

基于自适应光子发射的渐进式光子映射

李睿^{1,2}, 陈彦云¹, 刘学慧¹

(1. 中国科学院软件研究所 计算机科学国家重点实验室, 北京 100190;

2. 中国科学院研究生院, 北京 100049)

摘要: 通过对渐进式光子映射算法进行扩展, 提出了一种基于自适应光子发射的渐进式光子映射算法。渐进式光子映射是一个多遍的全局光照算法, 通过不断发射光子并渐进更新场景各点的光能估计能使其最终能收敛到无偏差的结果。由于渐进式光子映射完全使用密度估计来计算各点的光能, 因此其收敛速度受光子分布影响较大。利用渐进式光子映射算法中固有的场景统计信息以及其多遍的特点, 设计了一个自适应的光子发射策略, 使得发射的光子能更多的分布在最终绘制有效的区域, 提高了原算法的绘制效率。

关键词: 渐进式光子映射; 自适应光子发射; 视点重要性; 重要性采样; 全局光照

中图分类号: TP391 **文献标识号:** A **文章编号:** 1000-7024 (2012) 01-0219-05

Adaptive photon tracing based progressive photon mapping

LI Rui^{1,2}, CHEN Yan-yun¹, LIU Xue-hui¹

(1. State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China;

2. Graduate University, Chinese Academy of Sciences, Beijing 100049, China)

Abstract: A simple extension of progressive photon mapping for tracing photon is presented adaptively, which make the algorithm converge to the right result more quickly. Progressive photon mapping is a multi-pass global illumination algorithm. The algorithm can compute the correct radiance value at a point in the limit by tracing photons pass by pass which update the point's radiance progressively. Progressive photon mapping totally uses density estimation to compute radiance, so its convergence rate is affected by photon distribution. According to progressive photon mapping's inherent statistical information and its multi-pass feature, an adaptive photon tracing strategy is designed, which make a better photon distribution each pass and improve the rendering efficiency of the original algorithm.

Key words: progressive photon mapping; adaptive photon tracing; visual importance; important sampling; global illumination

0 引言

全局光照技术是图形学研究领域的一个重要方向, 而如何能高效准确的对场景, 尤其是复杂光照环境下的场景进行绘制一直是全局光照技术研究的重点和难点。在 1986 年求解全局光照的数学模型: 光能传播方程被提出后, 多种基于蒙特卡罗积分的光线跟踪算法被提出^[1-2], 并得到了广泛的应用。随着各种基于光线跟踪算法的工具包, 绘制引擎的发布, 以及图形处理器等硬件的发展, 人们开始考虑将这些算法应用到图形处理器上, 以加快算法的运行速度, 在对此进行了大量的工作^[3-5]。现有的比较成熟的系统

有 Manta^[6], Razor^[7], Arauna^[8] 以及 Optix^[9] 等。

在将考虑对传统算法加速的同时, 对原有算法的改进, 以及提出新的算法的工作也一直在持续进行着。Cline^[10] 等人以 MLT 算法为基础, 提出了 ERPT 算法, 该算法可以看成是 MLT 算法在像素上分层的一个版本, 具有更易实现的特点, 而 Lai^[11] 等人针对 ERPT 需要使用图像滤波而带来误差的缺点, 使用新的采样方法对其进行改进, 提出了 PMC-ER 算法。

光子映射算法作为全局光照领域的主流算法, 以其高效率, 能处理多种光照效果等特点, 一直受到广泛的关注。为了提高传统光子映射算法的效率, Havran^[12] 提出了反向

收稿日期: 2011-04-06; 修订日期: 2011-06-05

基金项目: 国家自然科学基金项目 (60573155)

作者简介: 李睿 (1986-), 男, 湖北襄阳人, 硕士研究生, 研究方向为基于全局光照真实感绘制; 陈彦云 (1971-), 男, 广东潮州人, 博士, 研究员, 研究方向为计算机图形学真实感绘制技术; 刘学慧 (1968-), 女, 湖南长沙人, 博士, 副研究员, CCF 高级会员, 研究方向为计算机图形学、虚拟现实。E-mail: lirui@ios.ac.cn

