

# 基于光线追踪的全局光照及降噪处理研究

李学阔<sup>1</sup>, 温佩贤<sup>2</sup>, 杨 林<sup>2</sup>, 方泽华<sup>2</sup>, 杜晓荣<sup>1</sup>

(1. 中山大学 物理学院, 广东 珠海 519000;

2. 金山网络科技有限公司, 广东 珠海 519000)

**摘 要:** 传统的路径追踪算法能够有效地追踪光线的运动轨迹, 从而在屏幕上呈现出真实的渲染效果, 但传统的光线追踪算法只能追踪镜面反射或规则的折射光, 追踪漫反射光消耗太大, 效率低下。基于蒙特卡洛概率方法的光线追踪能够从统计意义上很好地实现物体间漫反射的光照效果, 极大地解决了传统光线追踪的缺陷和效率问题, 然而只有当采样数为无限大时, 蒙特卡洛方法才是无误差的。为了提升渲染性能和质量, 尽可能缩小误差, 提出了一种根据颜色采样分布进行降噪处理的优化方法, 该方法可以减少采样数, 通过后期处理来去除噪声。通过在自研图形引擎中渲染虚拟场景对比实验发现, 根据颜色采样分布进行降噪处理的方法能够实现较好的渲染及降噪效果, 提高整个算法的性能以及画面表现。

**关键词:** 全局光照; 光线追踪; 蒙特卡洛; 采样数; 降噪

中图分类号: TP391

文献标识码: A

文章编号: 1673-629X(2022)09-0023-05

doi: 10.3969/j.issn.1673-629X.2022.09.004

## Research on Global Illumination and Denoising Based on Ray Tracing

LI Xue-kuo<sup>1</sup>, WEN Pei-xian<sup>2</sup>, YANG Lin<sup>2</sup>, FANG Ze-hua<sup>2</sup>, DU Xiao-rong<sup>1</sup>

(1. School of Physics, Sun Yat-Sen University, Zhuhai 519000, China;

2. Kingsoft Corporation, Zhuhai 519000, China)

**Abstract:** The traditional path tracing algorithm can effectively track the trajectory of light, thereby presenting the real rendering effect on the screen, but it can only track the specular reflection or regular refracted light, tracking diffuse light is inefficient. Ray tracing based on Monte Carlo probability method can achieve the lighting effect of diffuse reflection between objects in a statistical sense, which greatly solves the defects and efficiency problems of traditional ray tracing. However, Monte Carlo method is error free only when the number of samples is infinite. In order to improve the rendering performance and quality and reduce the error as much as possible, an optimization method for denoising according to the color sampling distribution is proposed. This method can reduce the number of samples and remove the noise through post-processing. Through the contrast experiment of rendering virtual scene in self-developed graphics engine, it is found that the method of denoising according to color sampling distribution can achieve better rendering and denoising effect, and improve the performance of the entire algorithm.

**Key words:** global illumination; ray tracing; Monte Carlo method; number of samples; denoising

## 0 引言

近年来,随着游戏、影视、虚拟现实、增强现实等技术的发展,对计算机图形质量和渲染速度的要求也越来越高,如何使用有限的计算机硬件资源快速地渲染出复杂精细的场景、逼真的色彩和光影一直是计算机图形学领域的研究热门问题。其中在全局光照模型的研究领域中,取得了不少成果。在早期的全局光照算法中,主要使用了文献[1]中提到的光线追踪算法,但该算法能表现镜面反射或规则的折射光,追踪漫反射光消耗大、效率低。在此之后,为了解决该算法的缺

点,研究人员不断进行改进,其中文献[2]提出的基于蒙特卡洛概率随机算法的光线追踪,能够从统计意义上很好地实现物体间漫反射的光照效果,然而只有当采样数为无限大时,蒙特卡洛方法才是无误差的,这将消耗大量的计算机硬件资源和时间。

随着研究的不断深入,越来越多的全局光照算法被提出,包括文献[3-4]提到的辐射度算法,该算法可以实时计算面片形状因子,但是对于面片数目较多的复杂场景,求解仍然非常耗时。文献[5-6]提到的光子映射算法可以有效地模拟焦散结果,作为一种有偏

