单次采样路径追踪与时空域降噪

黄霖, 刘世珩, 许天一

摘 要: 大量采样的路径追踪由于循环计算量大难以达到实时全局光照。而通过单次采样1 sample per pixel (1spp)的路径追踪再配合后期时空域降噪算法的图像处理可以近似实现高效高质量实时全局光照。路径追踪可以根据Lambert和GGX模型采用重要性采样分布模型随机选择一条光路。空域降噪需要利用G-buffer的双边滤波，而时域降噪可以采用时域超级采样算法（temporal super sampling）。渲染流程使用Visibility Buffer避免光线追踪瑕疵。

1 SPP Ray Tracing with Spatiotemporal Denoiser

Huanglin, Liu Shiheng, Group Member 3

**Abstract**: Due to the large amount of cyclic computation, it is difficult to achieve real-time global illumination for path tracing with a large number of samples. The high-efficiency and high-quality real-time global illumination can be approximately realized through the path tracking of one sample per pixel (1spp) in a single sampling and the image processing of the later spatio-temporal noise reduction algorithm. For path tracing, an optical path can be randomly selected by using the importance sampling distribution model according to Lambert and GGX models. Spatial noise reduction needs to use the bilateral filtering of g-buffer, while temporal super sampling algorithm can be used for time-domain noise reduction. The rendering process is the visibility buffer - > g-buffer mechanism to avoid ray tracing defects.

**Key word：**Ray tracing, Spatiotemporal Variance-Guided Filtering, denoiser, monte-carlo method

# 简介与意义/Introduction

## 项目意义和依据/Significance

近年来，路径追踪变为工业界常用的渲染算法。为了减少蒙特卡洛方法不可避免的噪声，刺激了更高级的滤波器的发展。游戏和电影近年来都从基于经验的模型转向了基于物理的渲染模型，但是为了得到更加准确的阴影与反射，开发者们还是更加愿意考虑使用光线追踪以及多次反射的全局光照。但是当前的光线追踪技术限制在200到300 Mrays/sec, 使在1920\*1080的分辨率和30Hz刷新率的情况下每个像素仅有几根光线。这个数字在不同性能的GPU/CPU下，以及在使用动态加速结构，在更大的场景下甚至更低。因此在可预见的未来，以及随着分辨率和刷新率不断提高的趋势下，这种方法是不可行的。

因此，通过研究重构滤波器，我们才有可能实现实时路径追踪。然而，在低采样率下进行重构是非常困难的。低采样率下的高方差使高频信号变得很模糊，并且，每个像素只有一个采样点分辨噪声的来源也是很困难的。

为了实现目标性能，本课题小组使用的滤波器使用前一帧的采样来区分细节以及消除噪声源。在1920\*1080的分辨率下，使用当前的GPU可以在10ms内进行渲染。我们所借鉴的方法具体有以下几点：