光子映射的概念,加速了原有算法在最终收集阶段 (final gathering) 的计算速度。Fradin^[13]提出了一个优化策略,用于优化处理当场景很大时,只有很小一部分的光子图会用于最终绘制的问题。光子映射算法的一个主要问题是其使用光子进行光能估计的过程引入了偏差。理论上,要完全消除偏差我们需要存储无穷的光子,针对这个问题,Toshiya Hachisuka^[14-15]提出了渐进式光子映射的算法。它使用一个多遍的绘制流程,通过不断的向场景中发射光子达到不断减小偏差的目的。

由于渐进式光子映射完全使用密度估计 (density estimation) 的方式进行光能计算,其收敛速度受到光子分布较大的影响,本文通过利用渐进式光子映射记录的统计信息,指导每遍的光子发射策略,通过自适应的发射光子,得到更好的光子分布,从而获得更快的收敛速度,提高了渐进式光子映射算法的实用性。

1 相关工作

为了在光子发射时考虑视点的因素,人们引入了视点重要性的概念。视点重要性可以决定场景中哪些区域对于视点是很重要的。获取视点重要性基本的做法是首先从视点发射类似光子的 importon,构造 importon 图将其存储起来,然后在发射光子时,利用存储的信息指导光子的发射和存储以求达到更好的光子分布。

同样为改善光子图中光子分布状况,Fan^[16]利用 metropolis sampling 技术构建从视点到光源的路径,然后这些路径中选择相应的顶点作为光子的存储点,这样保证了光子分布与视点相关。但是由于其构建光子图,需要使用完整的 MLT 算法获取路径,在实现和效率上会很慢。

渐进式光子映射完全使用密度估计来进行光能估计,因此其收敛效率与每遍参与渐进更新的光子数目直接相关。我们希望每遍发射的光子中有尽可能多的光子参与到渐进式密度估计中。受到视点重要性的启发,我们发现渐进式光子映射中的 hitpoints 以及其之后的不断更新的统计信息可以很好的指导之后的光子发射工作。

2 基于自适应光子发射的渐进式光子映射

2.1 渐进式光子映射概述

渐进式光子映射是一个多遍算法,它通过渐进更新光子的统计信息来完成对绘制方程的求解。从图 1 中可以看出它由一个 ray tracing pass 和多个 photon tracing pass 组成。与传统的光子映射算法流程正好相反,最初的 ray tracing pass 从视点发射光线与场景物体求交并存储光线跟踪路径中所有的非镜面交点 (hitpoint),然后在接下来每遍发射光子时,从光源处发射光子并更新相应 hitpoints 的统计数据。Hitpoint 存储的数据包括该点的位置 x ,该点相对于其所属像素的权重值 throughput,该点累计收集的

能 τ (为光子 flux 与 BRDF 的乘积),该点的 BRDF 函数索引,该点进行密度估计的半径 $R(x)$,以及该点累计收集到的光子数 $N(x)$ 。利用以上数据,在第 i 遍发射光子后,可以使用下式进行 radiance 估计

$$L(\vec{x}, \vec{\omega}) \approx \frac{\tau_i(\vec{x}, \vec{\omega})}{N_e(i) \pi R_i(\vec{x})^2} \quad (1)$$

式中: $N_e(i)$ 用于光子能量的归一化,表示到第 i 遍为止,总共发射的光子数量。由于每个 hitpoint 都会和一个像素对应,我们把所有 hitpoints 求得的 radiance 加权相加即可得到最终绘制的图像。

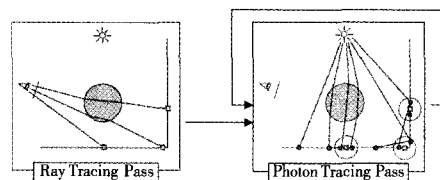


图 1 渐进式光子映射流程^[14]

