

Adaptive Polynomial Rendering

Dezeming Family

2022 年 12 月 16 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 重建框架	1
三 自适应多项式重建	2
3.1 偏差和方差的表示	2
3.2 多阶段误差估计	3
3.3 能量保留的离群点移除	3
四 自适应多项式采样	3
五 时序扩展	3
六 启发	3
参考文献	3

abstract

一篇 MC 路径追踪去噪的文章。

把图像用多项式近似，多项式的阶是估计得到的，使得误差可以最小化。为了鲁棒地估计最优阶，提出了一个多阶段的误差估计过程，可以迭代地估计我们的重建误差。另外，提出了一种能量保持的离群点去除技术 (energy-preserving outlier removal technique)，以去除尖峰噪声，而不会在重建结果中造成明显的能量损失。

此外，我们在误差估计的指导下，自适应地将额外的射线样本分配给高误差区域。我们证明，我们的方法优于最先进的方法，通过局部定义多项式阶来控制重建偏差和方差之间的权衡，甚至不需要滤波带宽优化 (filtering bandwidth optimization)（这是其他最近方法的常见方法）。

一 Introduction 和相关工作

有些自适应采样技术，用来估计每像素的最优滤波带宽 [Li et al. 2012; Rousselle et al. 2013; Moon et al. 2014; Moon et al. 2015] 来最小化均方误差 (MSE)。从技术上讲，这些尝试可以被认为是计算偏差（由过度模糊引起）和方差（由欠模糊引起）之间的最佳平衡的优化过程。但这些方法都是用的近似的低阶函数，而高阶函数是对结果的更好的近似，就我们所知，这是第一次使用近似多项式函数用于渲染。主要贡献是：

- 我们用多项式函数对渲染图像进行局部逼近，每个多项式都用估计的最优阶数构造，这样我们的重建误差可以最小化。
- 提出了一个多阶段误差估计过程，以鲁棒地估计我们的重建误差和每个多项式函数的最优多项式阶数。
- 提出了一种新的能量保持离群点去除方法，以去除渲染图像中具有过度强度的尖峰噪声，同时最小化离群点去除引起的显著能量损失。
- 通过估计的重建误差，在图像空间上局部控制采样率。

多维自适应采样容易受到维度诅咒，即维度变高以后效率会急剧下降。

基于频率的重建方法可以用来很好的重建一些特定效果，比如景深、环境光遮蔽和运动模糊等。

基于图像空间的自适应采样方法关键在于防止边缘模糊。这些技术的问题在于难以控制滤波器带宽（这里的带宽就是比如去噪器的权重，是要给邻域像素高一些权重还是低一些权重；以及滤波器范围要大一些还是小一些）。我们的方法不用考虑如何处理滤波器带宽。比如，我们选择高阶函数（例如，三次函数）来局部精确地重建高曲率区域，而不是收缩滤波带宽；并且为平滑区域选择低阶函数（例如，线性函数）。

二 重建框架

全部公式如下：

$$\begin{aligned}
y(i) &= \mu(i) + \epsilon(i), \quad (1) \\
\mu(i) &\approx \nabla g(f_c)(f_i - f_c)^T + p(c) + \sum_{1 \leq a \leq k} \frac{\nabla^a p(c)}{a!} ((i-c)^a)^T, \quad (2) \\
\sum_{i \in \Omega_c} \left(y(i) - \alpha(f_i - f_c)^T - \beta_0 - \sum_{1 \leq a \leq k} \beta_a ((i-c)^a)^T \right)^2 & K_h(i). \quad (3)
\end{aligned}$$

$$\hat{y}_k(c) = e_1(X_k^T W X_k)^{-1} X_k^T W y, \quad (4)$$

$$\hat{y}_k = X_k(X_k^T W X_k)^{-1} X_k^T W y = H(k)y, \quad (5)$$

$$\hat{y}(i) = \sum_{j \in \Omega_i} K_h^j(i) \hat{y}_k^j(i) / \sum_{j \in \Omega_i} K_h^j(i), \quad (6)$$

$$\xi_c(k) \equiv \frac{1}{\sum_{i \in \Omega_c} K_h(i)} \sum_{i \in \Omega_c} K_h(i) (\hat{y}_k(i) - \mu(i))^2. \quad (7)$$