收稿日期: 2021-09-17

修回日期: 2022-01-19

基金项目: 广东省重点领域研发计划资助项目(2019B010148001)

作者简介: 李学阔(1990-),男,硕士研究生,研究方向为计算机图形学和虚拟现实;杜晓荣,教授,研究方向为计算机图形学和虚拟现实。

方法,光子映射在有限的迭代次数或不合适的场景配置下,总存在偏差。文中提出了一种根据颜色采样分布进行降噪处理的优化方法,该方法可以减少采样数,通过颜色采样分布来有效去除噪声,提高算法的性能以及画面表现。

## 1 相关工作

### 1.1 传统的光线追踪

为了模拟真实的场景光照,传统的光线追踪算法被提出,它能够有效地追踪光线的运动轨迹,从而在屏幕上呈现出真实的渲染效果,但传统的光线追踪算法只能追踪镜面反射或规则的折射光,追踪漫反射光消耗太大,效率低下,对辉映、焦散等效果不能很好地模拟。传统光线追踪算法流程如下所述:

(1)从视点出发,经过投影屏幕上的每一个像素点向场景发射一条虚拟的光线,开始进入(2),若递归跟踪返回不相交信息,则该像素点的颜色渲染为背景色;若递归跟踪返回相交信息,该像素点的颜色即为递归跟踪光线得到的颜色。

(2)进入递归跟踪,若递归深度超过预设阈值或者不和任何场景物体相交,则停止递归,返回不相交信息,否则求光线与场景物体最近的交点,进入(3)。

(3)根据当前交点所在的物体表面材质信息,计算其入射光线,向其入射光线的反方向继续跟踪,若返回(2)继续递归仍然相交,则该交点的颜色根据该点物体表面的反射率对该点自身颜色和递归得到的颜色加权得到;若返回(2)继续递归不相交,则该交点的颜色为该点自身的颜色。

图 1 展示了传统光线追踪算法的效果。

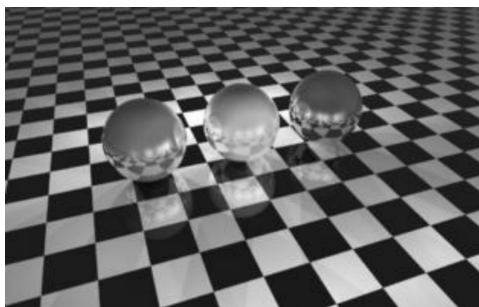


图 1 传统光线追踪渲染效果

### 1.2 基于蒙特卡洛概率算法的光线追踪

传统的光线追踪算法有一些突出的缺点,例如只能追踪表面属性单一情况下的光线,不考虑漫反射等。于是有研究人员通过模型的修正来解决这些问题,事实上一个物体表面的属性通常以混合的形式存在,例如 20% 概率的光线会进行镜面反射,30% 概率的光线会进行折射,50% 概率的光线会发生漫反射,通过蒙特卡洛采样算法,每次按照概率决定光线的反射属性,把

半球积分方程进行近似简化,能够用少量相对重要的采样来模拟积分。蒙特卡洛光线追踪已经是一个较为完备的渲染方案,它极大地解决了传统光线追踪的缺陷和效率问题,相对于普通光线追踪,蒙特卡洛光线追踪引入了更为复杂的漫反射模型,增加了需要跟踪的光线数量,但同时它使用的采样算法减少了需要跟踪的光线数量,其核心效率取决于其采样模型。

渲染方程及其在计算机图形学中的应用在 1986 年被 James Kajiya 提出<sup>[7]</sup>,蒙特卡洛光线追踪也同时在该文章中被提出,作为一种数值计算方法来计算渲染方程积分。大概十年后,Lafortune 对该方法提出了许多改进,包括双向光线追踪<sup>[8]</sup>。Eric Veach 和 Leonidas J. Guibas 提出的 Metropolis light transport<sup>[9]</sup>通过扰乱先前找到的光线路径来提高在复杂场景的渲染效率。Delbracio M 使用了一种 resampling<sup>[10]</sup>的方式提高光线追踪效果。Moon B 等使用了图像滤波<sup>[11]</sup>的方式,Bako S, Vogels T, Wong K M 等人使用了时域滤波<sup>[12-14]</sup>的方式,Schied C 和 Chaitanya C R 等人使用了深度学习<sup>[15-18]</sup>的方法去除噪声。