从式 (1) 中可以看到,渐进式光子映射完全使用密度估计来计算光能 (radiance),而密度估计在参与估计的光子越多且估计半径越小时其引入的偏差越小。因此如何通过多遍发射光子使每个 hitpoint 进行光能估计的光子数增多以及与此同时估计半径减小就是算法的关键。

渐进式光子映射的用于减小偏差的渐进更新基于两个假设:①在密度估计半径内光子的密度以及光能是均匀的;②每遍发射光子后,每个 hitpoint 累计收集到的光子都要增加。如图 2 所示,基于以上两个假设,为了达到密度估计半径缩小而参与密度估计光子增加的目的,在每遍发射光子后,使用下式对每个 hitpoint 记录的统计数据更新

$$N_{i+1}(\vec{x}) = N_i(\vec{x}) + \alpha M_i(\vec{x}) \quad (2)$$

$$R_{i+1}(\vec{x}) = R_i(\vec{x}) \sqrt{\frac{N_i(\vec{x}) + \alpha M_i(\vec{x})}{N_i(\vec{x}) + M_i(\vec{x})}} \quad (3)$$

$$\phi_i(\vec{x}, \vec{\omega}) = \sum_{p=1}^{M_i(\vec{x})} f_r(\vec{x}, \vec{\omega}, \vec{\omega}_p) \phi_p(\vec{x}_p, \vec{\omega}_p) \quad (4)$$

$$\tau_{i+1}(\vec{x}, \vec{\omega}) = (\tau_i(\vec{x}, \vec{\omega}) + \phi(\vec{x}, \vec{\omega})) \frac{R_{i+1}(\vec{x})^2}{R_i(\vec{x})^2} \quad (5)$$

式中: $M_i(x)$ ——第 i 遍发射光子时,点 x 处收集到的光子数, α ——在 $(0, 1)$ 区间内用户设定参数。表示在一遍中收集到的光子数有多少会累积到该点累积收集到得光子数中。

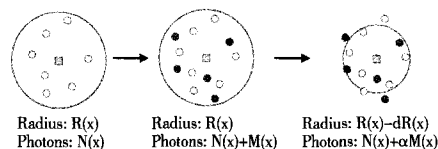


图 2 hitpoint 渐进更新^[14]

利用上述公式进行更新后, 可以在保证每遍绘制后, 每个 hitpoint 点累计收集到光子数都会增加, 并且该点进行密度估计的半径会减小。由于这个更新过程满足一致性要求, 所以每遍后图像的偏差会减少, 并且如果算法一直运行下去, 最终的光能估计会收敛到正确的结果

$$L(\vec{x}, \vec{\omega}) = \lim_{i \rightarrow \infty} \frac{\tau_i(\vec{x}, \vec{\omega})}{N_e(i) \pi R_i(\vec{x})^2} \quad (6)$$

渐进式光子映射的算法流程如下:

- (1) 从视点发射光线得到 hitpoints, 并存储起来。
- (2) 从光源发射光子, 并在光子路径中更新相应 hitpoints 的统计信息。
- (3) 对所有 hitpoints 利用光子发射时收集到得数据根据公式进行渐进更新。
- (4) 使用更新后的数据对每个 hitpoints 使用密度估计计算光能, 并根据 hitpoints 与像素的对应关系, 求得每个像素接受到的光能, 完成绘制。
- (5) 若达到预先设定的绘制遍数, 终止绘制; 否则回到第(2)步继续。

