**读书报告**

《Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting》是一篇在2020siggraph上英伟达发表的关于实时光线追踪技术的论文。

论文的背景和目的：

论文的研究背景是计算机图形学，具体是实时光线追踪技术。光线追踪是一种用于生成高质量三维图像的技术，它通过计算光线在三维场景中的传播，来模拟真实光照效果。然而，光线追踪的计算代价很高，无法满足实时渲染的要求。因此，论文的研究目的是提出一种新的实时光线追踪方法，即ReSTIR（reservoir-based spatiotemporal importance resampling，以提高实时光线追踪的效率，此算法通过高质量交互式的方式渲染这种照明情况，且无需使用复杂的数据结构。这种技术用于优化动态直接照明的实时光线追踪，以提高图像质量并减少计算复杂度。并且解决了使用蒙特卡洛从有数百万的动态光源的场景中渲染图片效率和质量较低的问题。

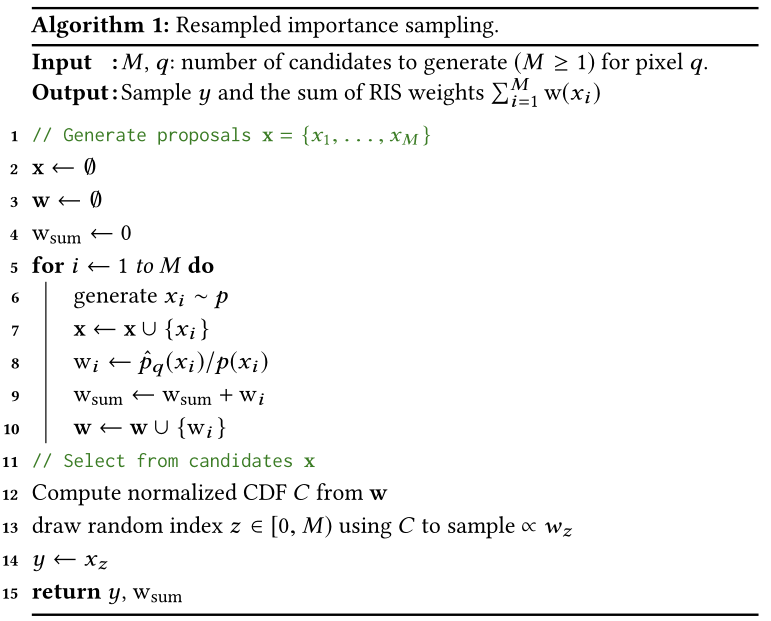
在传统的光线追踪技术中，光线追踪通常以固定的抽样率进行计算，即每个像素都会抽取相同数量的样本。这种方法通常在处理静态场景时是可行的，但对于动态场景，抽样率固定的方法可能会导致图像质量下降或帧率降低。ReSTIR技术解决了这一问题，通过在每个像素上动态调整抽样率来提高图像质量。这是通过创建一个“储水池”来实现的，其中储存了过去几帧图像中被抽样的所有光线的信息。在每一帧中，算法会根据当前像素的需要来从储水池中抽取足够数量的样本，以便在保证图像质量的同时最小化计算复杂度。

论文的内容和方法：

本篇论文是基于Weighted Resorvior sampling（WRS）算法对所渲染的画面进行重采样并在此基础上提出了ReSTIR算法的。这样的方法给实时光线追踪带来的极大程度的性能提升。而在过去WRS算法往往都是用在金融和统计学等领域上，这篇文章首次把WRS算法用在蒙特卡洛光追算法上。

