

Blockwise Multi-Order Feature Regression for Real-Time Path Tracing Reconstruction

Dezeming Family

2022 年 12 月 15 日

正常字体：表示论文的基本内容解释。

粗体：表示需要特别注意的内容。

红色字体：表示容易理解错误或者混淆的内容。

蓝色字体：表示额外增加的一些注释。

绿色字体：表示额外举的一些例子。

目录

一 Introduction 和相关工作	1
二 重建管线	2
2 1 输入	2
2 2 预处理	2
2 3 Blockwise Multi-Order Feature Regression (BMFR)	3
2 4 使用随机正则化的特征拟合	3
2 5 后处理	4
三 复杂性分析	4
四 特征 buffer 选择	4
五 建立测试和结果	5
六 文中方法的不足	5
七 能够带来的启发	5
参考文献	5

abstract

一篇关于蒙特卡洛路径追踪的后处理去除噪声的论文。

基于回归的噪声滤波方法 (regression-based noise filtering) 已经用于离线渲染, 这种方法计算消耗较大。本文提出了一个新的基于回归的重建管线, 叫做分块多阶特征回归 (Blockwise Multi-Order Feature Regression, BMFR), 该方法用于路径追踪 1spp 输入, 并且运行是实时的。

通过快速实现增强 QR 因子分解 (augmented QR factorization) 和使用随机正则化 (stochastic regularization) 来解决秩不足的特征 (rank-deficient feature) 数据, 实现了高速。所提出的算法比先前最先进的实时路径跟踪重建方法快 1.8 倍, 同时产生质量更好的帧序列。

一 Introduction 和相关工作

即使硬件水平上升, 渲染也逐渐对分辨率和刷新率有越来越高的要求, 因此, 实时应用中每帧只采样 1spp。实时去噪不但需要去除每帧噪声, 还需要消除帧与帧之间的 flickering 噪声。

本文的贡献在于:

- 一个新颖的 Blockwise Multi-Order Feature Regression (BMFR) 算法, 特征 buffers 的多个多阶的版本被用于拟合。
- BMFR 算法的快速 GPU 实现。
- 提出可以使用随机正则化来解决分块特征可能的秩不足, 避免数值不稳定性, 而不需要额外的 pivoting 复杂性。

基于回归的方法通常执行时间以秒为单位 (Moon et al. 2016), 并且被认为仅适用于离线环境 (Alla Chaitanya et al. 2017; Schied et al. 2017)。然而, 我们以一种不寻常的方式进行回归 (分块处理、带有随机正则化的增强因子分解), 因此, 所提出的方法是第一次实现实时性能的基于回归的方法。

路径追踪重建方法的综述可以见 (Zwicker et al. 2015), 大致分为三大类, 离线方法、可交互方法和实时方法。

离线方法肯定效果是最好的, 但是它的速度与计算图像着色样本的时间有很大关系。离线方法一般都需要目标输入大于 1spp, 大致有三种方法: (1) 基于双边或者引导滤波; (2) 基于神经网络; (3) 基于拟合无噪声特征 (Bitterli et al. 2016; Moon et al. 2014, 2015)。

剪切滤波也能够实现可交互帧率 (Yan et al. 2015), 且剪切滤波支持当特征 buffer 有噪声时的去噪, 比如运动模糊。

引导滤波可以实现实时交互 (Bauszat et al. 2011), 但是对于 GPU 应用并没有很多吸引力, 因为

因为它需要几十次移动窗口操作或生成同样多的区域求和表 (summed-area tables)。移动窗口操作涉及的并行工作比现代 GPU 可以同时处理的并行工作少几个数量级, 而生成区域求和表需要昂贵的并行扫描模式和存储在 buffers 中的更高精度值。

稀疏双边滤波器 (Mara et al. 2017); 多次迭代的层次滤波器 (hierarchical filter) (Burt 1981) 可以扩展到边缘停止函数 (edge-stopping functions) (Dammertz et al. 2010; Schied et al. 2017)。

对于时序稳定性, 可以应用重投影方法。如果能够找到当前像素在前一帧中的位置, 则重投影方法还可以支持刚体动画。时序稳定性可以进一步提高, 例如使用简单的时间抗混叠 (TAA) (Karis 2014), 其使用来自当前帧中像素附近的颜色来调整从前一帧采样的数据。在增强亚像素形态学抗锯齿 (Enhanced Subpixel Morphological Antialiasing) (SMAA) 中引入了在抗锯齿中使用时间数据的想法 (Jimenez et al. 2012)。