2.2 自适应光子发射

由于渐进式光子映射直接使用密度估计来计算光能, 其绘制效果受到光子分布的影响较大, 根据现有的实验数据, 在大多数情况下要得到一个无视觉噪声的结果图像, 需要发射几十亿量级的光子。图3是一个场景示意图, 其中虚线内的范围表示相机的可视区域。可以看出, 如果我们认定光源发射光子的红色路径在弹射后向左, 则其不会进入视点区域, 也就不会对最终绘制结果造成影响, 这造成了计算量的浪费; 同时图中B点和C点虽然都处于相机视野内, 但是由于B点离光源较近, 其收集的光子要多于C点, 这样可能造成B点收集的光子已经足够多却仍然向其大量的发射光子, 同样造成了浪费。因此如果我们可以更智能地向场景中发射光子, 就可以很大的提高绘制效率, 加快算法的收敛速度。

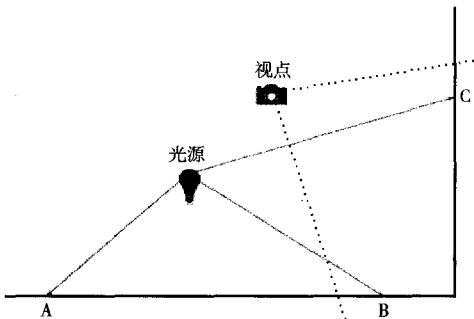


图3 场景

首先回顾一下光子发射的理论, 如果由 $L_e(x, \omega)$ 表示光源在点 \vec{x} 向 $\vec{\omega}$ 方向发射的光能, 则可知光源发射的总

能量为

$$\phi_L = \iint_{A\Omega_x} L_e(\vec{x}, \vec{\omega}) d\vec{\omega} d\vec{x} \quad (7)$$

光子发射需要保证能量守恒, 也即光源发射的所有光子的光能的总和应该与上式相等, 我们可以把每个光子看成是用蒙特卡洛采样方法求解上面积分式的一个采样的点, 由此可得每个光子的能量为

$$\phi_p = \frac{1}{N} \frac{L_e(\vec{x}, \vec{\omega})}{p(\vec{x}, \vec{\omega})} \quad (8)$$

式中: N ——光源总共发射的光子数。在传统的光子映射算法中, 用于采样的概率密度函数一般正比于 L_e , 这样可以使得所有发射出的光子具有相同的能量, 然而上式中用于采样的概率密度函数理论上可以任意指定, 但为了最大限度的减少绘制的方差 (variance) 和偏差, 由重要性采样的思想可知, 需要一个绘制效率相关的权重来构造发射光子使用的概率密度函数, 这样可以最大限度的提升绘制效率。

我们的策略是对每个光子路径都设置一个评价价值, 评价价值越高表示这条路径越渐重要, 然后我们设定进行光子发射采样的概率密度函数正比于其路径的评价价值, 就能达到向重要路径多发射光子的目的。渐进式光子映射在发射光子时, 每个光子在其路径中多需要更新相应的 hitpoints, 而 hitpoints 本身记录的信息如每个点的位置, 权重以及其累计收集到的光子数直接与最终绘制的结果相关, 具有视点重要性的特征, 因此可以为我们提供相应的权重信息, 作为评判光子路径好坏的参数。

考虑到拥有相同初始点和相同初始方向的光子, 具有较为相似的弹射路径, 为了能有指导性的发射光子, 我们将光源根据位置和方向分段。记每一段为 B_i , 则我们进行光子发射的概率公式可转化为

$$p(\vec{x}, \vec{\omega}) = p(B_i) * p(\vec{x}, \vec{\omega} | B_i) \quad (9)$$

用于指导光子发射就是需要求出适应场景的 $p(B_i)$ 。

初始的 $p(B_i)_0$ 在光源分层时求得, 在接下来的每遍发射光子时, 由于 hitpoints 记录了之前光子分布的信息, 利用这些信息对其进行调整, 从而在下一遍光子发射时达到更好的发射效果。在一个光子路径中, 可以在更新 hitpoints 信息的同时返回他们的权重信息和累计收集的光子数

$$photon_cnt_{path} = \sum_{hp \in path} photon_cnt_{hp} \quad (10)$$

$$weight_{path} = \sum_{hp \in path} throughput_{hp} * BRDF_{(hp, path)} \quad (11)$$