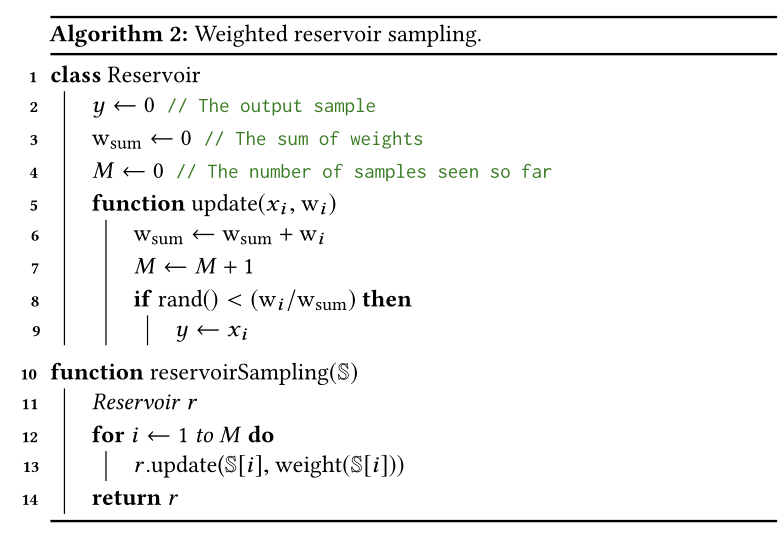
在本文中涉及到几个重要的算法：

1. Resampled Importance Sampling



这个算法也被简写为RIS算法，这是一种基于Importance Resamping并且通过这个过程获取数个样本，并使用这些样本进行蒙特卡洛积分的方法就是RIS算法。

2、Weighted reservoir sampling



这个算法也被简写为WRS算法，中文叫作加权蓄水池采样。这种采样方法，通过给定M个元素，并且从这些元素中随机的选取N个元素作为一个子集，N是 定义为一个小的常数，M 通常非常大，对应的数据无法本地存储，或者M个元素是一个数据流。

我们的蓄水池采样对每个元素只处理一次，并且存储器中只有N个元素必须保留在内存中。流的长度M不需要事先知道。

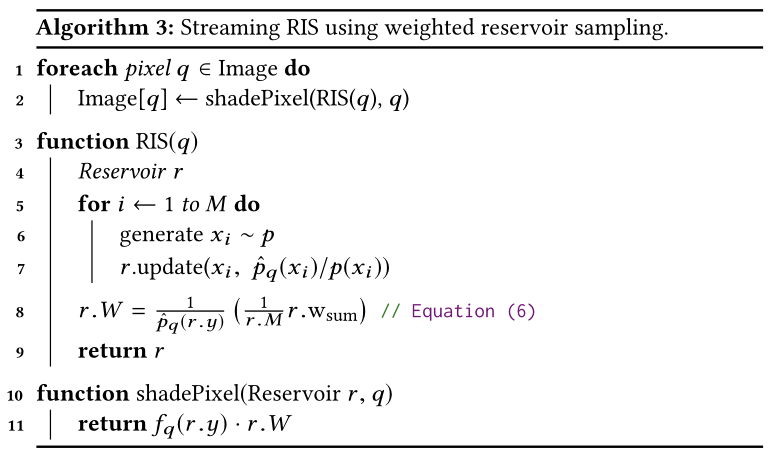
蓄水池采样顺序处理输入流的元素，存储N个样本的储层。在流中的任何一点，储层采样保持不变，即储层中的样本是从所需的分布中抽取的(对迄今为止处理的所有元素)。当处理流的过程结束时，算法就会返回。