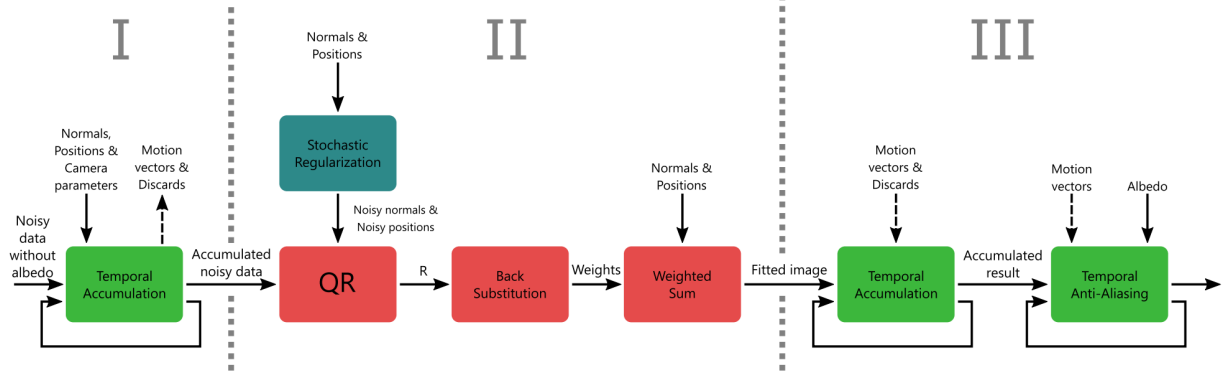
与之前的工作一样, 我们所提出的重建算法还利用 TAA, 并且还重新投影和累积来自先前帧的有噪声的数据。但是, 我们动态更改新的帧的权重, 以便遮挡后的第一个采样不会被过度加权。

此外, 我们在滤波后增加了额外的数据累积步骤, 以提高时间稳定性并避免伪影。此外, 我们使用基于回归的重建, 而不是使用双边滤波器的典型近似, 这在以前被认为对于实时用例来说太慢 (Alla Chaitanya et al. 2017; Schied et al. 2017)。通过应用增强 QR 因子分解和随机正则化, 我们使回归足够快, 以便实

时使用。最后，我们介绍了 BMFR，其中使用不同阶数的多个版本的特征 buffers 进行拟合，提高了拟合成功的机会。

二 重建管线

重建管线一般分为三个阶段：预处理、特征拟合和后处理。我们不需要对路径追踪方法有任何改变。



2.1 输入

输入 1spp 有噪结果，以及它的特征 buffers，albedo 使用 mipmapped 纹理。

然后从随机光源的随机点追踪阴影射线（应用多重重要性采样），之后继续追踪路径。每 1spp 样本包括了一条相机主射线采样和一条次级射线采样，一共两条阴影射线。采样使用 Wang hash(Wang et al. 2008) 生成随机数。之所以这么操作，是因为它可以实时跟踪路径，并且能够再现逼真的全局照明、柔和阴影和反射等效果。每当我们在本文中引用 1spp 数据时，我们都会引用此光线配置。

在输入去噪管线前，将第一次射线反弹时的 albedo 移除，因为要保证第一次反弹时的纹理不能被去噪器模糊。我们不将直接照明与间接照明分开（有些比如 (Bauszat et al. 2011; Mara et al. 2017) 会这么做），而是一起去噪。

如果场景包含多层材质，则必须为每个材质的照明组件单独运行去噪管线。然而，如果每一层的 albedo 相同，则所有 illumination 分量都可以同时组合和过滤。多层材料的一个优化机会是计算不同 albedo 和 illumination 的加权和，并同时过滤所有 illumination(Schied et al. 2017)。即使在滤波之前组合 illumination 不会产生物理正确的结果，该方法也可以用作快速近似。

2.2 预处理

该阶段对应上图 I，包括 1spp 数据的时序积累，把先前积累的数据重投影到新的相机帧，世界空间坐标和着色法向量被用于测试是否可以借助先前的数据。或者不得不只用当前帧 1spp 的数据。大多数像素的有效样本量都是大于 1spp 的。

我们跟很多去噪器一样都使用指数移动平均 (exponential moving average)，80% 的比例给过去帧，20% 的比例给当前帧。然而我们应用了一个显著的修改：我们首先计算样本的累积移动平均值（比如经过了 3 帧，获得了 3spp，就是这三帧直接取平均），并且仅在新样本的累积平均移动权重小于 20% 之后才使用指数移动平均值。在第一帧和 occlusions（即重投影失败）之后使用规则平均值确保第一个样本不会获得过高的权重，并且将权重限制为最小 20% 确保旧数据的影响逐渐消失。