$$E(\hat{y}_k(i) - \mu(i))^2 = bias^2(\hat{y}_k(i)) + \sigma^2(\hat{y}_k(i)). \quad (8)$$

$$\begin{aligned}
E(\hat{y}_k(i) - \mu(i)) &= \sum_{j \in \Omega_c} H_{ij}(k) E(y(j)) - \mu(i) \\
&\approx \sum_{j \in \Omega_c} H_{ij}(k) \mu(j) - \mu(i), \quad (9)
\end{aligned}$$

$$\sigma^2(\hat{y}_k(i)) \approx \sum_{j \in \Omega_c} (H_{ij}(k))^2 \sigma^2(y(j)), \quad (10)$$

$$k_{opt} = \underset{k}{\operatorname{argmin}} \sum_{i \in \Omega_c} K_h(i) ((E(\hat{y}_k(i) - \mu(i)))^2 + \sigma^2(\hat{y}_k(i))). \quad (11)$$

$$E_t(\hat{y}_k(i) - \mu(i)) \approx \sum_{j \in \Omega_c} H_{ij}(k) \hat{y}_{t-1}(j) - \hat{y}_{t-1}(i), \quad (12)$$

$$\sigma_t^2(\hat{y}_k(i)) \approx \sum_{j \in \Omega_c} (H_{ij}(k))^2 \sigma_{t-1}^2(y(j)), \quad (13)$$

$$\hat{y}(i) = \hat{y}(i) + \rho_o \hat{y}(i), \quad (14)$$

$$e_o = \sum_{i \in \Omega_o} \rho_o \hat{y}(i). \quad (15)$$

给定输入图像函数 $y(i)$, $i \in \mathbb{R}^2$ 是在图像的二维位置上的, 假设统计模型见公式 (1)。 $\mu(i)$ 表示 ground truth, $\epsilon(i)$ 表示 MC 噪声, 也就是方差其平均值为 0。我们把未知图像函数分解为 $g(f_i)$ 和 $p(i)$ 使得 $\mu(i) \equiv g(f_i) + p(i)$ 。其中, $g(f_i)$ 是一个线性函数, $f_i \in \mathbb{R}^8$, 即法向量占三维、纹理占三维, 深度占一维, 可见性占一维。第二个函数 $p(i)$ 是一个二维函数, 输入像素位置作为输入。我们将图像函数近似为与图像特征相关的函数 f_i 和未知的图像函数 $\mu(i)$ 的组合。当特征 f_i 并不是很有用时, 例如光泽反射, 我们可以用一个二维多项式函数 $p(i)$ 来近似残差项 $\mu(i) - g(f_i)$ (因为本质上这个残差项就是一个二维图像函数)。

根据公式 (2), 把未知函数定义为泰勒展开多项式。注意泰勒多项式里不需要 $g(f_c)$, 因为模型里已经有一个常数项 $p(c)$ 了 (根据泰勒展开, $g(f_c)$ 也是常数项, 所以两项合并了)。给定一个明确的阶数 k , 两个函数的系数可以用一个局部窗 Ω_c 来优化, 见公式 (3)。

公式 (3) 中的 $K_h(i)$ 是核函数, 表示滤波权重 (比如高斯权重), 其中 h 表示滤波带宽, 用于控制偏差-方差的平衡。我们可以对每个像素都计算最小二乘解, 也可以重建多个值。

注意如果 $g(f_i)$ 和 $p(i)$ 是空值或者零阶函数 (常量函数), 那么这个形式就是简单地高斯滤波或者双边滤波了。

图 2 表示不同的多项式阶数的作用, 更高阶多项式可以更好地保留边缘, 但是对噪声的抑制会变小, 因此我们需要一种数据驱动的方法来选择阶数。

三 自适应多项式重建

我们的方法是基于块的, 而不是基于像素的。定义一个对于中心像素 c 的多项式重建误差 ($L2$ 误差), 见公式 (7)。定义该公式, 就可以选择一个最优阶 k_{opt} 使得最小化误差 $\xi_c(k_{opt})$ 。然而由于计算误差时需要用到图像值 $\mu(i)$, 所以无法直接计算。所以需要去估计该误差。因此我们把重建的偏差和方差用数学来表示。