## 2 文中算法

### 2.1 整体流程

基于颜色采样分布降噪处理的蒙特卡洛光线追踪算法是将降低蒙特卡洛采样率后的噪声图像经过像素点采样,扩展到像素块,平均降噪,获得降噪后的图像,整体流程如图 2 所示。

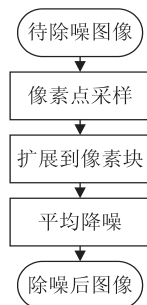


图 2 算法整体流程

### 2.2 基于颜色采样分布降噪处理的蒙特卡洛光线追踪

不同于传统图像处理中的去除噪声,在蒙特卡洛光线追踪的渲染场景中,拥有更多的可用信息,每个像素点的颜色是通过多条光线的路径追踪计算出来的,场景中的几何信息、光线路径、光线与物体的碰撞等信息可用被存储起来,为去除噪声提供有用信息。

正如 Veach<sup>[19]</sup>指出的那样,光线传输问题可以用空间路径来描述,全局光照可以通过计算光线传输的每条路径估量。在路径积分公式中,每个像素的颜色

$$u(x) = (u_r(x), u_g(x), u_b(x)) \quad (1)$$

可以由

$$u(x) = \int_{\Omega_x} f(p) d\mu(p) \quad (2)$$

得到,其中  $\Omega_x$  表示以像素  $x$  为起点的路径空间,  $p$  为任意长度的路径,  $d\mu(p)$  为路径空间的一种测量,函数  $f(p)$  描述了路径  $p$  的能量贡献。这个式子可以通过使用蒙特卡洛采样算法产生  $n_x$  条随机路径  $p_x^1, \dots, p_x^{n_x}$  来计算近似值。假设  $c_x^j$  表示由路径  $p_x^j$  得到的颜色,那么蒙特卡洛采样估计得到的  $u(x)$  可以表示为:

$$\tilde{u}(x) = \frac{1}{n_x} \sum_{j=1}^{n_x} c_x^j \quad (3)$$

那么蒙特卡洛误差  $n(x)$  可以表示为:

$$n(x) = \tilde{u}(x) - u(x) \quad (4)$$

因此,任务是尽可能缩小  $n(x)$ ,当采样数  $n_x$  减小时,  $n(x)$  将随之增大,只有  $n_x$  为无限大时,蒙特卡洛方法才是无误差的,而这是不可能实现的。为了尽可能缩小误差,可以减少采样数,并通过后期处理来去除噪声,从而提升渲染性能和质量。而为了使噪声去除算法更加准确,不引入新的误差,可以通过两个像素点  $x$  和  $y$  的颜色分布  $\{c_x^1, \dots, c_x^{n_x}\}$  和  $\{c_y^1, \dots, c_y^{n_y}\}$  来判断这两个像素点是否是近似的。若  $N$  个像素点拥有近似的颜色分布,那么这些像素点的所有采样的叠加可以被认为是遵循某个颜色分布的  $N$  倍的超集,通过平均这些相似像素点的颜色,可以降低图像的噪声。

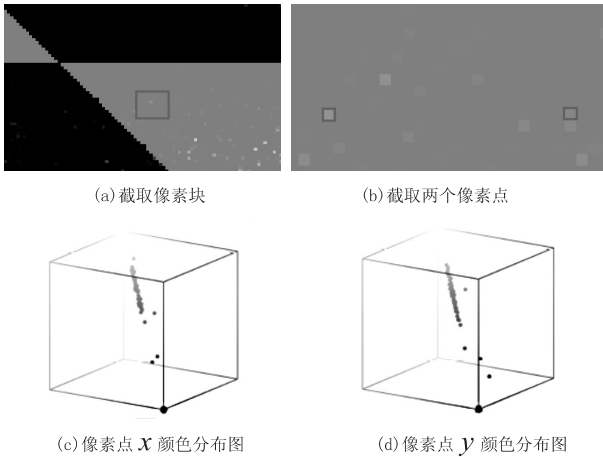


图3 像素颜色采样分布

图3截取了两个  $rgb$  值有较大差异的像素点,并画出了它们的颜色分布图,由于较强的光照,第一个像素点较亮,若仅仅对比像素点颜色,不可能将它们判断为相似的像素点,而通过结合了渲染采样信息的颜色分布图可以发现,这两个像素点其实有着共同的颜色分布特性,而这个结论不可能由通过对比它们的平均值得到。