式中: $photon_cnt_{hp}$ ——该 hitpoint 累计收集的光子数 $N(x)$, $throughput_{hp}$ ——该 hitpoint 记录的权重值, $BRDF_{(hp, path)}$ ——该 hitpoint 存储的 BRDF 函数索引通过光子和 hitpoint 的各自入射和出射方向求得的 BRDF 值。

将属于同一个区块的路径信息累加, 就可以得到各个区块 B_i 的权重信息。为了防止因为区块发射的光子数目多而导致的权重值增大, 从而造成权重值大的区块下一遍发射更多光子, 权重值继续增大的累积现象, 使用区块发射

的光子数目对上述权重进行归一化。通过对属于同一区块的路径收集的信息进行累加并使用路径数进行归一化,我们可以得到每个区块平均权重和平均收集到的光子数

$$weight_{B_s} = (\sum_{path \in B_s} weight_{path}) / path_cnt_{B_s} \quad (12)$$

$$photon_cnt_{B_s} = (\sum_{path \in B_s} photon_cnt_{path}) / path_cnt_{B_s} \quad (13)$$

式中: $path_cnt_{B_s}$ ——该区块在该次发射中, 总共发射的光子数目。

每个区块的 $weight$ 表示了该区块发射光子的路径具有的平均视点重要性, 目的就是希望向具有视点重要性的区域多发光子, 因此该值可以作为确定每个区块自适应采样概率密度函数的基础。

由式(1)知道一个 hitpoint 其累计收集到光子数目越多, 其进行密度估计产生的偏差就越小。然而很多情况下每个 hitpoint 累积收集的光子数差别很大。例如离光源较近的 hitpoint 通常比离光源较远的 hitpoint 能收集到更多的光子, 这样在一些点可能造成收集的光子过量的浪费情况(见图7), 因此我们期望每个 hitpoint 累积收集的光子数不要相差太大。假定每个区块发射的光子所能影响的 hitpoints 的累计收集的光子数和该区块发射光子数成正比, 而我们指导每个区块发射的光子数与该区块的 $p(B_s)$ 成正比, 因此我们可以用下式对区块的权重进行调制

$$photon_cnt_{avg} = (\sum_{B_s} photon_cnt_{B_s}) / M \quad (14)$$

$$p(B_s)'_{new} = p(B_s)_o * (weight_{B_s} * \frac{photon_cnt_{avg}}{photon_cnt_{B_s}})^\beta \quad (15)$$

式中: M ——weight 不为 0 的区块数, β ——用户指定的参数, 表示初始概率密度函数受自适应因子的调制大小, 若为 0 则表示不进行自适应调制, 一般取 1 即可。

调整之后, 需要对新的权重归一化

$$p(B_s)_{i+1} = p(B_s)'_{i+1} / \sum_{B_k} p(B_k)'_{i+1} \quad (16)$$

这样就有了新 $p(B_s)$ 用于指导下一遍的光子发射了。

由上面的讨论总结改进后的算法流程如下:

- (1) 从视点发射光线得到 hitpoints, 并存储起来。
- (2) 根据光源分块的权重指导发射光子, 在路径中更新相应 hitpoints 的统计信息的同时根据 hitpoints 的信息返回该路径的权重信息, 并对该路径所属相应区块进行权重累加。
- (3) 对所有 hitpoints 利用光子发射时收集到数据根据公式进行渐进更新, 同时对各段概率密度函数进行更新。
- (4) 生成绘制图像(同渐进式光子映射)。
- (5) 若达到预先设定的绘制遍数, 终止绘制; 否则回到第(2)步继续。

