重要性驱动的实时间接光泽反射算法

陈静[1,2)], 陈主昕[1,2)], 郭帆[1,2)], 张严辞[1,2)*]

[1)] (四川大学计算机学院 成都 610065)
[2)] (四川大学视觉合成图形图像技术国家重点学科实验室 成都 610065)
(yczhang@scu.edu.cn)

摘 要: 实时间接光泽反射效果有助于提高渲染场景的真实感. 为了解决虚拟点光源(virtual point lights, VPLs)在光泽材质处的着色走样问题, 提出基于 VPLs 光照贡献重要性的渲染算法. 首先提出基于 Tile 的光照贡献估算方法, 通过分块拟合和分块剔除快速估算 VPLs 的光照贡献; 然后通过自适应权重的时域复用方法, 生成重要性驱动的分布图; 最后结合 Mipmap 多级纹理映射技术, 快速在场景的不同区域生成不同密度的 VPLs. 在金属戒指和斯坦福兔子等场景下的实验结果表明, 使用相同数量的 VPLs 进行渲染, 所提算法渲染质量优于随机光源裁剪算法, 渲染结果的均方根误差不超过 2%.

关键词: 间接光泽反射; 重要性驱动; 虚拟点光源; 实时渲染
中图法分类号: TP391.41 **DOI:** 10.3724/SP.J.1089.2023.19757

Importance-Driven Real-Time Indirect Glossy Reflection Algorithm

Chen Jing[1,2)], Chen Zhuxin[1,2)], Guo Fan[1,2)], and Zhang Yanci[1,2)*]

[1)] (*College of Computer Science, Sichuan University, Chengdu* 610065)
[2)] (*National Key Laboratory of Fundamental Science on Synthetic Vision, Sichuan University, Chengdu* 610065)

Abstract: Rendering indirect glossy reflection in real-time helps to improve the realism of rendered scenes. To solve the problem of virtual point lights (VPLs) shading at glossy materials, a rendering algorithm based on the importance of VPLs lighting contribution is proposed. Firstly, a tile-based lighting contribution estimation method is proposed, which quickly estimates the light contribution of VPLs through tile fitting and tile culling. Secondly, an importance-driven distribution map is generated by an adaptive weight temporal algorithm. Finally, the Mipmap multi-level texture mapping method quickly generates different densities of VPLs in different areas in the scene. The experimental results on metal ring and Stanford rabbit scenes indicate that compared with stochastic light culling algorithm, the proposed algorithm achieves better rendering quality with the same number of VPLs, and the root mean square error of the rendering results in this paper are no more than 2%.

Key words: indirect glossy reflection; importance driven; virtual point lights; real-time rendering

随着工业应用对高真实感渲染场景的要求, 间接光泽反射效果变得越来越重要. 在实时渲染中, 基于虚拟点光源(virtual point lights, VPLs)的即时辐射度算法常被用于间接光照计算.

收稿日期: 2022-06-08; 修回日期: 2022-08-03. **基金项目**: 国家重大项目(GJXM92579). **陈静**(1998—), 女, 硕士研究生, 主要研究方向为计算机图形学; **陈主昕**(1996—), 男, 硕士, 主要研究方向为计算机图形学; **郭帆**(1999—), 男, 硕士研究生, 主要研究方向为计算机图形学; **张严辞**(1975—), 男, 博士, 教授, 博士生导师, CCF 会员, 论文通信作者, 主要研究方向为计算机图形学、3D 游戏技术、虚拟现实.

VPLs 在场景中的合理分布是渲染间接光泽反射的关键. 随机光源裁剪(stochastic light culling, SLC)算法[1]在裁剪过程中虽然可以实时生成间接光泽反射效果, 但是没有充分考虑 VPLs 的分布, 会造成在光泽材质处的着色走样. 原因是 VPLs 在漫反射材质处均匀辐射能量, 而在光泽材质处朝特定方向辐射能量, 使得间接光泽反射效果和方向紧密关联, 导致相同数量的 VPLs 在光泽材质处会出现明暗相间的走样现象.

为了解决上述问题, 本文提出重要性驱动的实时间接光泽反射算法, 依据 VPLs 光照贡献的重要性在场景中合理分布其位置, 生成重要性驱动的分布图(importance driven distribution map, IDDM), 进而根据 IDDM 指导 VPLs 在场景中的分布. 本文引入基于 Tile 的光照贡献估算方法, 通过分块拟合和分块剔除的方法快速地估算光照贡献, 满足实时性的要求; 采用自适应权重的时域复用算法高效、准确地生成 IDDM, 减少场景的闪烁; 采用 Mipmap 多级纹理映射技术在 IDDM 的指导下快速生成合理分布的 VPLs, 渲染得到高质量的间接光泽反射效果.