一种从每个像素一条采样路径的输入中进行实施重构的高效且时域稳定的算法。这种算法基于由空间域和时间域中的估计方差引导的滤波器组合。

使用基于GGX模型采用重要性采样分布模型随机选择一条光路。

渲染部分采用Visibility-buffer机制，避免渲染瑕疵。

## 本方法/系统框架/Article Structure

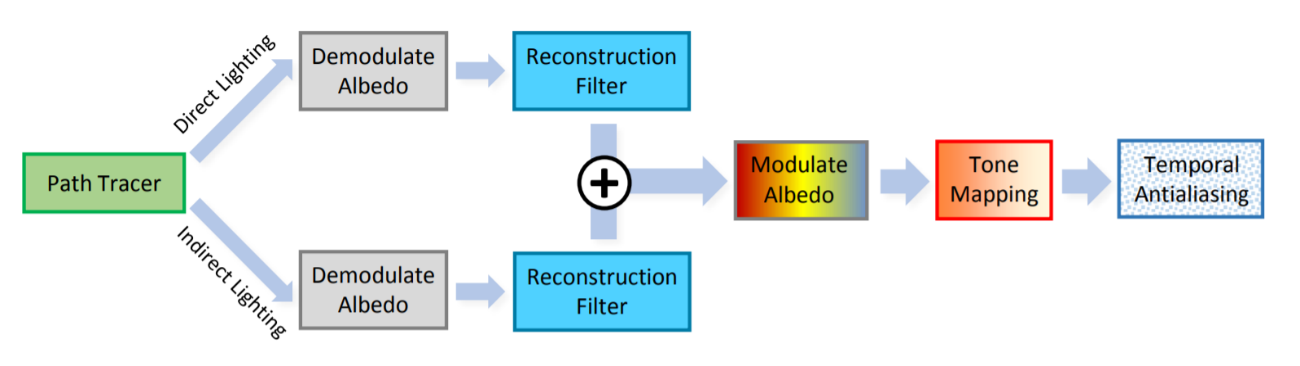


图 1：渲染流程图

图 1是本实验中用到的图形渲染流程图，包括了光栅化以及路径追踪的输入，以及如何通过分隔组件即，重构滤波器以及其他后处理步骤来隔离噪声源。

关于路径追踪。即利用了光线追踪以及蒙特卡罗方法。在本实验中我们利用了下一事件估计的标准路径追踪来生成1spp的样本。我们的路径追踪器使用了优化来更好地利用GPU资源，这包括利用光栅化器来更加高效地生成主光线。这提供了一个无噪声的包含用来指导重构滤波器的额外表面属性的G-buffer。一个低差异的Halton序列被用来采样光源以及分散光线。

我们的路径追踪器将直接光照和间接光照分别输出。这使得滤波器能够在两个分量重独立地考虑局部平滑度，并且允许低采样导致的边缘阴影能够更好的重构。

重构。我们先解调来自于采样颜色的可见表面的表面反照度。这避免了滤波器不得不防止对高频纹理细节的过度模糊。换句话说我们过滤非纹理组件，并在重建后重新应用纹理。除了消除防止滤波器过度模糊纹理细节的需要之外，这还增加了相邻样本的可能空间重用。在多层材料的情况下，我们添加每层反照率，并根据其采样概率进行加权。

我们的重构主要实施以下三个步骤：时间上积累1spp路径追踪的输入来增加有效采样率，并使用这些时域增强的颜色样本来估计局部照明方差以及利用这些方差估计来驱动一个分层多空小波滤波器。

后处理。重构结束后，实施类似于许多实时渲染器的后处理。滤波器结果通过色调映射算子来处理高动态范围。最后执行时间抗锯齿来增加时域稳定性以及我们的重构滤波器所保存的沿着几何边缘的滤波走样。

# 相关工作/Related Works

实现实时全局光照很多年来一直困扰着研究人员。当前的近似通常依赖于预先计算或缓存光传输计算[1], 在曲面上或在稀疏采样的体积中。为了增加缓存的照明， 其他技术允许包含特定的所需效果，例如环境光遮挡、 光泽屏幕空间反射和柔和阴影[2]. 虽然通常看似合理，但最终的照明效果却远非真实。为了在实时渲染中接近真实效果，本课题小组认为开发者们必须转向基于物理的蒙特卡洛光线追踪上来。

蒙特卡洛降噪。降噪滤波器通过组合多个每个像素的蒙特卡洛估计器来降低方差，以引入偏差为代价。他们希望在保留例如边缘和表面细节等清晰的图像特征的前提下实现平滑的输出。

基于回归的方法[3] 可以在高采样率的情况下产生好的结果，然而，这些滤波器在低分辨率下并不能可靠的工作由于他们对于异常值很敏感。

Munkberg [4]显示在纹理空间中操作允许使用更简单的滤波器，这些滤波器可以在时域中重复使用。然而，将这种方法扩展到用于实时渲染的大型资源并不容易。过滤也可以发生在路径空间中[5], 但是这使用了昂贵的KNN搜索技术，并且耦合了渲染和过滤算法。

