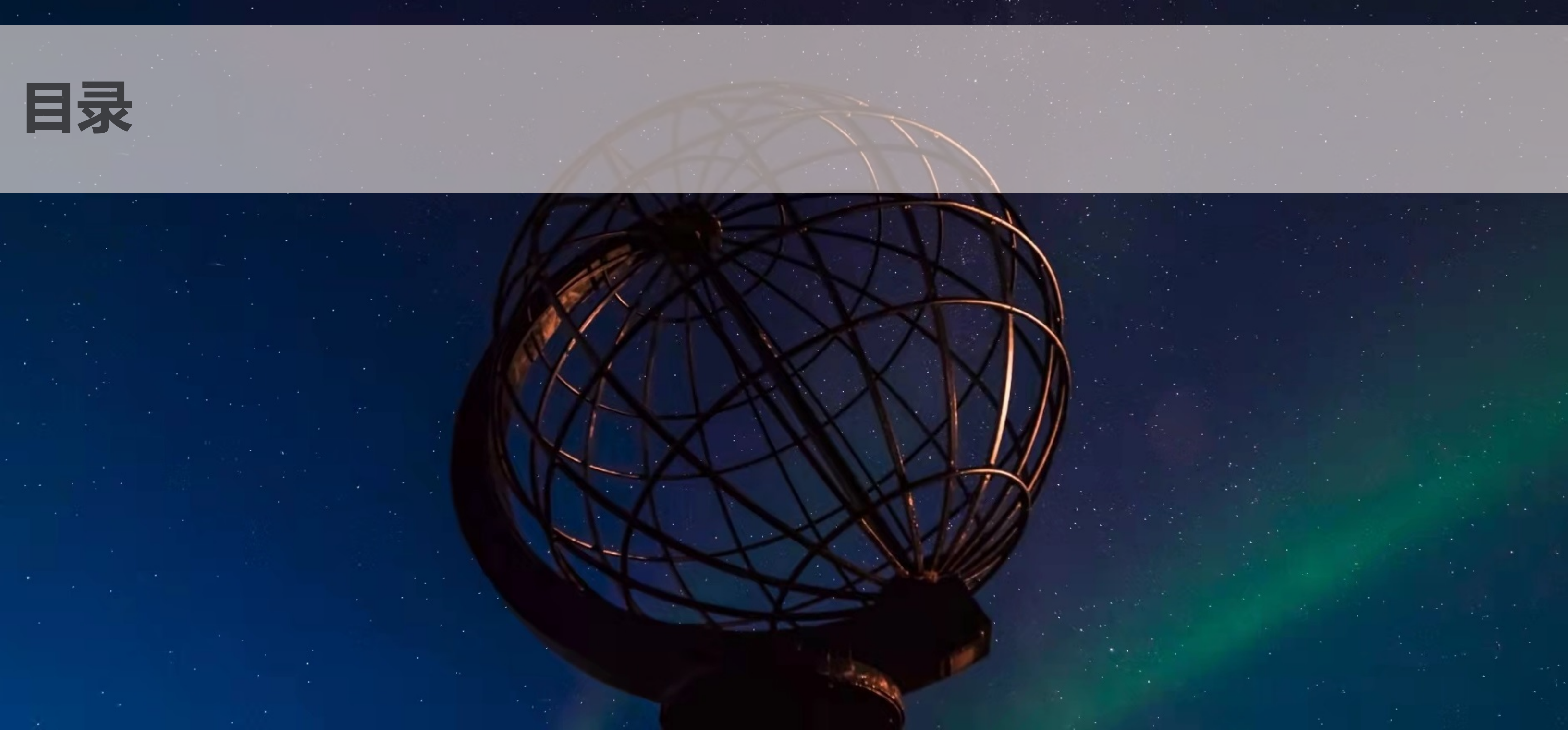
Dezeming Family

Dezeming

Art. No 0
ISBN 000–00–0000–00–0
Edition 0.0

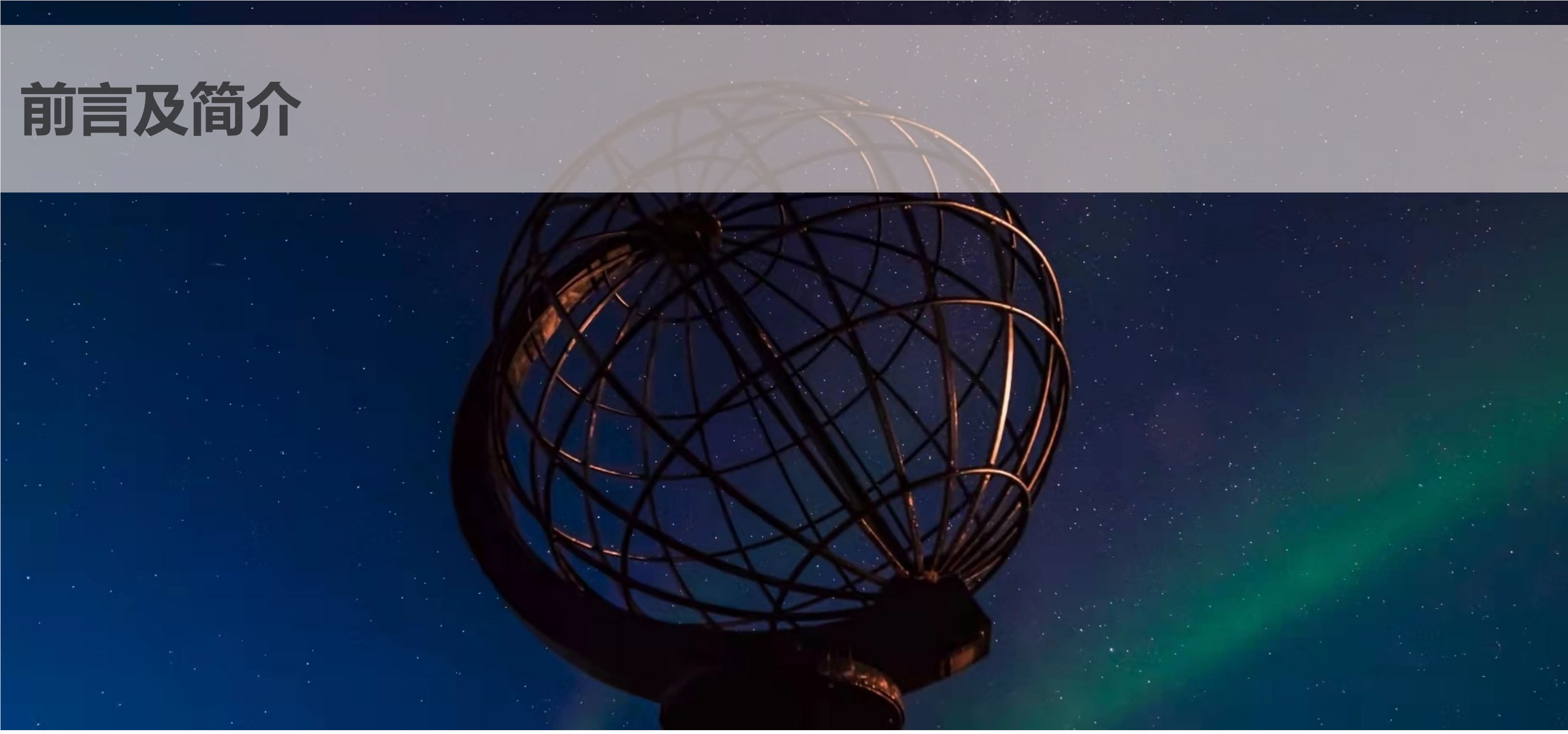
Cover design by Dezeming Family

Published by Dezeming
Printed in China



目录

前言及简介



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。

0.1 本书前言

蒙特卡洛光线追踪基于重要性采样，可以更快地收敛到方差较小的范围。

而基于此，重要性重采样技术（RIS）可以比一般的蒙特卡洛光线追踪方法有更低的方差。同时，重要性重采样技术也可以与分层抽样、多重重要性采样（MIS）等方法相结合。使用 RIS 可以在直接采样光中降低 33% 的方差，使用 RIS 来采样 BRDF 也同样会带来更好的结果。