1　相关工作

1.1　VPLs

全局光照在图形学渲染领域中广受关注, 其中, 间接光照计算更是相关学者研究的重点. Keller[2]提出通过 VPLs 拟随机游走的方法计算间接光照, 但是由于其计算耗时, 因此并不适用于实时全局光照. 为满足实时性, Dachsbacher 等[3]提出反射阴影贴图(reflective shadow map, RSM)算法, 将 RSM 中的每个像素作为间接光源, 通过采样和屏幕空间插值的方法实现动态场景的间接光照, 这也是后续很多算法的基础.

Lensing[4]提出使用 LightSkin 解决漫反射和光泽反射的全局光照, 并将其运用到虚拟现实和增强现实的应用中, 但是高分辨率的 RSM 导致帧率明显下降. 随后, Bischoff 等[5]使用一组稀疏分布的光照缓存累积计算间接光照来解决该问题; 同时, 使用重要性采样减少 VPLs 的数量, 用于提高在开放世界中的间接光照和阴影质量. Sriwasansak 等[6]使用光子溅射补偿 VPLs 的能量损失, 使其在漫反射场景中更加高效、准确. 在 AR/MR 等移动设备中, Schwandt 等[7]使用单个相机和预计算生成环境贴图, 实现了高质量的光泽反射.

1.2　VPLs 的生成

由于实时性的要求, 场景中生成 VPLs 的数量有限, 因此其生成方式被学者广泛研究. Keller[2]采用光线追踪的方法生成 VPLs; Dachsbacher 等[3]使用 RSM, 通过光栅化在场景中快速生成大量的 VPLs; Laine 等[8]使用梅特波利斯光照传输[9]方法的变体来放置 VPLs; Simon 等[10]使用 2 张分别来自光源空间和相机空间的光子贴图来生成 VPLs.

很多学者通过不同的采样方法在场景中分布 VPLs. Georgiev 等[11]采用拒接采样的方法, 利用 VPLs 对最终图像的贡献比例使更多的 VPLs 集中在照亮的区域; Ritschel 等[12]提出视图自适应的 VPLs 生成方法, 使用重要性采样从视图样本和双向 RSM 中选择 VPLs; Davidovič 等[13]追踪相机路径来分布 VPLs, 进而渲染光泽反射; Hedman 等[14]利用帧间相关性, 在多帧上采样生成 VPLs, 并通过复用历史帧中的信息来维持时间上的稳定性, 减少场景闪烁; Babahenini 等[15]将多重重要性采样和逆变换法相结合, 采样高光照贡献的 VPLs; Jin 等[16]提出混合采样算法, 根据相机的位置和方向分成近景区和远景区, 在不同的区域采用不同的采样方式, 使其合理分布 VPLs. 上述方法大多是通过采样来分布 VPLs, 但是采样方法必然会带来计算开销.

1.3　光照贡献

由于场景中每个光源对最终渲染结果的贡献程度不同, 学者往往通过减少间接光照计算中 VPLs 的数量来优化算法效率.

(1) 光源聚类. 光源聚类是常用算法, Walter 等[17]首先提出 Lightcuts 算法, 根据光源在场景中的位置和方向构建光源树, 遍历光源树的内部节点用于间接光照计算; 之后, Walter 等[18]提出 Multidimensional Lightcuts 算法, 通过构建隐式层次结构处理更复杂的渲染效果. Huo 等[19]在光源和像素上构建层次结构, 实现了复杂场景的光泽相互反射效果.

(2) 光源剔除. 光源剔除也是常用的加速算法. Harada[20]通过对深度进行分块, 裁剪掉没有贡献的光源; 之后, O'Donnell 等[21]不仅在屏幕空间细分, 而且沿视线方向细分, 从而进一步减少光源数量. Tokuyoshi 等[22]使用俄罗斯赌盘随机确定光源范围, 解决因光源范围限制导致的着色偏暗的问题.

2　本文算法

2.1　概　述

与间接漫反射相比, 间接光泽反射因其辐射

强度和方向的强关联性，导致其对 VPLs 的分布要求更高. 光泽材质处的 VPLs 非均匀辐射出能量，导致其只会对位于辐射范围内的着色点有光照贡献，如图 1 所示. SLC 算法[1]虽然通过随机光源范围剔除光泽材质处不重要的 VPLs，但是没有考虑 VPLs 在场景中的分布，容易造成着色走样现象，如图 2 所示. 因此，使 VPLs 在场景中自适应分布，满足光泽材质处的着色分布要求，是渲染间接光泽反射的一大难点.