我们对历史数据进行双线性采样，并对每个像素都进行丢弃测试，最终颜色仅仅与接受的样本有关，被丢弃的样本不会影响到最终颜色。积累的采样数也要被双线性采样，然后 round 到最近的整数值（采样数用于计算累积移动平均）。

比如我们某个当前估计值为 L_t 的像素 p 重投影到过去帧得到了历史样本，然后经过线性插值采样到了一个有效的历史样本 L_{t-1} ，采样到该历史样本是积累了 2.3pp 的结果，那么先 round 到整数，也就是 2pp，意味着历史像素是 2spp 的累积值，那么当前像素值应该用累积平均，即 $(2 \times L_{t-1} + L_t)/3$ 。

2.3 Blockwise Multi-Order Feature Regression (BMFR)

特征拟合阶段是上图 II，对于不重叠的块进行特征回归操作。这里我先把所有公式都列出来，因为懒得打了。

$$T = [F_{n_1}^{\gamma_1}, \dots, F_{n_m}^{\gamma_m}, \dots, F_{n_M}^{\gamma_M}], \quad (1)$$

$$\hat{\alpha}^{(c)} = \underset{\alpha^{(c)} \in \mathbb{R}^M}{\operatorname{argmin}} \sum_{(p,q) \in \Omega_{i,j}} \left(Z^{(c)}(p,q) - \sum_{m=1}^M \alpha_m^{(c)} F_{n_m}^{\gamma_m}(p,q) \right)^2, \quad (2)$$

$$\hat{Y}^{(c)}(p,q) = \sum_{m=1}^M \hat{\alpha}_m^{(c)} F_{n_m}^{\gamma_m}(p,q). \quad (3)$$

$$\tilde{\mathbf{T}}^{(c)} = [\mathbf{T}, \mathbf{z}^{(c)}] \quad (4)$$

$$\mathbf{R} \hat{\alpha}^{(c)} = \mathbf{r}^{(c)}, \quad (5)$$

$$\hat{\mathbf{y}}^{(c)} = \mathbf{T} \hat{\alpha}^{(c)}, \quad (6)$$

$$\tilde{\mathbf{T}}^{(c)} = [\mathbf{T} + \mathbf{N}, \mathbf{z}^{(c)}], \quad (7)$$

$$T = [\mathbf{1}, \mathbf{n}_x, \mathbf{n}_y, \mathbf{n}_z, \mathbf{w}_x, \mathbf{w}_y, \mathbf{w}_z, \mathbf{w}_x^2, \mathbf{w}_y^2, \mathbf{w}_z^2], \quad (8)$$

$F = [F_1, \dots, F_N]$ 表示一系列可用的无噪声特征 buffers，例如世界空间位置和着色法向量。

通常，这些 buffers 是路径跟踪中得到的信息，然而它们也可能包含人工创建的数据，如梯度。 F 中的每个 buffer 具有与有噪声的着色结果相同的分辨率。我们考虑 M 个特征 buffers 的扩展集合 T ：

$$T = [F_{n_1}^{\gamma_1}, \dots, F_{n_1}^{\gamma_2}, \dots, F_{n_M}^{\gamma_M}] \quad (\text{二.1})$$

其中 $\gamma_1 = 0$ ，当 $m > 1$ 时 $\gamma_m > 0$ ，表示指数，该指数不一定是整数，而且可以是小于 1 的小数。第一个 buffer，也就是 $F_{n_1}^{\gamma_1} = F_{n_1}^0 = \mathbf{1}$ 。

在一个图像块内，像素的绝对坐标集表示为 $\Omega_{i,j}$ ，块的位置表示为 (i,j) （注意这是块的位置，不是像素的位置）。对于应用多阶特征 T ，BMFR 可以表示为标准的最小二乘问题（见论文公式 (2)）。其中 $Z^{(c)}$ 是有噪路径追踪输入的通道 c （可以被时序积累），它可以是 R、G、B，也可以包括 luminance 或者 chrominance(色度) 组件。因此最后得到的对块 $\Omega_{i,j}$ 和通道 c 的估计可见公式 (3)，这里的 (p,q) 表示块内某个像素位置，可以看到，一个块内所有的像素共用同一组线性拟合参数。虽然是一个简单的线性解决方案，但它与特征 F 相比是非线性的，这使得它比基于 F 的线性回归的现有方法更灵活，能够更好地拟合数据。