我们将在本书中介绍和分析 RIS 原理，给出更通俗易懂的描述。

1. 重采样重要性采样基本原理

本章介绍重采样重要性采样的基本原理和简单的证明过程。

1.1 蒙特卡洛与重要性重采样

我们先使用蒙特卡洛方法进行估计：

$$I = \int_{\Omega} f(x) d\mu(x) \approx \frac{1}{N} \sum_{i=1}^{N} \frac{f(t_i)}{\hat{q}(t_i)} \tag{1.1.1}$$

这里的 $\hat{q}$ 是产生的样本 t_i 的概率密度函数。

首先我们介绍采样重要性重采样技术（Sampling Importance Resampling (SIR)），该技术由 [1] 提出。

假如我们想从概率密度函数为 g 的分布中生成样本 [2]，但是由于 g 没有一个解析解的形式或者太复杂，我们不能直接生成（无法求反函数），所以我们就先从一个分布 p 中产生样本分布，并把这些样本合理地加权。然后，通过从这些样本中抽取一个与权重成比例的样本，对这些样本进行重新采样。

[1]. 从 p 产生 M 个样本，$\boldsymbol{X} = \left\langle X_1, X_2, ..., X_M \right\rangle$。

[2]. 从每个样本计算权重 w_j。

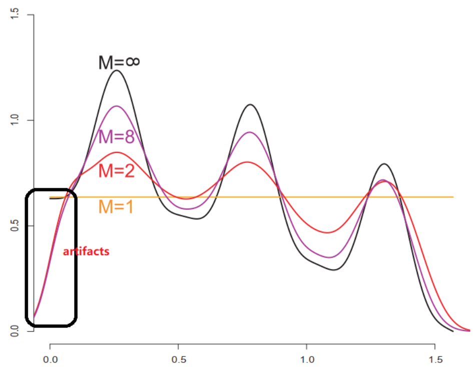
[3]. 从 $\boldsymbol{X}$ 中以正比于 $\left\langle w_1, w_2, ..., w_M \right\rangle$ 的概率密度抽取一个样本 Y。

如果我们权重计算 $w_j = \frac{g(X_j)}{p(X_j)}$，则产生的样本 Y 将近似于 g 分布。重采样的效果其实等同于从 p 中采样然后“过滤”它们，因此有一个近似于 g 的分布。

当样本数 M=1 时，Y 的分布就是 p，但是当 M 趋近于无穷大的时候，Y 的分布就会接近于 g。我们可以这么思考，比如 p 是 $[0,1]$ 之间的均匀采样，那么 pdf=1 是个常数，因此我们可以发现 w_j 权重其实就是 $g(X_j)$，因此相当于我们采样到的这堆样本中，样本 X_j 对应于权重 $g(X_j)$。

当样本无限多的时候，其实样本 X_j 就恰好具有 g 分布。因此，M 必须要非常大，才能让偏差小到可以被忽略。

我们引用 [2] 中的示例，假设 g 正比于 $\cos(\theta)+\sin^4(\theta)$，且 p 是均匀的，则 Y 的分布随着 M 的不同如下：



对 M 的不同值进行重要性重采样后得到的分布。在 M=1 时，分布为 p。在 M 为无穷大时，分布为 g。对于 M 的其他值，分布在 p 和 g 之间插值，尽管插值的确切方式未知。左侧 M=2 和 M=8 的低值是密度估计方法的 artifacts。

1.2 重采样的重要性采样

把重要性重采样技术与蒙特卡洛方法相结合就叫做“重采样的重要性采样”Resampled Importance Sampling（RIS）。

现在我们想估计函数 f 的积分值。假设我们有两个概率密度函数，一个为 p，另一个为 g。p 可以被直接采样，但是并不是 f 的很好的近似；g 是 f 很好的近似，但是可能无法被归一化或者很难来进行采样。