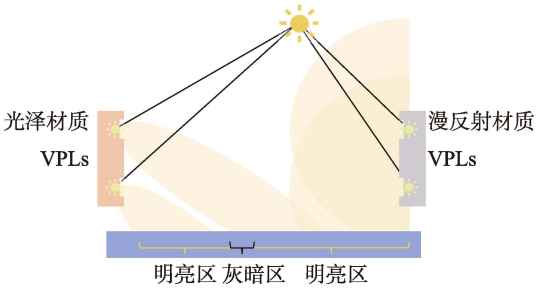


图 1 VPLs 在不同材质处的辐射强度示意图

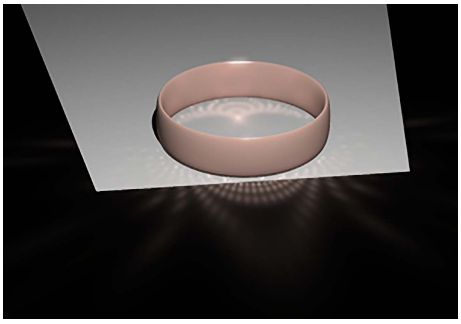


图 2 光泽材质处的着色走样

针对上述难点，本文提出重要性驱动的实时间接光泽反射算法，根据光照贡献指导 VPLs 在场景中的分布，使得 VPLs 较多地聚集在高贡献区域. 实现这个目的的关键是依据 VPLs 的光照贡献生成 IDDM，该图以光源为视点，记录在场景中不同区域的 VPLs 对各个着色点的重要性值，重要性值越大，表明该区域的 VPLs 对着色点的影响越大，需要增加该区域中 VPLs 的数量，该重要性值依据 VPLs 的光照贡献计算. 由于在未生成 VPLs 之前很难计算其光照贡献，本文利用历史帧中 VPLs 的信息计算重要性值.

为了计算生成 IDDM，并基于 IDDM 生成 VPLs，需要解决 3 个问题.

(1) 快速计算 VPLs 的光照贡献. VPLs 的光照贡献计算范围是全部着色点，准确计算无法满足实时的要求. 因此，本文提出基于 Tile 的光照贡献估算方法.

(2) 减少场景闪烁. 实时渲染中，光源的移动会造成相邻帧之间 IDDM 变化较大，造成场景闪烁. 为此，本文在生成 IDDM 时引入自适应的权重，结合时域复用可有效地抑制场景闪烁.

(3) 基于 IDDM 快速生成 VPLs. 本文利用 Mipmap 多级纹理映射技术生成多层级 IDDM，通过分层调整生成符合重要性分布的 VPLs.

本文利用重要性驱动来分布 VPLs，极大地提高了间接光泽反射的渲染质量. 本文算法渲染流程如图 3 所示，其中，IDDM 使用灰度图表示，可视化为第 3 张图.

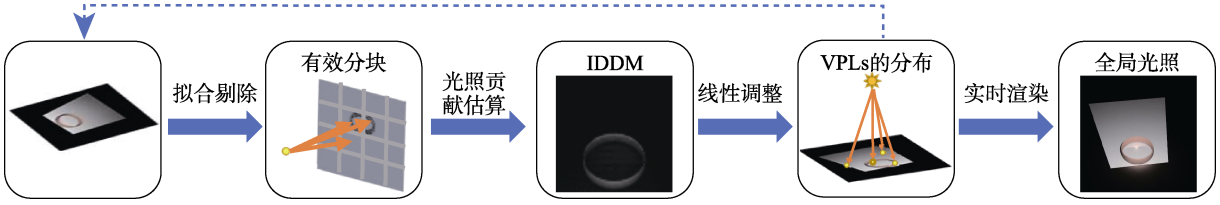


图 3 本文算法渲染流程

2.2 基于 Tile 的光照贡献估算方法

VPLs v 的光照贡献 C_v 依赖屏幕空间中所有着色点，公式为

$$C_v = \frac{1}{N}\sum_{i=1}^{N} L_v(\omega_i) G(p_v, p_i) \langle \omega_i, n_i \rangle .$$

其中，N 表示着色点数量；ω_i，p_i，n_i 分别表示 v 到着色点 i 的入射方向、着色点 i 的位置及其法线；$G(p_v, p_i)$ 表示 v 与第 i 个着色点之间的几何项；$L_v(\omega_i)$ 表示 v 在 ω_i 方向的辐射率. 由于光泽材质处 VPLs 非均匀地辐射出能量，即 L_v 与 ω_i 紧密关联，导致需要根据着色点的位置来计算 $L_v(\omega_i)$，进一步增加了计算开销.

为了满足实时性，本文提出基于 Tile 的光照贡献估算方法，通过减少参与计算的着色点数量来提高计算效率. 对屏幕空间中的着色点进行分块拟合，得到该分块的代表点. 估算 VPLs 光照贡献时，使用代表点代替该分块中的所有着色点进行计算.

