Interactive Monte Carlo Denoising using Affinity of Neural Features

Dezeming Family

2022 年 11 月 30 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

abstract

针对蒙特卡洛渲染结果去噪技术，我们的方法可以运行在交互帧率上。我们在像素邻域中的特征上定义了一种新的成对相关性 (pairwise affinity)，从中我们组装了扩张的空间核 (dilated spatial kernels) 来过滤有噪声的辐射度。

由于两种机制（成对相关性和扩张空间核），我们的去噪器在时序上是稳定的。

首先，我们保持有噪声的辐射度和中间特征 (intermediate features) 的运行平均值，使用具有学习到的权重 (learned weights) 的每像素递归滤波器。第二，我们基于连续帧的特征之间的成对相关性使用小的时序核。

我们的实验表明，与具有可比计算成本的技术相比，我们的新亲和性导致更高质量的输出，并且有比内核预测方法 (kernel-predicting approaches) 更好的高频细节。我们的模型在低样本计数状态下（每个像素 2-8 个样本）匹配或优于最先进的离线去噪器，并以 1080p 分辨率的交互式帧速率运行。

相关内容：蒙特卡洛 (MC) 路径追踪、深度学习、蒙特卡洛光追去噪、指数移动平均滤波去噪。

一　介绍和相关工作

现有的对 MC 去噪的方式使用大核预测 (large kernel-predicting) 的神经网络，计算消耗很大，无法实时。实时去噪上也有手工设计的去噪器（结合双边滤波、中值滤波方案等）或者更紧凑的神经网络（例如 RAE），牺牲了图像质量。去噪的伪影通常在视频动画中被放大，因此去噪器必须注意能够产生没有相邻帧之间的高频伪影的时序稳定的结果。可交互去噪器中，那些使得光线追踪更有作用的效果，比如高光、折射和复杂的全局照明，去噪效果并不是很好。

屏幕空间基于采样的技术 (Sample-based image-space methods)。Gharbi 等人 [2019] 使用神经网络进行基于样本的去噪。他们预测每个样本的飞溅核 (splatting kernels)，并特别注意确保其网络对像素内样本的排列是不变的。他们的方法对于交互式渲染设置的缩放效果很差，因为网络评估成本随样本数量线性增长。Munkberg 和 Hasselgren[2020] 通过将样本划分并平均为固定数量的层、独立过滤层并把它们从前到后合成来缓解这一问题。我们的方法也处理独立的样本（它的意思是不像 RAE 那样将整个图像一起去噪），但将内核应用于每像素平均值而不是样本，这显著降低了运行时成本。

我们的神经网络是轻量级的，直接对路径追踪的结果信息去噪，将每个样本的信息汇总为一个每像素的低维的特征向量，然后把这些特征定义为成对的相关性。我们使用相关性在局部加权平均滤波的步骤中对相邻每像素辐射度值的贡献进行加权。与核预测技术相比，这些新的基于相关性的核导致了更好的去噪性能。

我们在一组不同的静态渲染和动画中评估去噪器。我们表明，与运行时成本相当的最先进的交互式去噪器相比，我们的去噪器产生了更干净的输出和更低的数字误差。对于低样本数渲染，我们的模型甚至匹配或优于最先进的离线去噪器，其计算消耗更低。

我们的这种新的滤波算法，使用从原始路径跟踪样本中学习的每像素深度特征的成对相关性来学习迭代 2D 扩展核。一种新的时间聚集机制，它使用相同的成对相关性来显著提高蒙特卡洛去噪的时序稳定性。