3.1 偏差和方差的表示

我们把误差表示为方差和偏差的和 (类似于 Rousselle et al. [2011]), 见公式 (8)。其中, 偏差可以用公式 (5) 中的 $\hat{y}$ 矩阵 $H(k)$ 来计算, 见公式 (9)。这里会假设 $E(y(j)) = \mu(j)$ 即假设图像是无偏的 (有噪, 但无偏), 对于光子映射来说这只是一中近似, 但我们会假设这里都是传统的路径追踪结果, 因此输入是无偏的 (我觉得论文这里扯上光子映射是有点碰瓷和不明所以)。

对于某个详述 i 的偏差, 直觉上来说, 第 i 行 $\hat{y}$ 矩阵中的元素可以理解为是邻域像素的权重 (参考公式 (9)), 因此偏差都是取决于 $H(k)$ 的, 并且当改变多项式阶数 k 时, 矩阵 $H(k)$ 也是在变化的。

方差可以近似地写为公式 (10)。因此我们可以用公式 (9) 和 (10) 来合并成公式 (11), 即选择 k 来最小化偏差和方差。

这个最优化需要利用未知量 μ 和 σ 。我们需要去估计这些量。

3.2 多阶段误差估计

直接用 MC 渲染结果 $y(i)$ 去估计偏差和方差也是无偏估计，但是噪声过大。我们采用的是一种多阶段迭代式误差估计，见公式 (12)。

在第一次迭代时，就是使用 MC 估计的结果，求出一个 k ，使用该 k 进行估计，得到 $\hat{y}_1(i)$ ；第二次估计使用 $\hat{y}_1(i)$ 来估计出新的 k ；以此类推。这个过程一直迭代，直到结果稳定。估计方差也是用类似的方法。

我们是用最优阶来估计方差，又需要方差来得到最优阶。为了避免这种鸡生蛋蛋生鸡的问题，我们简单地使用迭代得到的方差来计算最优阶，这样还能重用 $H(k)$ 。图 (3) 表示迭代的次数增加时，估计的最优阶 k_{opt} 会与用 reference 得到的最优阶越来越相似。

3.3 能量保留的离群点移除

如果中心像素比周围像素的平均值亮三倍，就认为是 outlier（离群点）。我们在计算重建开始之前就移除这种过亮的像素（只对过亮的像素进行中值滤波）。我们在移除时会存储能量损失，在最后会把损失的能量分配到更大范围的窗所在的区域中。该方法很直接，也很随意，我觉得算不上什么创新点。

根据图 (4)，(c) 是没有移除过亮点的结果，有很多 artifacts；(d) 是移除了过亮点的结果，但是没有恢复能量，所以显得有些暗。(e) 是我们的结果。

四 自适应多项式采样

本节介绍如何自适应地给一些高误差区域分配更多的采样 ray。我们主要使用的是 [Overbeck et al. 2009] 的方法。

第一次迭代我们使用均匀采样，然后使用我们的方差和偏差估计来重建和计算重建误差。由于我们最后的重建输出是通过加权函数 $K(\cdot)$ 局部平均得到的，所以误差也要用加权混合来计算。我们给误差大的地方分配更多样本。

特别是我们可以计算当分配更多样本时的 MSE 衰减率，具体时间怎么操作的我没有完全理解，但也是一种迭代方法，作者写了满满两大段。

讲道理昂，我觉得根据一个像素和周围相似像素（相同 albedo 或者材质粗糙度）的样本之间的方差（一个像素会有多个样本）就能分配自适应采样了。

五 时序扩展

为了保证时序稳定性，将 2D 滤波窗扩展为 3D，用来处理时序样本。我们把多个帧的有噪声数据排列成一个体数据，然后用多项式拟合。

六 启发

所谓拟合方法，难的是迭代和估计误差。有些迭代方法直接使用有噪样本，意图用有噪样本得到比较准确的最小二乘估计。但问题是这些有噪样本可能会很影响拟合函数的各个参数。

这里的误差估计方法还是很 trick 的，也就是“以毒攻毒”的感觉，用有噪声的数据去估计误差，然后根据估计的误差来生成参数，生成的参数用来计算去噪结果；而去噪结果反过来继续估计新一轮的误差。

遇事不决，迭代解决。想起之前自己投了一篇论文就是用的迭代法，有噪数据一步解决不了，就迭代两次；可惜不知道能不能中，就快要出结果了，先提前请个愿。

参考文献

- [1] xxx