储水池算法的更新规则随机地将储层中的替换为下一个样本。其服从于以下的概率

这样便能够确保以所需的频率出现在蓄水池中。因此，任何先前的样本都有概率在储层中。

这样的算法便是的我们可以将其运用于对画面渲染时的时间复用和空间复用，并且由于这个算法时依概率的并使用了权重，便使得其在运行时更加的高效。

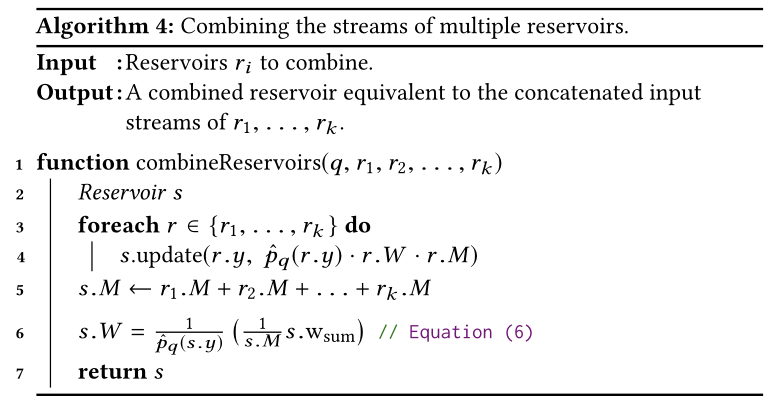
3、Streaming RIS using weighted reservoir sampling



这个算法时将WRS算法直接运用于RIS算法中所诞生的一种算法，其通过使用按顺序生成的候选样本和此样本对应的权重来更新其存储库。这个算法也是之后ReSTIR算法的进行对渲染画面的时间复用和空间复用的基础。

在论文中，作者们使用具有不同的候选样本数M的流式RIS以及使用了每个像素中有相同数量的光线的实时光追BVH[Moreau et al. 2019]进行了比较。在这个对比中，发现随着M的数值增加，流式RIS比当前的光线采样技术更加具有优势。而且其也不需要借助于预处理或者依赖其他复杂的数据结构。

4、Combining the streams of multiple reservoirs

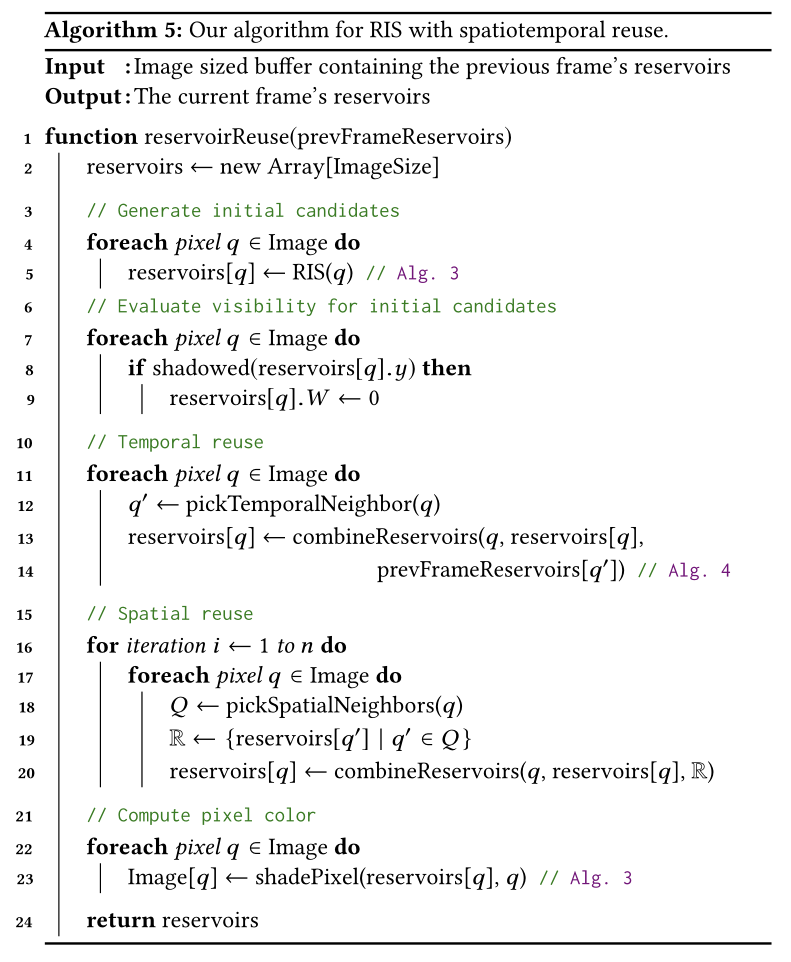


这个算法可以用于结合多个蓄水池的流式输入，通过这个算法，我们可以在执行对渲染画面的时间复用的时候，把此前的多个时间构成的多个蓄水池合并为一个；在进行对渲染画面的空间复用的时候，把某一像素周围的多个像素构成的多个蓄水池合并为一个。借此来更好的依概率的对时间复用和空间复用的过程进行采样的操作。