因为调整策略完全基于 Russian roulette, 因此没有引入能量偏差, 并且由于附加计算量很小, 因此基本不会加重算法的负担。

3 实验结果及分析

使用 2.1 节的算法流程实现了渐进式光子映射, 在此基础上, 对光源分层实现了自适应光子发射(为了不失一般性在实验中我们使用分层的点光源)。在从视点发射光线后, 将得到的 hitpoints 存入一个 kd 树中, 之后根据分层光源的每块权重 $P(B_s)$ 决定每块的发射光子数, 每块分别采样发射光子, 在光子弹射路径的每一个顶点处, 使用当前所有 hitpoints 中的最大半径对 hitpoints 的 kd 树进行查找, 然后得到该光子可能影响的 hitpoints 集合, 然后通过再次判断该光子与集合中每个 hitpoint 的距离决定是否用该光子更新 hitpoint 的统计数据, 并同时根据 hitpoint 已有的信息记录路径权重, 在路径结束后返回权重更新光源的分块信息。

实验所用硬件平台为 Intel Core Quad CPU Q9400, 内存 4G。实验程序使用单核进行计算。实验图像分辨率分别为 600X450, 800X600 和 800X600。

图4是一个光源在室外, 然后观察对象为室内的场景。第一排为没有使用自适应光子发射的结果而第二排为使用了自适应光子发射的结果。其中二者每遍都发射 50 万光子, 每排 3 张图片分别是各自在 1, 20, 100 遍绘制后的结果。我们可以看到使用了自适应光子映射后, 每遍有更多的光子被发射进了屋内, 使得收敛速度加快了。因为该场景的状况使得不进入屋内的光子不会和任何物体相交, 从而计算量很小。因此不使用自适应光子发射时每遍所用时间比使用时要少, 但我们从图5可以看到即便在不使用自适应光子映射的情况下将迭代次数增加到 400 遍, 使其与使用自适应迭代 100 遍时具有相同的绘制时间, 后者的绘制效果仍明显高于前者, 证明了算法的实用性。

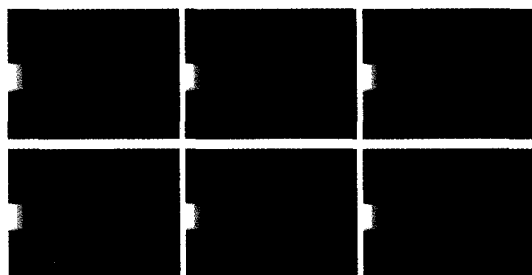


图4 同等光子数量下效果比较



(a) 未使用自适应光子 400 pass (b) 使用自适应光子发射 100 pass

图5 同等绘制时间比较

图6为一室外场景, 光源在平面的右上方, 左右两图绘制时间相同。由于场景中的圆环使用的是分段建模, 其焦散形状不同于光滑圆环。可以看到在离光源较近的平面右侧, 以及焦散等光子容易聚集的区域, 二者都能收敛较好, 但是在离光源较远的左边区域使用自适应光子发射的结果图噪声明显小于不使用的情况。

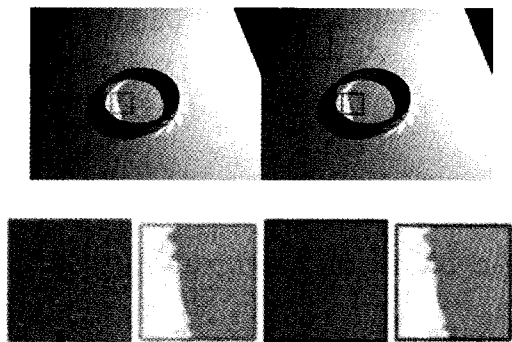


图6 室外场景比较

为了测试使用 hitpoints 累计收集光子信息进行权重调制后的效果, 建立了图7的实验场景, 在平面上方放置一点光源, 如图7(a)所示。图(b), 图(c)为光子统计温度图, 颜色由暗到亮分别表示收集到的光子由少到多。比较图(b), 图(c)两图可以发现不使用累计光子进行调制时, 光子会大量集中于中心区域, 而使用了累计光子进行调制后光子分布变得更为平滑, 这样可以减少了光子浪费, 提高了绘制效率。