为了得到更好的效果,对比像素块之间的相似度而不是通过对比单一像素的相似度,那么除噪将会通过平均相似像素块的颜色来实现,因此能够得到更为

柔和的除噪效果。给定在像素点  $x$  和像素点  $y$  的采样颜色  $c_x$  和  $c_y$ ,以及它们对应的颜色分布  $h(x) = (h_1(x), h_2(x), \dots, h_{n_b}(x))$  和  $h(y) = (h_1(y), h_2(y), \dots, h_{n_b}(y))$ ,可以基于卡方距离公式<sup>[20]</sup>做如下定义:

$$d_{\chi^2}(C_x, C_y) = \frac{1}{k(x, y)} \sum_{i=1}^{n_b} \frac{(\sqrt{\frac{n_y}{n_x}} h_i(x) - \sqrt{\frac{n_x}{n_y}} h_i(y))^2}{h_i(x) + h_i(y)} \quad (5)$$

其中,  $n_x = \sum_i h_i(x)$ ,  $n_y = \sum_i h_i(y)$ 。  $k(x, y)$  为  $h_i(x) + h_i(y)$  中的非空颜色分布,以上表示的为像素点间的相似度距离,下面将其扩展为边长为  $2\omega + 1$  的像素块。

$$d_{\chi^2}(P_x, P_y) = \sum_{|x-y| \leq \omega} d_{\chi^2}(C_{x+i}, C_{y+i}) \quad (6)$$

对于每个像素点  $x$ ,定义  $N_k(x)$  为满足  $d_{\chi^2}(P_x, P_y) \leq k$  的所有像素点  $y$  的集合,那么经过除噪后的颜色可以表示为:

$$\bar{u}(x) = \frac{1}{|N_k(x)|} \sum_{y \in N_k(x)} \tilde{u}(y) \quad (7)$$

上式只对像素块的中心点进行平均,若要对整个像素块进行平均,则计算出除噪后的像素块:

$$V_x = \frac{1}{|N_k(x)|} \sum_{y \in N_k(x)} \tilde{u}(P_y) \quad (8)$$

至此,已经计算出除噪后的像素块,再通过它来计算每个像素点,因为每个像素块包含  $2\omega + 1$  个像素,反过来每个像素点被  $2\omega + 1$  个像素块所包含,那么最终的除噪图像可以由下式得到:

$$\bar{u}(x) = \frac{1}{(2\omega + 1)^2} \sum_{|y-x| \leq \omega} V_y(y - x) \quad (9)$$

### 2.3 算法框架

基于前两小节的阐述,可以得到程序的大体框架如算法1所示。

ALGORITHM 1: Single-Scale Ray Histogram Fusion

Input: MC image  $\tilde{u}$ , corresponding histograms  $h$ , patch size  $w$ , search window size  $b$ , distance threshold  $k$

Output: Filtered image  $\bar{u}$

```

1:  $\bar{u} \leftarrow 0$ 
2:  $n \leftarrow 0$  // auxiliary counter at each pixel in the image
3: for every pixel  $i$  do
4:    $P_i \leftarrow$  patch centered at pixel  $i$ 
5:    $W_i \leftarrow$  search window with size  $b$  for pixel  $i$ 
6:    $c \leftarrow 0$  and  $V \leftarrow 0$ 
7:   for every  $j \in W_i$  do
8:      $Q_j \leftarrow$  patch centered at pixel  $j$ 
9:      $d \leftarrow \text{ChiSquareDistance}(h(P_i), h(Q_j))$ 
10:    if  $d < k$  then
11:       $V \leftarrow V + \tilde{u}(Q_j)$ 
12:       $c \leftarrow c + 1$ 

```



```

13:   end if
14: end for
15:  $V \leftarrow V / c$ 
16:  $n(P_i) \leftarrow n(P_i) + 1 // +1$  for each pixel in  $P_i$ 
17:  $\bar{u}(P_i) \leftarrow \bar{u}(P_i) + (V - \bar{u}(P_i)) ./ n(P_i)$ 
18: end for

```

Notation convention:  $\bar{u}(P_i)$  is the evaluation of  $u$  on each pixel in patch  $P_i$  (the same applies for  $\bar{v}, n, h$ ). The operator  $./$  (line 17) represents element-wise division.