在空间复用的时候：储层采样的这一特性使得在RIS中重用计算的实用算法成为可能。我们首先使用Streaming RIS using weighted reservoir sampling算法为每个像素q生成M个候选，并将结果存储在图像大小的缓冲区中。在第二步中，每个像素选择k个邻居，并使用Combining the streams of multiple reservoirs算法将它们的蓄水池与自己的蓄水池结合起来。每个像素成本是O(k + M)，但每个像素有效地看到k·M个候选。至关重要的是，空间重用可以重复，使用先前重用传递的输出作为输入。执行n次迭代需要O(nk + M)次计算，但每个像素有效地产生个候选，并且要假设在每一步都使用不同的相邻像素。

在时间复用的时候：图像通常不是孤立呈现的，而是动画序列的一部分。在这种情况下，先前的框架可以为重用提供额外的候选框架。渲染一帧后，就可以存储每个像素的最终存储库，以便在下一帧中重用。如果我们按顺序渲染帧并前馈它们的蓄水池，帧不仅将候选帧与前一帧的候选帧结合在一起，而且将该序列中的所有先前帧结合在一起，这将极大地提高图像质量。

5、RIS with spatiotemporal reuse

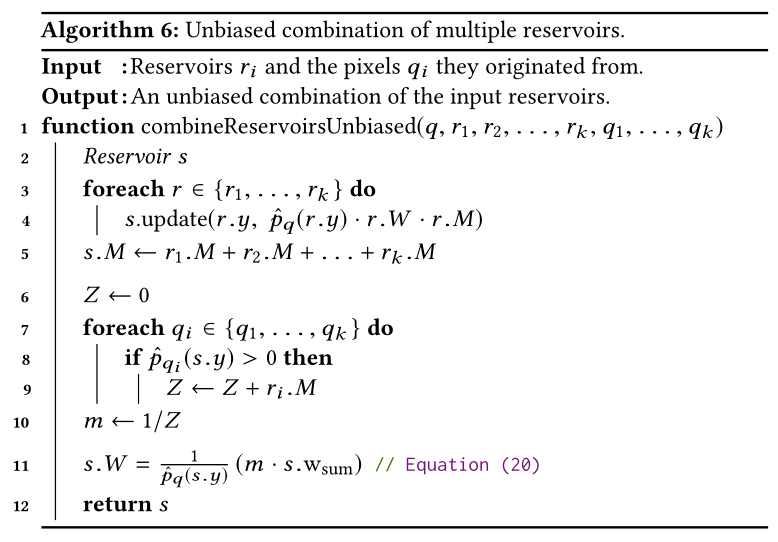


通过上述的四个算法这篇文章最终引出了最后的算法RIS with spatiotemporal reuse，这个算法让我们首先从M个独立的逐像素候选光中生成并重新采样。从这一步中选择的样本进行可见性测试，遮挡的样本被丢弃。然后，我们将每个像素中的选定样本与前一帧的输出结合起来，使用反向投影确定。我们执行n轮空间重用，以利用来自像素相邻像素的信息。最后，我们将图像着色，并将最终的蓄水池中的结果转发到下一帧。

以上便是ReSTIR算法的基础。通过以上的几个算法就可以实现对一个具有大量动态光源的复杂场景的渲染，其具有较高的渲染质量和效率。

对于此算法的拓展：

这个算法由于采用了时间服用和空间复用，在进行复用的时候，每个像素会使用来自不同域中的目标分布，并且使用这些来自于相邻像素中的候选可能会导致引入偏差，这是由于这些像素之间的分布是不同的所导致的问题，重采样之后的PDF会有所不同，因此就会导致算法出现偏差，是渲染出的画面产生较多的噪点。