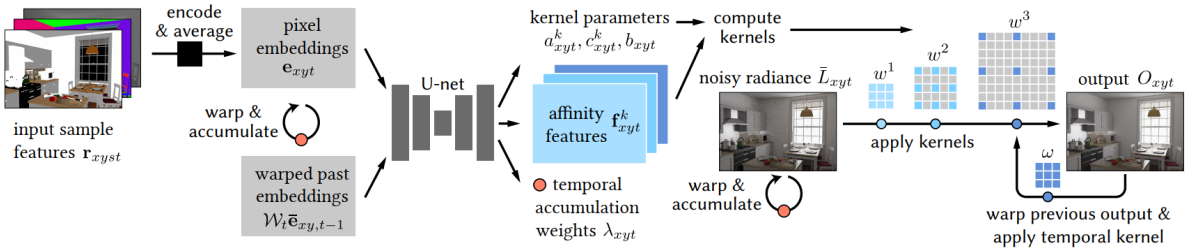
我们的去噪器也使用核，但我们的网络不是预测整个 (full-rank) 核，而是输出每个像素的特征向量，我们从中定义成对的相关性。这使得学习问题更简单，这导致了一个更简洁的模型，适合于交互应用。

Meng 等人 [2020] 将有噪声的辐射度扩散到双边网格中。沿着学习得到的的引导图在网格中进行切片，得到去噪输出，他们还使用具有固定权重的递归滤波器来实现时间稳定性。我们在第论文 3.7 节中讨论了神经双边网格和我们的相关性之间的相似性。我们的自适应时间累积提高了去噪性能，特别是在遮挡和镜面反射周围。我们基于相关性的时间内核进一步提高了时间稳定性，去除了低频噪声，并在时间上扩大了滤波器的足迹（即滤波器的滤波核覆盖范围）。

二　去噪算法

我们的方法的关键在于对学习到的特征的成对相似性度量，从中我们可以导出时空去噪核。

对于一系列帧，$t \in \{1, ..., T\}$ 表示帧索引；$s \in \{1, ..., S\}$ 表示一个像素内的样本索引，S 一般小于 8；(x, y) 表示像素坐标。每个像素的特征向量就可以表示为 $\mathbf{r}_{xyst}$。



上图，首先提取出每个样本的内在表示 (embeddings)$\mathbf{e}_{xyt}$，该步骤使用一个小的逐点网络 (pointwise network)。然后，我们对这些 embeddings 使用一个时序递归滤波器，使用自适应学习到的参数来传播帧之间的信息，然后将每个像素的 embeddings 分别取平均。

然后当前 embeddings 和 warped 过去的 embeddings（将过去帧的 embeddings 通过重投影到当前帧）被送入一个 U-net 网络（某种卷积网络，卷积可以使得邻域信息共享），用卷积网络来给像素预测一个参数向量。之后使用每像素参数来为每个像素构建滤波核，我们将内核分解为一系列 $K = 3$ 个扩张 2D 内核（类似于 Dammertz2010 年的多孔小波滤波核）。

扩张核滤波可以实现更大的空间足印，这有助于消除低频噪声，且与使用大型密集滤波核（将邻域全部样本都用来滤波）相比对性能的影响最小。时序内核是根据当前帧和先前帧的特征之间的相关性构建的；它使输出随时间平滑。

2 1　输入路径追踪样本特征

先前有的工作是使用样本，而不是把每个像素内的样本先平均，直接使用单个样本会带来更多计算负担。我们使用混合策略，使用每个样本的信息来计算滤波权重，但滤波器是作用在 box 滤波（其实就是把一个像素内的样本取平均）的像素辐射度 $\mathbf{L}_{xyt}$ 上的，而不是样本上。

每个样本都有 18 个渲染特征，组成一个特征向量 $\mathbf{r}_{xyst}$，我们把辐射度分为两个部分（diffuse 和 specular），这两个部分都被色调映射（$x \mapsto \log(1 + x)$）到更低的动态范围。色调映射只作用于输入到神经网络的特征，而去噪核是作用在原来的辐射度上的。我们把材质的粗糙度限定在一个固定的阈值下（PBRT-V3 中的 0.1 线性粗糙度）；使法向量（3 通道）和深度（1 通道）作为几何特征，再加上 albedo（3 通道）、粗糙度（1 通道），和四个二进制变量（true or false）：发射项 emissive，表示是否是光源；金属项 metallic，表示是否是金属；穿透项 transmissive，表示是反射材质还是折射材质；镜面反弹项 specular-bounce，表示相机路径第一个顶点是否是镜面交点。注意除了辐射度以外，其他所有的特征都是在第一个非镜面交点上计算得到的；辐射度毕竟是直接照射到人眼的部分，所以与第一个交点有关。

