

# 在参数空间使用ReSTIR加速可微渲染

## 0. 摘要

基于梯度下降的逆向渲染方法中经常使用可微渲染来求解场景参数（例如反射率、光照属性等）。计算出准确的、低方差的梯度，对于加速收敛至关重要。单独地降低方差策略往往需要大量的样本和计算。每一轮迭代之间的梯度差异往往较小，这为样本复用提供了潜在的可能。基于ReSTIR的思路，我们实现了一种复用多轮迭代之间梯度样本的算法。但ReSTIR并不能直接用到可微渲染中，这主要是由以下几个原因构成：1. 可微渲染的过程中需要同时计算大量场景参数的导数，而不是单个像素；2. 梯度的变化可能会同时影响多个像素；3. 梯度会有负值。我们通过在参数空间上重新表述可微渲染的积分形式来解决上述问题，我们开发了一种新的重采样估计用以处理负值函数。

## 1. 引言

精确计算图像对于场景参数的导数在逆向渲染中越发成为重要议题。现代可微渲染算法通过蒙特卡洛积分来计算这些导数。但过大的导数方差会导致逆向渲染的优化效率大幅降低。

迭代优化即是通过沿着损失函数的估计梯度进行步进来更新目标图像的估计。鉴于多轮迭代之间图像的变化很小，完全丢弃已经计算出来的样本梯度就显得非常浪费。

为了在可微渲染中用上ReSTIR，本文做出了以下贡献：

- 参数空间的可微渲染：可微渲染需要计算出损失函数对参数的导数向量，也就是参数空间的损失梯度。这需要重新制定底层的积分方式。
- 对正值函数和负值函数的重采样：将GRIS推广到可能为负值的函数上，我们开发了一种正值化技术用于重采样。
- 将上述方案应用到纹理优化上。

### 1.1 相关工作

本文的主要关注点在于渲染积分的连续部分的求导。前人的相关工作主要包括Zeltner的蒙特卡洛导数估计、Zhang和Yu等人的对偶采样（利用了导数的对称性）。本文则在梯度下降多次迭代之间进行样本复用，和前人的工作互不影响。

传统的自动微分会随着散射次数的增加而同步增加内存开销，而辐射度反向传播通过重新计算需要的数据解决了该问题。本文主要关注直接光照的影响，所以内存问题其实没有那么重要，但亦可以推广来解决间接光照问题。

Nimier-David等人提出过一种纹理空间的采样方法来使得视频中的材质重建收敛得更加均匀。本文所使用的参数空间形式和该方法高度相关，但主要用在不同迭代轮次的样本复用上。因此，我们需要导数计算的数学形式来定义一个适合重采样的目标函数。

## 2. 参数空间的可微渲染

基于物理的可微渲染通常在图像空间中对每个像素的渲染积分估计导数，这可以直观地计算出图像损失用于逆向优化。但对于结合ReSTIR的方法来说并不适合，所以我们直接在参数空间计算导数用于复用。

## 2.1 正向渲染和可微渲染

渲染方程：

$$I_j(\pi) = \int_{\Omega} h_j(\mathbf{x}) f_c(\mathbf{x}, \pi) d\mu(\mathbf{x})$$

其中， $f_c$ 是贡献度量函数。

渲染方程对场景参数 $\pi_i$ 求偏导可以得到：

$$\begin{aligned}\partial_{\pi_i} I_j(\pi) &= \partial_{\pi_i} \left[ \int_{\Omega} h_j(\mathbf{x}) f_c(\mathbf{x}, \pi) d\mu(\mathbf{x}) \right] \\ &= \int_{\Omega} h_j(\mathbf{x}) \partial_{\pi_i} f_c(\mathbf{x}, \pi) d\mu(\mathbf{x})\end{aligned}$$

需要注意的是，如果被积函数对场景参数的微分存在不连续部分，则被积函数可能包含狄拉克分布需要特殊处理。但本文着重关注被积函数对场景参数连续的部分。导数计算需要分别对不同参数计算，也就是对每个像素来说，其梯度为 $\partial_{\pi} I_j = (\partial_{\pi_1} I_j, \dots, \partial_{\pi_l} I_j)$ 。