RIS 方法可以让我们用无偏的方式来使用 g，可以用 SIR 技术从样本中获得近似 g 的分布。这里需要注意的是，虽然我们得到的分布并不是完美和 g 分布一样的，但是对于使用这个样本去估计函数值来说，方法可以是无偏的。

加权重要性采样：

$$\hat{I}_{RIS} = \frac{1}{N}\sum_{i=1}^{N} w(\boldsymbol{X}_i, Y_i)\frac{f(Y_i)}{g(Y_i)} \tag{1.2.1}$$

注意权重函数 w 必须要正确的选择，我们注意到 g 并不是归一化的，且 Y 的密度只是 g 的近似。其实计算也很简单，在重采样阶段的权重的平均即可：

$$w(\boldsymbol{X}_i, Y_i) = \frac{1}{M}\sum_{j=1}^{M} w_{i,j} \tag{1.2.2}$$

$$\Longrightarrow \hat{I}_{RIS} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{f(Y_i)}{g(Y_i)}\cdot\frac{1}{M}\sum_{j=1}^{M}\frac{g(X_{i,j})}{p(X_{i,j})}\right) \tag{1.2.3}$$

当 $M=1$ 时，RIS 其实就是标准的重要性采样了，概率密度为 p。

为了保证 RIS 的估计是无偏的，需要保证两个条件：（1）g 和 p 必须要处处大于等于 0。（2）M 和 N 必须大于 0。

无偏性以及方差的证明可以在 Talbot 的硕士论文 [3] 的附录中看到。

1.3 全局光照中的应用

在 RIS 中，我们可以看到权重的计算 $w(\boldsymbol{X}_i, Y_i)$ 与 $\boldsymbol{X}_i$ 和 Y_i 都有关。我们需要首先产生一组样本 $\boldsymbol{X}_i$，然后从中抽取一个样本 Y_i。

在光线追踪的直接光照中，我们可以如此应用 RIS 方法。首先我们给出光传输方程：

$$L(p' \to p) = L_e(p' \to p) + \int_A f(p'' \to p' \to p) L(p'' \to p') V(p'' \leftrightarrow p') G(p'' \leftrightarrow p') dA(p'') \tag{1.3.1}$$

V 代表表面可见性。由于计算可见性分布很难，所以我们令 $g = f \cdot G \cdot L_e$。我们可以用其他概率密度来计算光照，然后生成符合 g 分布的样本。这样的估计就会获得更理想的结果。

尽管这种方式是无偏的，但是估计方差还是会与 M 和 N 的选取有关。如何适当选择理想的 M 和 N 是一个很重要的挑战。

1.4 期望分析

首先来分析一下期望：

$$E\left(\hat{I}_{RIS}\right) = E\left[\frac{1}{N}\sum_{i=1}^{N}\left(\frac{f(Y_i)}{g(Y_i)} \cdot \frac{1}{M}\sum_{j=1}^{M}\frac{g(X_{i,j})}{p(X_{i,j})}\right)\right] \tag{1.4.1}$$

由于：

$$E(X \cdot Y) = E\Big(E(Y|X) \cdot X\Big) \tag{1.4.2}$$

所以可以转化为：

$$E\left(\hat{I}_{RIS}\right) = E\left[E\left(\frac{1}{N}\sum_{i=1}^{N}\frac{f(Y_i)}{g(Y_i)}\middle| X_1, \dots, X_M\right) \cdot \frac{1}{M}\sum_{j=1}^{M}\frac{g(X_{i,j})}{p(X_{i,j})}\right] \tag{1.4.3}$$

注意上式中，内部的期望表示在给定使用 p 分布抽样好的 M 个数据前提下，$\frac{f(Y)}{g(Y)}$ 的期望。Y 是从这 M 个数据中抽取的。

因为所有的 Y 样本都是独立同分布的，所以求和的期望其实就是单个的期望：

$$E\left(\hat{I}_{RIS}\right) = E\left[E\left(\frac{f(Y)}{g(Y)}\middle| X_1, \dots, X_M\right) \cdot \frac{1}{M}\sum_{j=1}^{M} w(X_j)\right] \tag{1.4.4}$$