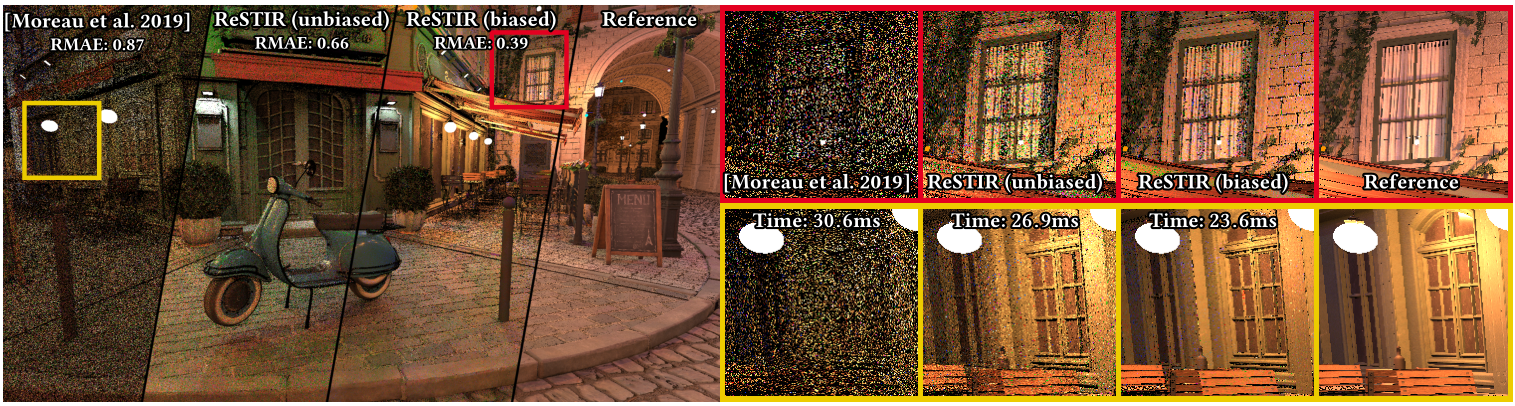
在此基础上提出了Unbiased combination of multiple reservoirs算法，通过此算法可以使得在合并多个输入流的时候保持无偏性。这个算法与Combining the streams of multiple reservoirs是类似的。但是，无偏的版本可能要使用更大的开销:如果我们使用可见性重用，评估它需要跟踪额外的阴影射线。这样的无偏算法，可以显著的提高渲染质量减少渲染时可能产生的噪点数量，但是其渲染的效率会有所降低，但根据论文中的实验，其渲染效率依旧高于现有的实时光线追踪渲染效率。

论文的结论和贡献：

本篇论文首次的将运用于金融领域和统计学领域的蓄水池算法，将其运用在计算机图形学的蒙特卡洛光线追踪的采样算法上。并利用了这一算法可以依概率的对输入到算法中的流式数据进行采样的特点，优化了算法在对光线采样时的空间存储开销和时间开销。

并通过实现，充分的验证了ReSTIR算法在渲染一个具有大量动态光源的场景中有显著的优势。

论文的优点和不足：



《Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting》是一篇质量很高的论文，其中提出的ReSTIR算法是原创性的贡献。本文使用了数个场景对ReSTIR算法进行实验，通过与Pierre Moreau在2019年提出的实时光线追踪的效果进行对比，可以发现其在一帧画面上的渲染时间耗时较少，并且在同一画面上产生的噪点也更少。即，使用此篇论文提出的ReSTIR算法可以在实时光线追踪上得到更好的渲染质量和效率。同时本篇论文其也解决了在一个场景下如果同时出现了大量了动态光源会让光追的渲染效果大幅度下降的情况。

但是这篇文章所使用的Weighted reservoir sampling.的依概率采样算法来实现了对画面的时间复用和空间复用，而这样的复用可能产生如下的问题，即，在复杂的渲染场景中如果出现场景中所含有的几何形体过于复杂或者场景中存在不连续的光照效果时，这样的服用算法可能会导致较差的画面渲染质量。