可微渲染算法需要计算每个像素的全微分，所以直接应用ReSTIR就会引发一个问题：ReSTIR中，每个像素都需要将一个样本存在它的蓄水池中，所以存储开销与像素量呈正比。但对于可微渲染来说，每个像素具有场景参数个偏导数。当然，也可以只在蓄水池中存储对像素有贡献的参数导数，但在可微渲染之前确定哪些参数对像素产生影响是无法做到的。举例来说，穿过一个像素的路径可能会命中不同的场景参数（例如两个相邻的贴图纹理），那么求导数就需要分别对不同的场景参数求导；但如果重新整理成参数空间的可微渲染形式，那么只需为每个场景参数储存一个蓄水池，累积来自所有相关像素的导数估计。

## 2.2 从以像素为中心到以参数为中心的估计器

我们发现，在基于可微渲染的逆向渲染中，最终的目标量是损失函数关于场景参数的梯度向量，与像素无关。换句话说，可微渲染依赖于参数空间，而正向渲染得到的是像素。逆向渲染问题应该求解的是：

$$\hat{\pi} = \underset{\pi}{\operatorname{argmin}} \mathcal{L}(I(\pi), G)$$

其中， $\mathcal{L}$ 是一个可微的损失函数， $I$ 是渲染图像， $G$ 是参考图像。

我们使用基于梯度的优化方法来求解该问题，所以需要计算出损失函数对每个参数的梯度。

损失函数对不同参数的导数为：

$$\partial_{\pi_i} \mathcal{L} = \partial_I \mathcal{L} \cdot \partial_{\pi_i} I = \sum_{j=1}^n \partial_{I_j} \mathcal{L} \cdot \partial_{\pi_i} I_j$$

也就是伴随渲染 $\partial_I \mathcal{L}$ 和各像素对场景参数导数 $\partial_{\pi_i} I$ 的点积。

如果使用蒙特卡洛估计来计算上述点积，则要求图像 $I$ 和它关于场景参数的导数 $\partial_{\pi_i} I$ 之间是无关的，而且 $\partial_I \mathcal{L}$ 在 $I$ 中需要是仿射的，这样才能得到无偏的结果。好在常用的损失函数，例如 $l^2 - norm$ 以及relMSE都满足这些条件。

将渲染方程代入上式，整理即可得到：

$$\begin{aligned}
\partial_{\pi_i} \mathcal{L} &= \sum_{j=1}^n \partial_{I_j} \mathcal{L} \int_{\Omega} h_j(\mathbf{x}) \partial_{\pi_i} f_c(\mathbf{x}, \pi) d\mu(\mathbf{x}) \\
&= \int_{\Omega} \underbrace{\left[ \sum_{j=1}^n \partial_{I_j} \mathcal{L} \cdot h_j(\mathbf{x}) \right]}_{\mathbf{w}(\mathbf{x})} \partial_{\pi_i} f_c(\mathbf{x}, \pi) d\mu(\mathbf{x}) \\
&= \int_{\Omega} \mathbf{w}(\mathbf{x}) \partial_{\pi_i} f_c(\mathbf{x}, \pi) d\mu(\mathbf{x})
\end{aligned}$$

其中， $\mathbf{w}$ 即表示为路径 $\mathbf{x}$ 对于伴随渲染和局部像素滤波的权重，它隐藏了像素的求和操作。我们称上述积分分为**参数空间的可微渲染方程**。

上述方程形式意味着我们没有必要先计算出每个像素的导数，再将它们求和以得到损失函数的导数。我们可以直接计算积分来估计损失函数的导数。当对场景参数 $\pi_i$ 使用蒙特卡洛积分，且路径对场景参数的贡献非零时，路径 $\mathbf{x}$ 的贡献等于其微分 $\partial_{\pi_i} f_c$ 乘以权重 $\mathbf{w}$ （例如一个未被遮蔽的路径，它有一个在物体表面的顶点的场景参数是 $\pi_i$ ）。微分 $\partial_{\pi_i} f_c$ 表示参数的轻微改变对路径上辐射度的影响，权重 $\mathbf{w}$ 则描述了参数的轻微改变对损失函数的影响。