由于 Y 是从 X 中抽取的，所以我们可以这么来表示里面的期望：

$$E\left(\frac{f(Y)}{g(Y)}\middle| X_1, \dots, X_M\right) = \sum_{k=1}^{M}\left(\frac{f(X_k)}{g(X_k)} \cdot \frac{w(X_k)}{\sum_{j=1}^{M} w(X_j)}\right) \tag{1.4.5}$$

$$E\left(\hat{I}_{RIS}\right) = E\left[\sum_{k=1}^{M}\left(\frac{f(X_k)}{g(X_k)} \cdot \frac{w(X_k)}{\sum_{j=1}^{M} w(X_j)}\right) \cdot \frac{1}{M}\sum_{j=1}^{M} w(X_j)\right] \tag{1.4.6}$$

化简一下就可以得到：

$$E\left(\hat{I}_{RIS}\right) = E\left[\frac{1}{M}\sum_{k=1}^{M}\left(\frac{f(X_k)}{g(X_k)}\cdot w(X_k)\right)\right] = E\left(\frac{f(X_k)}{g(X_k)}\cdot w(X_k)\right) \tag{1.4.7}$$

$$= E\left(\frac{f(X_k)}{g(X_k)}\cdot\frac{g(X_k)}{p(X_k)}\right) = E\left(\frac{f(X_k)}{p(X_k)}\right) \tag{1.4.8}$$

注意最终得到的期望就是重要性采样的期望，因此它是无偏的。

1.5 参数的选择简介

我们先来讨论参数 g 和 p 的选择。其实也很简单，越与被积函数 f 相似，肯定方差就越小；同时我们还需要尽量保证 p 分布的样本比较容易获取。

在考虑计算函数时，应该从多个方面同时考虑来估计 M 和 N 的取值关系：

[1]. 从 p 分布中获取一个样本并估计其权重的时间 T_X

[2]. 进行重采样的时间 $r(M)$（很小，可以被忽略）

[3]. 计算 RIS 估计的时间（根据抽取的样本计算估计）T_Y

因此，总时间消耗就是：

$$T = MNT_X + N\left(r(M) + T_Y\right) \approx MNT_X + NT_Y \tag{1.5.1}$$

我们的目标其实就是尽可能在减少方差的同时降低运行时间，因此，方差和 T_X 和 T_Y 共同决定了 M 和 N 的选取。当我们不能去估计 T_X 和 T_Y 时，一个合理的方案是令 M=1，即直接使用重要性采样。

2. 编程测试

本章通过程序来测试无偏性。

2.1 准备工作

我们假设要求函数积分：

$$f(x) = \int_0^2 x^2 \sin x dx \tag{2.1.1}$$

程序如下：

```
float function1(float x) {
  return x * x * sin(x);
}
```

我们希望用 $g(x) = x \sin x$ 的概率密度来求，我们目前只能比较容易地得到 $p(x) = \frac{x}{2}$ 的概率密度。

$$P(x) = \frac{x^2}{4} \tag{2.1.2}$$

$$P(x)^{-1} = 2\sqrt{x} \tag{2.1.3}$$

下面的代码产生的 $[0-2]$ 之间的数字符合概率密度函数 $p(x)$：

```
#include <math.h>
#include <stdio.h>
#include "time.h"
float myRandom() {
  return rand() / (RAND_MAX + 1.0);
}
```

```
float generate_p_data(float* pdf) {
        //输出pdf为p的data
        float x = 2 * sqrt(myRandom());
        *pdf = 0.5 * x;
        return x;
}
```

产生符合 g 分布的程序如下：

```
float g_distribution(float x) {
  //输出g分布权重
  return x * sin(x);;
}
```