2.4 使用随机正则化的特征拟合

Householder 的 QR 因子分解可以解最小二乘问题。我们把 M 个特征 buffers 进行 reshape（为了得到矩阵向量的形式），把它们作为长度为 W 的列向量， W 是一个块 $\Omega_{i,j}$ 内的像素数，令 \mathbf{T} 是 $W \times M$ 的矩阵，以及得到公式 (4) 所示的 $\tilde{\mathbf{T}}^{(c)}$ 。该矩阵尺寸是 $W \times (M+1)$ ，希望 $W \gg M$ ，意味着一个块内的像素数要比特征数更多。QR 分解后得到 $(M+1) \times (M+1)$ 的上三角矩阵 $\tilde{\mathbf{R}}^{(c)}$ 和 $(W) \times (M+1) \tilde{\mathbf{Q}}^{(c)}$ 的正交列矩阵。计算最小二乘解可以只用 $\tilde{\mathbf{R}}^{(c)}$ ，论文中做了一些观察，这些观察可以减少计算量。大家有兴趣可以看论文中给出的书链接 [2]，书中第 126 页有关于 Householder 的 QR 分解的介绍，需要根据 Householder 变换来实现。（

这里还有一些优化，比如发现式子在求解 $\hat{\alpha}$ 时可以用 \mathbf{R} （具体可见论文）时， \mathbf{R} 与 $\mathbf{z}^{(c)}$ 无关，所以对每个颜色通道不用都求去 \mathbf{R} 和 \mathbf{R} 的逆矩阵。想看懂这种方法可以具体参考一下 [2] 的实现过程，就能很容易理解为什么求解线性拟合参数可以做这些简化了。

注意 $\tilde{\mathbf{T}}^{(c)}$ 可能是非满秩的 (rank-deficiency) 因而无法因子分解, 虽然经常通过 pivoting 来管理, 但我们应用随机正则化, 也就是对输入 buffers 增加噪声 (这里增加的是 $W \times M$ 的平均值为 0 的噪声矩阵 \mathbf{N}):

$$\tilde{\mathbf{T}}^{(c)} = [\mathbf{T} + \mathbf{N}, \mathbf{z}^{(c)}] \quad (\text{二.2})$$

在增加噪声之前需要先把 \mathbf{T} 内的值缩放到范围 $[-1, 1]$ 之间。由于噪声的平均值是 0, 我们可以不用担心拟合会出现偏差。由于前面公式 (6) 中的 \mathbf{T} 是无噪声的, 所以噪声本身在重建结果中并不可见。噪声在 $[-\epsilon, +\epsilon]$ 之间均匀分布, 方差为 $\epsilon^2/3$ 。 $\epsilon = 0.01$ 比较好, 太大 (比如 1.0) 会导致过量或者过暗的块, 太小则起不到正则化的作用, 导致因子分解中出现除以 0 的情况。

2.5 后处理

后处理阶段 (III) 为了增加时序稳定性和感知的视觉质量。

首先对拟合帧进行时序累积, 这是为了减少在操作 BMFR 算法时的块 artifacts, 并且可以提高时序稳定性。此外当多帧累积时, 由于随机正则化导致的拟合误差可以被消除 (这也与噪声均值为 0 有关)。为了帮助降低 blockiness (块与块之间有一些明显的不同), BMFR 在处理每个帧时, 并不是使用非重叠的块, 而是进行随机偏移, 这些偏移就可以阻止一定程度的 artifacts (比如两帧中重用相同的块位置, 且相机和场景也是静态的, 导致块与块之间的区别会逐渐加大)。第二次后处理更快, 因为它重用了之前的移动向量和丢弃决策。

在第二次时间累积中, 我们使用 10% 的新数据和 90% 的旧数据, 因为这些值隐藏了块位置的变化。与第一次时间累积类似, 我们使用累积移动平均值, 直到新样本的权重达到选定的 10%。

在第二次时间累积中使用累积移动平均值是至关重要的, 因为在遮挡之后拟合的第一个块更可能包含离群数据 (意味着它周边都能有历史样本, 而就它没有, 这样它无法进行时序混合, 因此可能结果会与周边像素有较大差别), 并且使用累积移动均值, 它与后续帧更快地混合。例如, 如果我们使用指数移动平均值, 那么在三帧之后, 第一个拟合数据的权重仍然超过一半。累积移动平均值的权重与常规平均值相同, 是三分之一。