基于以上形式，每个场景参数只需要存储一个蓄水池，而不是每个像素对每个场景参数都需要一个蓄水池。

## 3. 对导数使用GRIS

### 3.1 RIS, GRIS 和 ReSTIR

对于独立候选样本池 $\{x_1, \dots, x_M\}$ ，可以通过一个正比于目标函数 $q$ 的分布 $p$ 来重新采样该蓄水池以得到服从 $q$ 分布的样本：

$$\langle F \rangle_{ris} = \frac{f(x_z)}{q(x_z)} \frac{1}{M} \sum_{s=1}^M \frac{q(x_s)}{p(x_s)}$$

其中， $f$ 是被积函数， $x_z$ 是重新采样出来的样本。如果对于任意的 $f \neq 0$ 都有 $p > 0$ 且 $q > 0$ ，则上述估计是无偏的，方差表示为：

$$V[\langle F \rangle_{ris}] = \frac{1}{M} V\left[\frac{f}{p}\right] + \left(1 - \frac{1}{M}\right) V\left[\frac{f}{q^*}\right]$$

其中， $q^*$ 表示归一化的目标函数密度。直观来说就是：随着候选样本数 $M$ 的增加，样本 $x_z$ 会更趋近于服从 $q^*$ 的分布。通常会选用一个近似于被积函数 $f$ 的 $q$ ，这样关于 $q^*$ 重要性采样得到的方差就会大幅低于对 $p$ 重要性采样得到的方差。

在**RIS**中，所选样本 $p_z(x_z)$ 的精确的概率密度是无法以闭合形式计算的。

为了增加每个像素的候选样本数，**ReSTIR**一文复用了时域和邻域的样本作为**RIS**的候选样本。为了避免存储所有的候选样本，**ReSTIR**使用了加权蓄水池采样（weighted-reservoir sampling, **WRS**），每个像素存有一个蓄水池，其中含有选中的样本 $x_z$ 、权重和 $w_{sum} = \sum_s w_s$   $w_s = q(x_s)/p(x_s)$ ，以及候选样本数 $M$ 。样本的选择在一个pass中完成，通过计算权重和 $w_{sum}$ 、增加 $M$ ，依概率 $w_s / \sum_{t \leq s} w_t$ 替换候选样本 $x_s$ 。

由于**ReSTIR**在时空邻域内复用了之前可能用过的样本，这引起了两个问题：

1. 由于 $p_z$ 无法计算，初始分布 $p$ 也变得难以处理；
2. 样本 $x_s$ 可能是相关的，也可能是来自于其他积分域的，但**RIS**假定了样本都来自同一个积分域；

为了解决以上问题，**GRIS**使用了如下的估计器：

$$\langle F \rangle_{gris} = f(y_z)W \quad \text{with} \quad W = \frac{1}{q(y_z)} \sum_{s=1}^M m_s(y_s) q(y_s) W_s \left| \frac{\partial T_s}{\partial x_s} \right|$$

其中,  $y_s = T_s(x_s)$ ,  $T_s: \Omega_s \rightarrow \text{dom} f$  是一个偏移映射 (shift mapping), 在被积函数域中将样本  $x_s$  映射到  $y_s$  (也就是将邻近像素上生成的路径转换到当前像素的路径上);  $|\partial T_s / \partial x_s|$  是这个映射的雅克比矩阵; 无偏贡献权重  $W$  是概率密度倒数  $1/p_z(y_z)$  的无偏估计。在后续的重采样中,  $y_s$  和  $W$  就可以用作  $x_s$  和  $W_s$ 。对于多重重要性采样 (MIS) 的重要性权重  $m_s$  而言, 也可以得到相应的平衡启发式表达:

$$m_s(y) = \frac{q_{\leftarrow s}(y)}{\sum_{t=1}^M q_{\leftarrow t}(y)}$$

不过, 使用  $q_{\leftarrow s}(y)$  来替换概率密度函数, 则它是第  $s$  个样本的原始域中估计目标函数在  $x$  上的值:

$$q_{\leftarrow s}(y) = \begin{cases} q_s(T_s^{-1}(y)) \left| \partial T_s^{-1} / \partial y \right|, & \text{if } y \in T_s(\text{supp } x) \\ 0 & \text{otherwise.} \end{cases}$$