2.2 均匀采样

均匀采样来估计函数积分的表示如下：

```
//均匀采样
int count = 1000000;
float sum1 = 0.0f;
for (int i = 0; i < count; i++) {
  sum1 += function1(2 * myRandom()) / 0.5f;
}
sum1 /= (float)count;
printf("result1 = %f\n", sum1);
```

采样 1000000 次，然后取平均。注意，从概率估计的角度，除以 0.5 表示 [0-2] 之间的均匀样本 pdf 为 0.5；从函数均值的角度来说就是函数均值乘以函数区间 2。

2.3 用 p 重要性采样

重要性采样的程序稍微复杂一些：

```
//使用p进行重要性采样
float sum2 = 0.0f;
for (int i = 0; i < count; i++) {
  float pdf;
  float x = generate_p_data(&pdf);
  if (pdf > 0) {
    sum2 += function1(x) / pdf;
  }
```

```
}
sum2 /= (float)count;
printf("result2 = %f\n", sum2);
```

为了防止出现除以 0 的情况导致结果为 nan，我们判断 pdf 是否大于 0，并不把 pdf=0 的情况去除掉（因为这里的情况下 pdf 为 0 说明 x=0，计算出的函数值也是 0，因此直接舍弃即可）。

2.4 最基本的 RIS 算法

RIS 的过程稍微比较长：

```
//使用重采样重要性采样技术
float sum3 = 0.0f;
count = 100000;
for (int i = 0; i < count; i++) {
  //生成100个样本
  float sum_distribution = 0.0f;
  float X[100], w[100];
  for (int j = 0; j < 100; j++) {
    float pdf;
    X[j] = generate_p_data(&pdf);
    float g_dis = g_distribution(X[j]);
    w[j] = g_dis / pdf;
    sum_distribution += w[j];
  }
  //根据权重采样出其中一个样本
  float sample_w = sum_distribution * myRandom();
  int num = 0;
  float accum_distribution = 0.0f;
  for (int j = 0; j < 100; j++) {
    accum_distribution += w[j];
    if (accum_distribution > sample_w) {
      num = j;
      break;
    }
  }
  //使用该样本去估计
  float oneEstimate = function1(X[num]) / g_distribution(X[num]) *
      0.01 * sum_distribution;
  if (!isnan(oneEstimate)) {
    sum3 += oneEstimate;
```

```
    }
}
sum3 /= (float)count;
printf("result3 = %f\n", sum3);
```

在程序中，设 M=100，N=100000。同理，在对分布为 0 的情况中，为了程序更简单，我们对每一次的估计做判断，如果产生了 nan 数据，就把此次估计去除。