2 2　把样本映射到每个像素的特征

为了满足交互率，把对每个像素的处理都变小。使用一个很浅的全连接网络 (fully-connected, FC) 来将样本提取内在特征，输出的每个样本的特征在一个像素内取平均得到：

$$\mathbf{e}_{xyt} = \frac{1}{S} \sum_{s=1}^{S} FX(\mathbf{r}_{xyst}) \tag{二.1}$$

我们表明，这种提取内在特征的策略优于先前工作 [Bako 2017, Vogels 2018] 中使用的原始渲染缓冲区的简单的一阶和二阶统计数据。

2 3　时空特征传播

我们使用轻量级 U-net 处理每个像素的内在表示 (embeddings)。这个网络把当前帧和先前帧的 embeddings 作为输入，为每个像素产生一个相关性特征 (affinity features) 向量 $\mathbf{f}^k_{xyt}$（d 维，我们这里的维度是 8）和一个标量 a^k_{xyt} 和 c^k_{xyt}，其中，$k = 1, ..., K$。K 表示扩展核的扩展数量。

通过相关性特征之间的距离来得到空间滤波核，该滤波核通过带宽参数 a^k_{xyt} 来缩放，而 c^k_{xyt} 表示核的中心权重。U-net 也会输出两个额外的标量应用于时序去噪，b_{xyt} 描述时序去噪核之间相邻帧的特征相似性，λ_{xyt} 表示用于时序地积累像素 embeddings 和有噪声的 radiance 的指数移动平均滤波 (exponential moving average filter)。

$\mathbf{W}_t$ 是一个 warp 操作（将先前帧信息重投影到当前帧）：

$$(\mathbf{f}^k_{xyt}, a^k_{xyt}, c^k_{xyt}), b_{xyt}, \lambda_{xyt} = UNet(\mathbf{e}_{xyt}, \mathbf{W}_t\bar{\mathbf{e}}_{xy,t-1}) \tag{二.2}$$

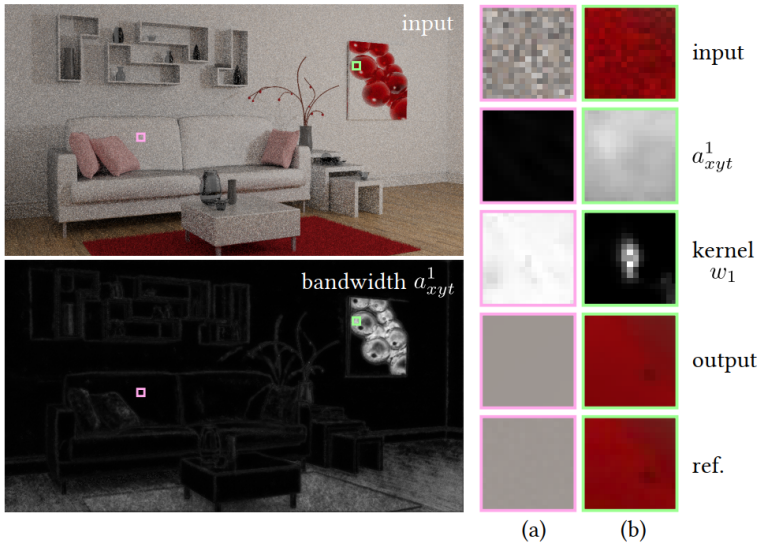
$\bar{\mathbf{e}}_{xy,t-1}$ 表示过去帧的历史积累值：

$$\begin{cases} \bar{\mathbf{e}}_{xy0} = \mathbf{e}_{xy0} \\ \bar{\mathbf{e}}_{xyt} = (1-\lambda_{xyt})\mathbf{e}_{xyt} + \lambda_{xyt}\mathbf{W}_t\bar{\mathbf{e}}_{xy,t-1} \end{cases} \tag{二.3}$$