之后, 再进行 TAA 抗锯齿。

三 复杂性分析

前两个阶段很容易并行化, 像素之间彼此无关。

了解 QR 阶段的计算复杂性最重要, 当像素数上升时, QR 块的数量也会上升, 这些块之间并不互相影响, 所以它们都是可以并行处理的。

但是, 如果添加了一个特征 buffer, 则它必须由之前的所有特征 buffers 进行转换, 转换要求来自 Householder 反射法。需要转换的数量级是 $O(M^2)$ 。但是对每个特征 buffer 的工作量都很小, 导致随着特征数上升, 总时间是线性增长的 (参考论文中的图 3)。

四 特征 buffer 选择

特征越多, 计算时间越多, 但是质量提升会更多。

我们进行了一些评估, 来找作用更高的特征用于拟合。而且由于世界坐标对于产生软阴影更有帮助 (见论文图 6), 所以它的二阶形式也用作了特征。最终的特征向量 T 可以见公式 (8)。

在所提出的方法中, 镜面高光是从恰好具有与高光相似的数据的特征 buffer 生成的。如果可用的特征数据没有很好地呈现高光, 那么当累积来自连续帧的多个块的位置时, 结果会得到改善。将材质粗糙度添加到特征 buffers 中可以使得块内的区域之间的照明发生变化, 这仅在存在具有精细细节的粗糙度纹理的材质时才有帮助。然而, 如果块内区域具有均匀的粗糙度, 则常量特征 buffer 就会产生和加入了纹理粗糙度相同的结果。

五 建立测试和结果

关键是结果之间的对比，这里并没有太多需要刻意注意的地方。

ONND 很容易产生 reference 里都没有的细节。

由于多孔特性 (A-Trous), SVGF 很容易出现光伪影 (参考图 7 的 Living room 场景)。

GF 方法容易将 illumination 给模糊，这可能是由于参数选取不当造成的 (但用了原文中作者给出的参数)。

六 文中方法的不足

(1) 由于块的尺寸是固定的，所以对于很小的阴影 (比块小很多) 则不容易渲染出来。有些高光也是，但是可以加一些材质信息作为特征，就能得到很好的效果。

(2) 本文提出的算法容易与先前使用重投影的方法一样容易受到影响，因为重投影点是相机射线与场景中的第一个交点，如果存在镜面高光，就会被模糊。如果是完全镜面的材质可以用虚拟的世界空间坐标，但是如果同时存在镜面反射和漫反射组件，则就不得不把它们分开存储和重投影。另外由于遮挡，会导致重投影失败，导致可用的样本量下降 (图 11(a))。重投影还会导致在光源是移动的情况下，导致光照的延迟。(Schied et al. 2018) 使用光照变化的梯度，但是如果它直接用在我们的算法中，当光快速变化时，就会导致块 artifacts。

(3) 分块的特性会带来块伪影，例如图 11(b)，可以通过运行阶段 II 两次，每次用不同的分块方法，然后将两次的重叠像素区域融合。然而，随着相机逐步变化，这种块效应就会逐步被淡化。如图 11(c)。分块也会使亮度被过渡，比如图 13 的使用 BMFR 去噪以后，红框里本该黑的墙会有一块有些亮。

七 能够带来的启发

本文源码可以见 [1]。

本文我觉得最大的优势就是实现了实时 (当然用多阶特征来获得更好的性能也算是一个创新点吧)。实现实时的方法主要是使用了加速的 QR 因子分解和随机正则化 (随机正则化仅为避免非满秩矩阵导致的数值不稳定性)。

本文是一篇很常见的 SIGGRAPH 套路文，引入一种数学方法，然后适当地解决一些存在的问题，然后就把论文发出来了。该论文实现上其实并不是很困难，我觉得主要是后面的对比实验比较麻烦。同一个块内一起做线性拟合得到同一组参数，这种方式就相当于用多个像素进行滤波。所幸因为有无噪声纹理，所以结果不会很模糊。如果纹理细节并不明显且有高频 illumination，那么效果其实也很难做到很好。

至于有什么启发性的思考，大概可以根据论文提到的文中方法的不足选择一些适当的方式进行优化。但拟合的方法本身我觉得能达到的效果有限，优点在于实时性，缺点在于模型简单，效果有限。

参考文献

[1] <https://github.com/maZZZu/bmfr>

[2] Michael Heath. 1997. Scientific Computing. McGraw-Hill.