运行程序可得：

```
result1 = 2.467711
result2 = 2.467355
result3 = 2.468436
```

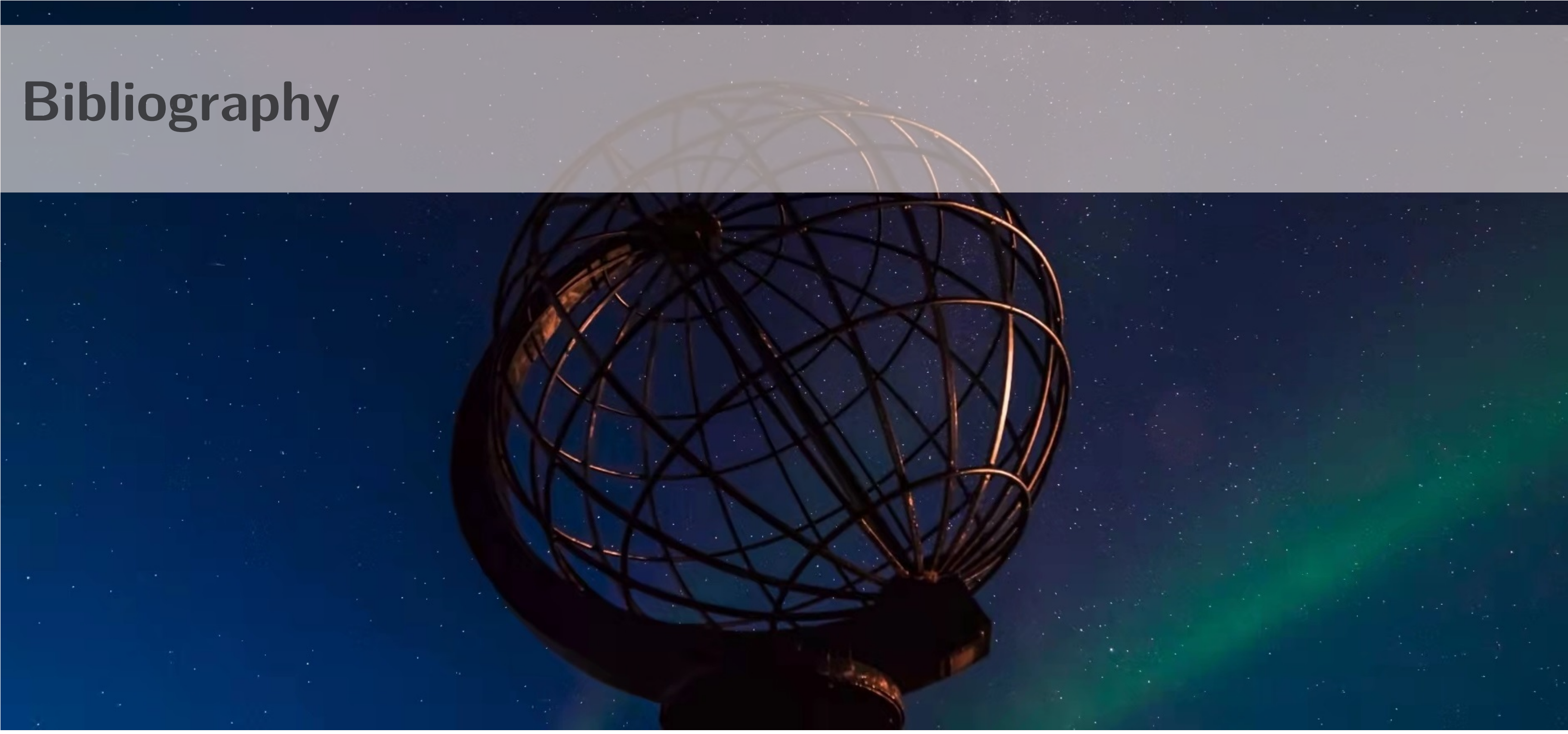
由此可知我们的估计方法得到的值和均匀采样以及重要性采样基本一致。

2.5 改进 RIS

上面的代码中我们只每次在产生的 M 个样本中抽取一个，这样似乎效率并不高。我们试试抽取十个取平均作为一次估计：

```
//根据权重采样出其中一个样本
float sum_1M = 0.0f;
for (int k = 0; k < 10; k++) {
    float sample_w = sum_distribution * myRandom();
    int num = 0;
    float accum_distribution = 0.0f;
    for (int j = 0; j < 100; j++) {
        accum_distribution += w[j];
        if (accum_distribution > sample_w) {
            num = j;
            break;
        }
    }
    //使用该样本去估计
    float oneEstimate = function1(X[num]) / g_distribution(X[num]) *
        0.01 * sum_distribution;
    if (!isnan(oneEstimate)) {
        sum_1M += oneEstimate;
    }
}
sum_1M /= 10.0f;
sum3 += sum_1M;
```

因为对于每次抽样而言，无论抽一个还是抽多个，期望都是一样的，所以也是无偏的。当我们选择的样本量 M 比较大时，这可以帮我们有效的节省时间。



Bibliography

[1] RUBIN D. B.: A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: the SIR algorithm. Discussion of Tanner and Wong. Journal of the American Statistical Association 82 (1987), 543–546.

[2] Bala, K., & Dutré, P. Importance Resampling for Global Illumination.

[3] Talbot, J. F. (2005). Importance resampling for global illumination. Brigham Young University.