### 3 实验及运行结果

使用 C++ 语言进行程序编写,实验运行在以下环境:CPU Intel i7 主频 2.40 GHz,内存 8 G。基于蒙特卡洛的光线追踪算法的采样率决定了画面质量,当物体表面属于漫反射属性时,基于蒙特卡洛的光线追踪算法向随机方向发射一定数量的光线进行采样,光线的数量决定着画面的真实度,发射的光线数量越多,效率越低,但画面越真实,而发射的光线数量越少,画面失真越严重,噪声越多。因此在实际应用中,采样率是一个需要调整平衡的重要参数,通过调整采样率来平衡实际渲染效果的性能和质量。

峰值信噪比 (PSNR) 是一种评价图像的有效标准,它的公式为:

$$\text{PSNR} = 10 \log_{10} \frac{(2^n - 1)^2}{\text{MSE}} \quad (10)$$

其中  $n$  表示每个采样值的比特数, MSE 指的是 Mean Square Error。PSNR 的单位为 dB。图 4 根据采样数与渲染耗时和 PSNR 的关系数据绘制,阐述了随着每个像素点采样数的增加,渲染耗时呈指数级增加,而画面真实度逐渐提高。

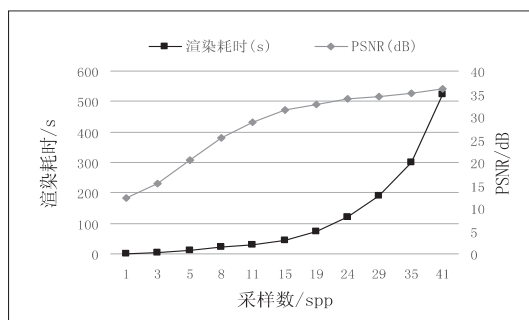
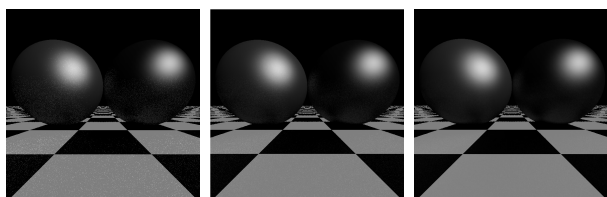


图 4 采样数 (spp: sample per pixel) 与渲染耗时和 PSNR 的关系

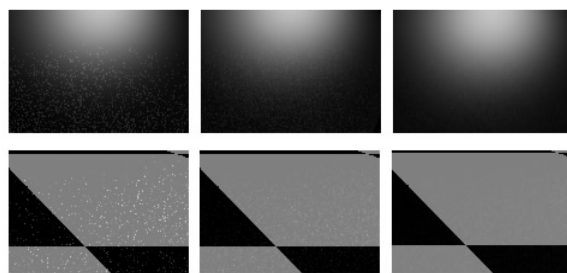


(a) 一条光线采样 (b) 五条光线采样 (c) 二十条光线采样

图 5 不同采样率下的蒙特卡洛光线追踪

图 5 中,球表面为 100% 漫反射属性,图 5(a) 只随机发射一条光线,图 5(b) 随机发射五条光线,图 5(c) 随机发射二十条光线。可以发现,蒙特卡洛概率采样方法的采样率越高,渲染出来的图像越接近真实场景,噪声越低。

图 6 对比了固定 8 spp 下未除噪、非局部均值除噪、基于颜色分布除噪三种渲染效果。



未除噪 (22.4 dB) 非局部均值除噪 (43.4 dB) 基于颜色分布除噪 (45 dB)