非线性滤波器不缩小滤波器覆盖区，而是调整滤波器权重以保留显著特征。非线性蒙特卡洛降噪的开创性工作使用异常值去除[6], 平稳的能量再分配和各向异性扩散。一种边缘保持双边滤波器可应用与蒙特卡洛降噪。一种新的交叉（或联合）双边滤波器[7] 将每个像素替换为附近像素的加权平均值，使用考虑距离、颜色和 其他差异的高斯分布权重，通过边缘停止函数引导图像。 在边缘停止函数中考虑几何信息，提高了交叉双边滤波器 对输入噪声的鲁棒性。

时间滤波利用跨多帧的时间信息有助于解决空间滤波器的缺点，并提高低采样率下的时间稳定性。Delbracio等人 [8] 考虑跨三个帧的光线直方图以减少闪烁，Meyer和 Anderson[9] 在所有帧上计算PCA，丢弃不重要的基以提高 时间稳定性。齐默等人[10] 将路径空间分解用于运动估计，并对多个缓冲器应用去噪。但是这些方法需要 先计算的帧的输入集合，而不是如实时所需的那样仅在先 前帧上进行时间滤波。 交互式滤波器通常基于运动矢量将样本从一个帧重新投影到另一个帧[11] 这类似于视图插值[12], 其从预渲染的一组图像中重新投影样本以生成新的视点。

最近，通过广泛采用时间抗锯齿（TAA），重投影得到了普及。 [13]. TAA从临时摊销超级抽样中获得灵感[14], 但不是丢弃陈旧的样本，而是对它们进行调节，以 匹配其重新投影位置的颜色。Patney [15] 通过估计颜色的统计分布来改进这一点。

# 研究内容与方法/Contnts and Methods

## 路径追踪

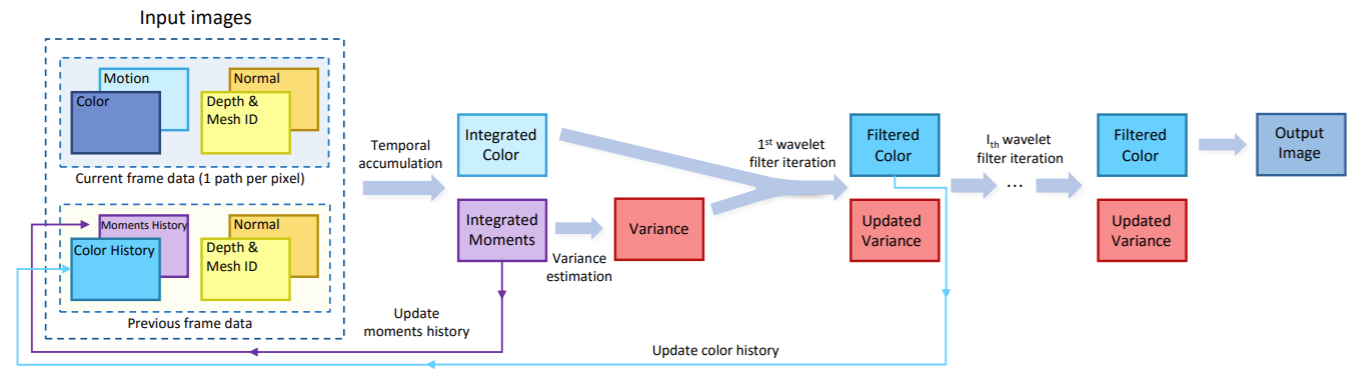
本项目使用标准的路径追踪来生成1 spp的颜色样本。路径追踪给之后的降噪阶段提供了一个visibility buffer，其中的光照、深度和法线属性可用于辅助降噪。光线散射方向采样利用了低差异Halton序列，产生的噪点少于一般的伪随机序列。路径追踪的输出中，直接光照和间接光照是分开的，允许对两种光照分别降噪以获得更好的阴影边缘。因为两者共享visibility buffer，所以分开计算也不会明显增大开销。