图7 调制结果比较

4 结束语

从上述实验可以看出, 利用渐进式光子映射的固有信息以及其多遍的特点, 指导光子发射, 使每遍发射的光子尽可能多的分布在视点相关的区域以及缺乏光子的区域可以很大的提升渐进式光子映射算法的绘制效率。今后我们可以考虑根据现有光子路径权重估计的算法, 使用 metropolis sampling 方法, 通过扰动现有光子路径直接在路径层面上进行采样发射光子, 这样可以省去对光源分层的步骤, 并可能能够获取更精确的基于权重概率的路径分布, 同时考虑使用图形处理器对算法进行整体加速。

参考文献:

[1] Matt Pharr, Greg Humphreys. Physically based rendering;

From theory to implementation [M]. 2nd ed. Morgan Kaufmann Publishers Inc, 2010.

[2] Dutre P, Bekaert P, Bala K. Advanced global illumination [M]. 2nd ed. USA: A K Peters Ltd, 2006.

[3] Aila T, Laine S. Understanding the efficiency of ray traversal on GPUs [C]. Proceedings of High-Performance Graphics, 2009: 145-149.

[4] Horn D R, Sugerman J, Houston M, et al. Interactive k-d tree GPU raytracing [C]. Proceedings of the Symposium on Interactive 3D Graphics and Games. New York, NY, USA: ACM, 2007: 167-174.

[5] Popov S, GU Nther J, Seidel H-P, et al. Stackless kd-tree traversal for high performance GPU ray tracing [C]. Computer Graphics Forum (Proceedings of Eurographics), 2007, 26 (3): 415-424.

[6] Bigler J, Stephens A, Parker S G. Design for parallel interactive ray tracing systems [C]. Proceedings of the IEEE Symposium on Interactive Ray Tracing, 2006: 187-196.

[7] Djeu P, Hunt W, WANG R, et al. Razor: An architecture for dynamic multiresolution ray tracing [R]. Tech Rep TR-07-52, 2007.

[8] Bikker J. Real-time ray tracing through the eyes of a game developer [C]. Proceedings of the IEEE Symposium on Interactive Ray Tracing, 2007: 1-10.

[9] Parker S, Bigler J, Dietrich A, et al. Optix: A general purpose ray tracing engine [C]. ACM Transactions on Graphics, 2010.

[10] Cline D, Talbot J, Egbert P. Energy redistribution path tracing [J]. ACM Trans Graph (SIGGRAPH Proceedings), 2005, 24 (3): 1186-1195.

[11] LAI Y-C, FAN S H, Chenney S, et al. Photorealistic image rendering with population Monte Carlo energy redistribution [C]. Proceedings of the Eurographics Symposium on Rendering, 2007: 287-295.

[12] Havran V, Herzog R, Seidel H-P. Fast final gathering via reverse photon mapping [C]. Computer Graphics Forum (Proceedings of Eurographics), 2005, 24 (3): 323-333.

[13] Fradin D, Meneveaux D, Horna S. Out of core photon-mapping for large buildings [C]. Proceedings of Eurographics Symposium on Rendering, 2005.

[14] Hachisuka T, Ogaki S, Jensen H W. Progressive photon mapping [J]. ACM Transactions on Graphics (SIGGRAPH Asia Proceedings), 2008, 27 (5): 130.

[15] Hachisuka T, Jensen H W. Stochastic progressive photon mapping [C]. SIGGRAPH Asia' 09 ACM SIGGRAPH Papers. New York, NY, USA: ACM, 2009: 1-8.

[16] Fan Shaohua, Stephen Chenney, Lai Yu-Chi. Metropolis photon sampling with optional user guidance [C]. Eurographics Symposium on Rendering, 2005: 127-138.