图 6 未除噪、非局部均值除噪、基于颜色分布除噪的渲染效果对比

图 7 根据未除噪、基于颜色分布除噪、非局部均值除噪三种情况下的 PSNR 进行对比数据绘制,表明基于颜色分布的除噪方法能够比非局部均值除噪算法更好地得到真实的渲染图像。

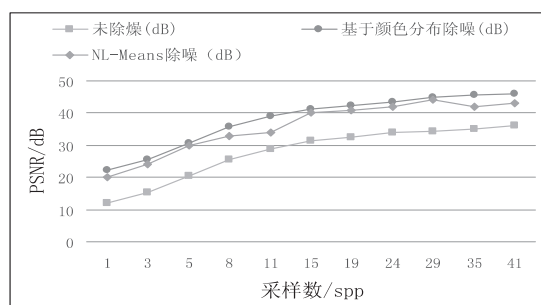


图 7 未除噪、基于颜色分布除噪、非局部均值除噪的 PSNR 对比

在基于颜色分布的除噪方法中,随着式子  $d_x(P_x, P_y) \leq k$  中  $k$  阈值的增大,除噪效果更为明显。图 8 分别列出  $k = 0.1, k = 0.3, k = 0.7$  以及  $k = 1.0$  的除噪效果。

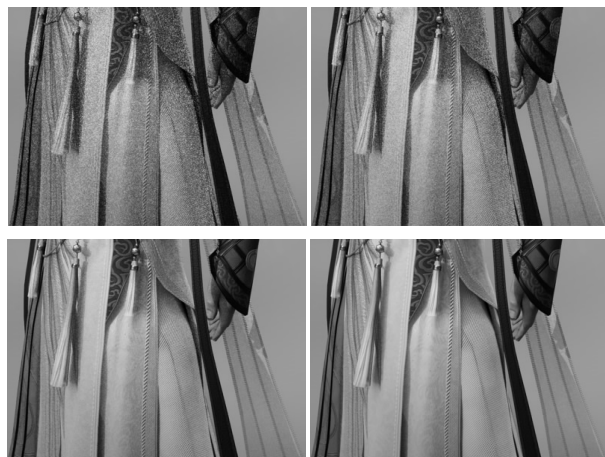


图 8 基于颜色分布的除噪效果

果(固定 5 spp 采样率)。但随着  $k$  的增大,过多的像素会被融合,因此得到的渲染图像会趋于模糊,虽然像素颜色值的方差会减小,但  $k$  过大会引入新的误差,导致 MSE 值反而变大。式(10)中的 MSE 可以化简为除噪后像素颜色值的方差加上除噪后的像素颜色偏差,如下式所示:

$$\text{MSE}(\hat{\theta}) = E[(\hat{\theta} - E(\hat{\theta}))^2] + (E(\hat{\theta}) - \theta)^2 = \text{Var}(\hat{\theta}) + \text{Bias}(\hat{\theta}, \theta)^2 \quad (11)$$

图9表明  $k$  的增大能够有效减小 MSE 值,从而增大 PSNR,但  $k$  过大会导致  $\text{Bias}(\hat{\theta}, \theta)$  的增大,虽然  $\text{Var}(\hat{\theta})$  减小,但其减小的程度小于  $\text{Bias}(\hat{\theta}, \theta)$  增大的程度。另一方面,  $\text{Var}(\hat{\theta})$  的减小意味着像素间的相似度更高,因此图像会趋于模糊。

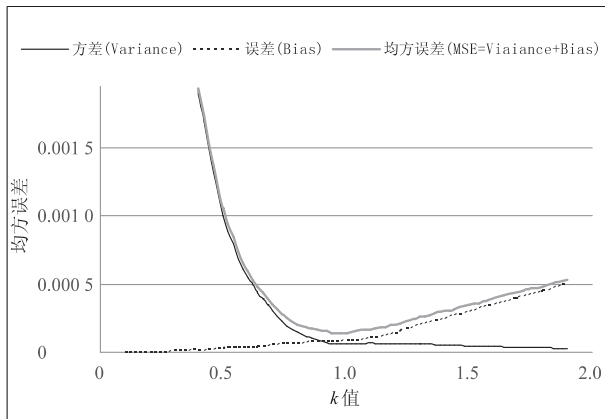


图9 MSE 与  $k$  的关系曲线

#### 4 结束语

文中介绍了使用光线追踪方法实现的全局光照算法,阐述了传统的光线追踪算法以及基于蒙特卡洛概率采样算法的光线追踪的优缺点,提出了一种基于颜色采样分布降噪处理的蒙特卡洛光线追踪的方法。在实验中通过设置不同的采样率以及物体表面反射属性占比,对比了不同情况下的渲染效果以及渲染效率。实验结果表明,使用基于颜色采样分布降噪处理的蒙特卡洛光线追踪的方法降噪效果更为明显,可以在更小的采样数下有效提升画面表现,从而提高整个算法的性能。