图 2：降噪流程图

## 降噪

路径追踪产生的颜色样本没有计算纹理表面反射率的影响，而是在降噪完成后再应用纹理，避免纹理的高频细节在降噪时被模糊，并且增加了空间降噪中相邻像素的利用率。降噪分为三个主要步骤：首先，在时间上累积路径追踪的数据以提高采样率，其次，利用样本估计局部亮度方差，最后，根据估计的方差执行atrous小波滤波。

### 时域滤波

我们的滤波器的输入是路径追踪颜色缓冲区、来自前一帧的visibility buffer和历史缓冲区，输出是降噪后的图像和以下帧的历史缓冲区。visibility buffer包含深度、对象和世界空间法线、网格 ID 和屏幕空间运动矢量，这些运动矢量是从光栅化过程中生成的。我们的历史缓冲区包括时间整合的颜色和颜色矩数据以及前一帧的深度、法线和网格 ID。

图 2展示了我们过滤管道中的主要步骤。

时域上累积得的颜色样本由第i帧（当前帧）的颜色样本*Ci*和历史缓冲区中对应的颜色样本*Ci-1*计算而来：

其中*α*是时域的衰减参数，根据经验取为0.2。

### 空域滤波

滤波应当避开噪声很少的区域。本项目利用时域累积的颜色样本来检测噪声。通过分析时间线上的不同样本，可以检测到某次采样的可信度。要注意，空域上的方差并不能有效代表噪声的程度，因为噪声会提高方差，但是方差并不一定来自噪声。

本项目使用每像素原始亮度的一阶矩*μ1i*和二阶矩*μ2i*来估计它的真实亮度。为了收集足够的样本来提高估计的可信度，重用了几何一致性检测。然后，用公式来计算时域上的方差。

相机移动、动画、视图边缘都会导致像素在时域上的连贯性，进而影响方差的估计。如果历史帧数太少（小于4帧），转而使用空域的方差。使用由深度和世界空间的法线驱动的双边滤波计算空域方差。

下一步是参照估计的方差来进行á-trous小波变换。本项目参照……的交叉双边滤波，滤波公式为：

其中 为像素和的权重函数，

### 边缘停止函数

考虑到实时要求，本项目选取了上式中三个边缘停止函数以最大化时域上的稳定性和鲁棒性，作为交换潜在地增加空域上的偏差。每个函数拒绝样本的能力由以下各参数单独控制，即。

深度。现实场景通常包含较大的几何比例变化，这使得全局边缘停止函数难以控制， 因此，本项目假设了一个表面深度的局部线性模型并且测量其与剪裁空间平面的偏差。本项目使用剪裁空间深度的屏幕空间偏导数来估计局部深度模型。这里给出了权重函数，定义为：

其中，时剪辑空间深度相对于屏幕空间坐标的梯度，是避免被0除的较小值。

法线。本项目对世界空间法线上的边缘停止函数采用余弦项：

其中为图像平面上的点处的输入法线。网格简化和反走样算法中的前期工作中使用类似的项来控制是否将两个表面合并在一起。

亮度。本项目的亮度边缘停止函数的一个关键点是它能通过基于局部标准偏差重新归一化亮度来自动适应所有尺度。但是在低样本数量下操作给方差和标准偏差的估计引入了不稳定性，这可能会造成伪影。为避免这些问题，本项目使用3×3高斯核心函数对方差图像进行与滤波，因此亮度边缘停止函数变为：

其中为高斯核心函数，为位置p处的亮度。由于亮度方差倾向于随着随后的滤波迭代而减小，因此的影响随着每次迭代而增长，从而防止过度模糊。

# 实验结果与分析/Experiment Results and Analysis （略）

# 特色与创新/ Distinctive or Innovation Points （略）

References:

1. James T Kajiya. 1986. The Rendering Equation. In Computer Graphics, Vol. 20. 143–150.
2. Tobias Ritschel, Carsten Dachsbacher, Thorsten Grosch, and Jan Kautz. 2012. The State of the Art in Interactive Global Illumination. Computer Graphics Forum 31, 1 (2012), 160–188.
3. Benedikt Bitterli, Fabrice Rousselle, Bochang Moon, Jose A Iglesias-Guiti ´ an, David ´ Adler, Kenny Mitchell, Wojciech Jarosz, and Jan Novak. 2016. Nonlinearly Weighted ´ First-order Regression for Denoising Monte Carlo Renderings. In Computer Graphics Forum, Vol. 35. 107–117.
4. Jacob Munkberg, Jon Hasselgren, Petrik Clarberg, Magnus Andersson, and Tomas Akenine-Moller. 2016. Texture Space Caching and Reconstruction for Ray Tracing. ¨ ACM Transactions on Graphics 35, 6 (2016), 249:1–249:13.
5. Nikolaus Binder and Alexander Keller. 2016. Effcient Stackless Hierarchy Traversal on GPUs with Backtracking in Constant Time. In High Performance Graphics. 41–50.
6. M. E. Lee and R. A. Redner. 1990. A Note on the Use of Nonlinear Filtering in Computer Graphics. IEEE Computer Graphics and Applications 10, 3 (1990), 23–29.
7. Elmar Eisemann and Fredo Durand. 2004. Flash Photography Enhancement via Intrinsic ´ Relighting. ACM Transactions on Graphics 23, 3 (2004), 673–678.
8. Mauricio Delbracio, Pablo Muse, Antoni Buades, Julien Chauvier, Nicholas Phelps, ´ and Jean-Michel Morel. 2014. Boosting Monte Carlo Rendering by Ray Histogram Fusion. ACM Transactions on Graphics 33, 1 (2014), 8:1–8:15.
9. Mark Meyer and John Anderson. 2006. Statistical Acceleration for Animated Global Illumination. ACM Transactions on Graphics 25, 3 (2006), 1075–1080.
10. Henning Zimmer, Fabrice Rousselle, Wenzel Jakob, Oliver Wang, David Adler, Wojciech Jarosz, Olga Sorkine-Hornung, and Alexander Sorkine-Hornung. 2015. Path-space Motion Estimation and Decomposition for Robust Animation Filtering. Computer Graphics Forum 34, 4 (2015), 131–142.
11. Diego Nehab, Pedro V. Sander, Jason Lawrence, Natalya Tatarchuk, and John R. Isidoro. 2007. Accelerating Real-time Shading with Reverse Reprojection Caching. In Graphics Hardware. 25–35.
12. Shenchang Eric Chen and Lance Williams. 1993. View Interpolation for Image Synthesis. In Proceedings of SIGGRAPH 93. 279–288.
13. Brian Karis. 2014. High-ality Temporal Supersampling. In SIGGRAPH Courses: Advances in Real-Time Rendering in Games.
14. Lei Yang, Diego Nehab, Pedro V Sander, Pitchaya Sihi-amorn, Jason Lawrence, and Hugues Hoppe. 2009. Amortized Supersampling. ACM Transactions on Graphics 28, 5 (2009), 135:1–135:12.
15. Anjul Patney, Marco Salvi, Joohwan Kim, Anton Kaplanyan, Chris Wyman, Nir Benty, David Luebke, and Aaron Lefohn. 2016. Towards Foveated Rendering for GazeTracked Virtual Reality. ACM Transactions on Graphics 35, 6 (2016), 179:1–179:12.

时间安排与分工统计表

|  |  |  |  |
| --- | --- | --- | --- |
| **组员信息**（含组长） | | | |
| 学生姓名 |  | 学 号 |  |
| 项目分工 |  | | |
| 学生姓名 |  | 学 号 |  |
| 项目分工 |  | | |
| 学生姓名 |  | 学 号 |  |
| 项目分工 |  | | |
| **时间安排/**  **Schedule** | （如选题、方案制定、试验研究、数据处理、研制开发、撰写总结报告等）(Such as topic selection, program formulation, experimental research, data processing, research and development, writing summary reports, etc.) | | |