## 3.2 正值化RIS

**RIS**和**GRIS**都只能用于正向渲染, 也就是被积函数  $f$  非负的情形下。当被积函数为  $\partial f$  可能为负值时, **RIS** 仍然可以作为一个有效的估计器, 但目标函数  $q$  则必须是非负的, 因为它表示一种未归一化的概率分布。一种可能的解决方案是使用永远非负的目标函数, 例如使用函数  $g$  来近似  $f$ , 并且将目标函数设为  $q = |g|$ 。该方法也可以实现无偏估计器, 但无法做到零方差 (即使  $q = |f|$  且  $M = \infty$ )。该 **RIS** 估计器的方差只收敛于从目标密度中精确抽取的单个样本的方差, 也就是  $V[f/|f|^*]$ 。换句话说, 由于被积函数存在负值, 方差永远无法为零。

本文使用正值化技术来解决该问题。正值化技术将被积函数拆分为正值部分和负值部分, 也就是  $f(x) = f_+(x) - f_-(x)$ , 其中, 我们定义  $f_+(x) = \max(f(x), 0)$   $f_-(x) = \max(-f(x), 0)$ 。通过构造出分别与  $f_+$ ,  $f_-$  成比例的概率密度函数  $p_+$ ,  $p_-$ , 就可以在两边分别得到方差为零的估计。

不过正值化技术使用起来并没有看上去那么简单。依据概率密度函数  $p_+$ ,  $p_-$  对任意函数采样并不容易做到。所以我们打算使用 **RIS** 来近似地解决该问题。那么, 通过对目标函数  $q$  使用正值化技术, 我们就可以得到一个正值化的 **RIS** 估计器 (**PRIS**):

$$\langle F \rangle_{pris} = \frac{f(x_{z+})}{q_+(x_{z+})} \frac{1}{M} \sum_{s=1}^M \frac{q_+(x_s)}{p(x_s)} + \frac{f(x_{z-})}{q_-(x_{z-})} \frac{1}{M} \sum_{s=1}^M \frac{q_-(x_s)}{p(x_s)}$$

其中,  $q$  是某种近似被积函数  $f$  的函数 (函数值可以为负值)。我们为正值和负值部分使用相同的概率密度函数  $p$  和相同的候选样本集 (由于候选样本要么为正值, 要么为负值, 要么就是零, 所以最多只能在其中一个估计器上有非零值)。

相似的, 我们也可以推导出正值化的 **GRIS** 估计器 (**PGRIS**):

$$\langle F \rangle_{pgris} = \frac{f(y_{z+})}{q_+(y_{z+})} \sum_{s=1}^M m_s(y_s) q_+(y_s) W_s \left| \frac{\partial T_s}{\partial x_s} \right| + \frac{f(y_{z-})}{q_-(y_{z-})} \sum_{s=1}^M m_s(y_s) q_-(y_s) W_s \left| \frac{\partial T_s}{\partial x_s} \right|$$

在 **ReSTIR** 和 **GRIS** 中, 选中的样本仍然会在后续重采样中复用。同时, 目标函数也可以在后续重采样中更迭。在我们的实现里, 选中的样本可能会在重采样阶段改变符号, 因此, 将选自正值估计器和负值估计器的样本用作两类估计器的候选样本就显得尤为重要。举例来说, 之前从正值估计器中选出的正值样本可能会在这次重采样中改变符号作为负值样本用于负值估计器。这额外的处理确保了所有的样本被两类估计器选中的概率都不为零。

**PGRIS** 的 MIS 权重  $m_s$  直接使用了 **GRIS** 中的权重, 唯一的区别在于对正值估计器和负值估计器是不同的 MIS 策略, 应当据此调整 MIS 权重:

$$m_s(y) = \frac{N_s q_{\leftarrow s, \pm}(y)}{\sum_{t=1}^M N_t q_{\leftarrow t, \pm}(y)}$$