注意 λ_{xyt} 是通过 sigmoid 来激活的，保证其值在 $[0,1]$ 之间。

2 4　应用成对相似性的空间核

其实就是一个双边滤波器，双边权重用相关性特征来计算。a^k_{xyt} 是取平方来保证其大于 0 的。a^k_{xyt} 趋近于 0 则表示当前更接近于 box 滤波器。下图是各个参数的可视化结果：



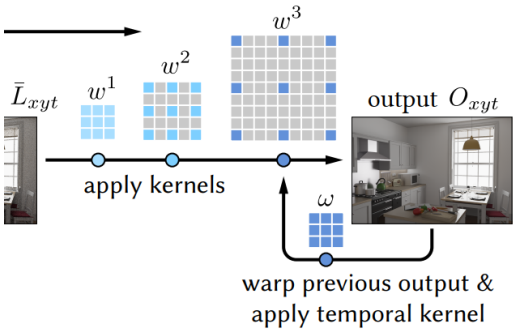
可以看到，粉色框中的 a^k_{xyt} 值都趋近于 0，所以更像是一个区域内的平均滤波；绿色框中的 a^k_{xyt} 都很大，所以倾向于双边滤波。

当 $c^k_{xyt}=1$，意味着中心像素可以对最终输出有很高贡献，而当 $c^k_{xyt}=0$，则网络可以跳过低样本采样中造成的比如过于明亮的区域。

2 5　时序稳定的去噪

对辐射度值 $\mathbf{L}_{xyt}$ 的时序滤波和 $\mathbf{e}_{xyt}$ 是相似的。通过 λ_{xyt} 将当前帧的辐射度和已经积累到当前帧的辐射度值进行组合。

时序滤波后的 $\mathbf{L}_{xyt}$ 再通过扩展核来空间滤波。在去噪的扩展核中，前两个步骤使用空间核，最后一个步骤通过重投影先前帧，同时使用时序核和空间核来滤波：



2 6 与核预测 (kernel-predicting) 网络的区别

如 Munkberg 和 Hasselgren[2020]、Vogels 等人 [2018] 所示，内核预测方法 [Bako 等人 2017] 需要更深更大的网络才能充分受益于更大的核，这是因为核内像素之间的成对相互作用的数量和复杂性随着内核大小的增加而增加。相比之下，我们的方法不需要网络“容量”的增加，因为我们用闭式的相似性来预测每个像素的特征，而不是整个核。

2 7 与神经双边网格的关系

神经双边网格 (neural bilateral grid) 来近似双边滤波，Meng[2020] 使用神经双边网格（3D 网格），前两个维度是屏幕空间坐标，第三维是学习到的标量参数（该参数与范围滤波器有关），而我们的特征 $\mathbf{f}^{\kappa}_{xyt}$ 是一个 8 维向量，维度更高意味着更有效。

三 数据集和训练步骤

相机自由穿过静态场景来生成动画，每个序列渲染 256×256 的分辨率。

使用 Gharbi 的场景生成器来生成不同的场景。

我们训练去噪器为了最小化重建误差、时序稳定误差以及对相似度参数进行正则化（论文公式 (9)）。使用 Symmetric Mean Absolute Percentage Error over the linear radiance (SMAPE) 来度量重建误差。

在训练中，整个流程必须都用自动微分机来实现，这样才能够利用反向传播来训练（也就是说后面的卷积操作都需要实现到神经网络中）。后面列出的参考网站 [1] 中介绍说，最后的扩展核可以理解为是一个空洞卷积，将辐射度图像 $\overline{\mathbf{L}}_{xyt}$ 使用滤波核参数来卷积，具体如何实现以及加速也是一个比较复杂的过程。论文中没有提供源码，因此复现过程还是比较复杂的。

参考文献

[1] https://blog.csdn.net/weixin$_4$5435206/$article/details$/122831297