#### 参考文献:

- [1] WHITTET T. An improved illumination model for shaded display[J]. Communications of the ACM, 1980, 23(6): 343-349.
- [2] COOK R L. Stochastic sampling in computer graphics[J]. ACM Transactions on Graphics, 1986, 5(1): 51-72.
- [3] COHEN M F, GREENBERG D P. The hemi-cube: a radiosity solution for complex environments [J]. ACM SIGGRAPH Computer Graphics, 1985, 19(3): 31-40.

- [4] 袁璐. 基于立即辐射度的实时全局光照算法[J]. 现代计算机, 2018(2): 63-66.
- [5] JENSEN H W. Realistic image synthesis using photon mapping [M]. [s. l.]: AK Peters/CRC Press, 2001.
- [6] 康春萌. 基于光子映射的全局光照技术研究[D]. 济南: 山东大学, 2016.
- [7] KAJIYA J T. The rendering equation [J]. ACM SIGGRAPH Computer Graphics, 1986, 20(4): 143-150.
- [8] LAFORTUNE E P, WILLEMS Y D. Rendering participating media with bidirectional path tracing [C]//Eurographics workshop on rendering techniques. Vienna: Springer, 1996: 91-100.
- [9] VEACH E, GUIBAS L J. Metropolis light transport [C]//Proceedings of the 24th annual conference on computer graphics and interactive techniques. New York: ACM, 1997: 65-76.
- [10] DELBRACIO M, MUSÉ P, BUADES A, et al. Boosting Monte Carlo rendering by ray histogram fusion [J]. ACM Transactions on Graphics, 2014, 33(1): 1-15.
- [11] MOON B, MCDONAGH S, MITCHELL K, et al. Adaptive polynomial rendering [J]. ACM Transactions on Graphics, 2016, 35(4): 1-10.
- [12] SCHIED C, KAPLANYAN A, WYMAN C, et al. Spatiotemporal variance-guided filtering: real-time reconstruction for path-traced global illumination [M]//Proceedings of high performance graphics. New York: ACM, 2017: 1-12.
- [13] CHAITANYA C R A, KAPLANYAN A S, SCHIED C, et al. Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder [J]. ACM Transactions on Graphics, 2017, 36(4): 1-12.
- [14] SCHIED C, PETERS C, DACHSBACHER C. Gradient estimation for real-time adaptive temporal filtering [J]. Proceedings of the ACM on Computer Graphics and Interactive Techniques, 2018, 1(2): 1-16.
- [15] BAKO S, VOGELS T, MCWILLIAMS B, et al. Kernel-predicting convolutional networks for denoising Monte Carlo renderings [J]. ACM Trans. Graph., 2017, 36(4): 97:1-97:14.
- [16] VOGELS T, ROUSSELLE F, MCWILLIAMS B, et al. Denoising with kernel prediction and asymmetric loss functions [J]. ACM Transactions on Graphics, 2018, 37(4): 1-15.
- [17] WONG K M, WONG T T. Robust deep residual denoising for Monte Carlo rendering [M]//SIGGRAPH Asia 2018 technical briefs. New York: ACM, 2018: 1-4.
- [18] WONG K M, WONG T T. Deep residual learning for denoising Monte Carlo renderings [J]. Computational Visual Media, 2019, 5(3): 239-255.
- [19] VEACH E. Robust Monte Carlo methods for light transport simulation [D]. Stanford: Stanford University, 1998.
- [20] BUADES A, COLL B, MOREL J M. A review of image denoising algorithms, with a new one [J]. Multiscale Modeling & Simulation, 2005, 4(2): 490-530.