其中,  $N_s$  是策略  $s$  的置信度, 将候选样本总数限制为最大是  $M_{cap}$ , 来约束复用样本的权重 (M-capping)。

## 4. 在纹理优化中的应用

我们选择了逆向渲染优化BRDF的参数纹理作为验证方式，所以参数数量和纹素数量成比例。虽然ReSTIR为每个像素存一个蓄水池来执行时空复用，但我们这里为每个纹素使用两个蓄水池（正值蓄水池和负值蓄水池）来复用梯度下降迭代中的样本。与ReSTIR不同的是，我们从相机生成候选样本，但将蓄水池按纹素存储，所以需要在对屏幕像素并行处理时，为更新蓄水池使用同步方法，因为来自不同像素的多个候选样本可能都对同一个纹素有非零的贡献。

### 4.1 候选样本的生成

虽然为每一个纹素直接生成重采样的候选样本是可行的，但并不是所有的纹素都对相机可见。为不可见的纹素生成样本是浪费的，因为它们对图像损失并没有贡献。据此，我们提出使用从相机发出的光线追踪来生成候选样本，并在所有纹素间共享使用。这样就相当于拥有了像素数（ $n$ ）乘以spp（ $m$ ）的候选样本数量。那么，对参数 $\pi_i$ 的估计器即可表示为：

$$\langle \partial_{\pi_i} \mathcal{L} \rangle = \frac{g_i(x_z)}{q_i(x_z)} \frac{1}{M} \sum_{j=1}^n \sum_{k=1}^m \frac{q_i(x_{j,k})}{p_j(x_{j,k})}$$

其中，样本总数 $M = mn$ ，样本 $x_{j,k}$ 是从第 $j$ 个像素上生成的第 $k$ 次路径的样本。

$g_i(x) = \mathbf{w}(x) \partial_{\pi_i} f_c(x)$ 则是对参数 $\pi_i$ 的参数空间被积函数。

虽然对于一个特定的纹素，只有少数几个 $x_{j,k}$ 具有非零贡献，但得益于贡献候选样本，生成这些候选样本的开销已经分摊到所有参数上了。

目标函数。我们将目标函数 $q$ 定义为参数空间被积函数 $g$ 的亮度值（也就是一种可以负值的RGB）。值得注意的是，在只有单一分布的RIS中，将目标函数设置为被积函数并不能带来收益，这时RIS就会退化为关于分布 $p$ 的重要性采样。由于我们选中的样本会作为后续重采样的候选样本，所以候选样本的分布 $p$ 会逐渐收敛到目标分布 $q$ 。因此，我们的估计器也会在多轮迭代中持续改进。

### 4.2 样本复用

我们将不同的迭代轮次看作不同的帧进行时间上的样本复用。我们在每一轮梯度下降迭代中复用上一轮迭代中的样本。

从当前轮次和上一轮次中重新采样候选样本需要混合它们的蓄水池。由于计算MIS权重需要计算上一轮的目标函数，所以我们还需要将每一轮的纹理缓存下来。

为了复用上一轮的样本，我们选用了随机重演的shift-mapping方案。该方案复制已经用过的随机数来生成符合要求的样本。选用该方案主要是出于它的简洁性和泛用性。随机重演的一个缺点是计算开销较大，但仍然能带来很高的收益。

## 5. 总结与展望

我们创新地将ReSTIR应用到了可微渲染中来复用梯度下降过程中的样本。

局限性：我们假定了多轮迭代之间的梯度信息是足够相关的，但当学习率很高的时候就未必适用了，因为这种情况下相邻两轮迭代的参数可能发生很大的变化，从而导致梯度也发生很大的变化。

展望：跨参数复用，类似于ReSTIR中的空间复用，在参数空间也是可行的。另外，在逆向渲染优化中使用具有相关性的梯度对最终结果的影响也很值得研究。最后，我们的PGRIS估计器也不局限与渲染领域，也可以用于其他的领域（被积函数可能存在负值的情况）。