利用最小二乘法对屏幕空间进行分块拟合，根据文献[23-24]求解得到代表点的位置和法线.

然而，拟合计算必然带来估算误差，对于位置变化较大或法线差异较大的分块，VPLs 对其近似的光照贡献会出现较大的误差. 为解决该问题，本文通过方差校正的方法减少估算误差，计算公式为

$$\begin{cases}\sigma_{\hat{p}}=\sum_{i=1}^{M}\left(p_i-\hat{p}\right)^2\\ \sigma_{\hat{n}}=\sum_{i=1}^{M}\left(n_i-\hat{n}\right)^2\\ C_v\approx\frac{1}{\hat{N}}\sum_{i=1}^{\hat{N}}\frac{L_v\left(\omega_i\right)G\left(p_v,\hat{p}_i\right)\left\langle\omega_i,\hat{n}_i\right\rangle}{1+\sigma_{\hat{p}_i}\sigma_{\hat{n}_i}}\end{cases}\tag{1}$$

其中，M 表示分块中着色点数量；$\hat{p}$ 和 $\hat{n}$ 分别表示某个分块中代表点的位置和法线；$\hat{N}$ 表示屏幕空间中的分块数量；$\hat{p}_i$ 和 $\hat{n}_i$ 分别表示第 i 个分块中代表点的位置和法线.

由于 VPLs 的光照影响范围有限，并不是 VPLs 对所有的分块都有贡献. 为了进一步加速计算，本文结合分块剔除的思想，只计算与 VPLs 光照影响范围相交的有效屏幕分块. 因此，将式(1)修改为

$$\begin{cases}C_v\approx\frac{1}{\hat{N}'}\sum_{i=1}^{\hat{N}'}\frac{L_v\left(\omega_i\right)G\left(p_v,\hat{p}_i\right)\left\langle\omega_i,\hat{n}_i\right\rangle}{1+\sigma_{\hat{p}_i}\sigma_{\hat{n}_i}}\\ \text{s.t. } s\left(p_i,n_i\right)\in I\end{cases}.$$

其中，$\hat{N}'$ 表示有效分块数量；$s\left(p_i,n_i\right)$ 表示代表点的位置和法线；I 表示有效分块集合.

2.3　自适应的时域复用算法

VPLs 在场景中的分布要与光照贡献成正比，但是在当前帧生成 VPLs 之前很难计算其光照贡献. 为了解决该问题，本文利用渲染过程中相邻帧之间高度相关的特点，复用历史帧中 VPLs 的信息生成当前帧的 IDDM，并且采用自适应的复用策略来保持其时间稳定性.

2.3.1　生成 IDDM

为了方便计算光照贡献，在光源空间中生成 IDDM；为了提高计算效率，生成低分辨率的重要性图.

本文根据上一帧 VPLs 的光照贡献在光源空间中生成当前帧的 IDDM. 首先在光源空间中生成低分辨率的二维纹理；然后将上一帧的全部 VPLs 投影到该纹理贴图中，计算其在对应纹素中所提供的光照贡献，并且为了防止 VPLs 在局部区域中密度过高，进行平滑处理，计算公式为

$$f\left(C_i\right)=\begin{cases}C_{\min}, & \text{if } C_i<C_{\min}\\ C_{\max}, & \text{if } C_i>C_{\max}\\ C_i, & \text{otherwise}\end{cases}.$$

其中，$f\left(C_i\right)$ 表示第 i 个 VPLs 的光照贡献；$C_{\max}$ 和 $C_{\min}$ 分别表示用户自定义的最大光照贡献值和最小光照贡献值，是 VPLs 分布的控制量. 最后，根据纹素中的光照贡献计算其重要性值，n 个 VPLs 投影到 IDDM 纹素 t 的重要性值计算公式为

$$E_t=\sum_{i=0}^{n}\omega_i f\left(C_i\right).$$

其中，ω_i 表示第 i 个 VPLs 的重要性权重，为了简化计算，权重取常值为 1 的加权平均.

2.3.2　自适应方法

在动态场景中，光源的移动会导致上一帧的 VPLs 在当前帧的投影位置发生变化，而相邻帧之间 IDDM 差异过大会造成间接光照闪烁. 为了保持 IDDM 在时间上的稳定性，本文基于时域复用的思想对历史帧的 IDDM 进行复用，公式为

$$E=\alpha E+\left(1-\alpha\right)E'\tag{2}$$

其中，α 表示混合权重；E' 表示历史帧对应纹素的重要性值.

确定混合权重 α 是关键. 一方面，光源的移动会改变相邻 2 帧中 VPLs 反射方向，如图 4a 所示；另一方面，光源的移动可能会造成上一帧的 VPLs 在当前帧完全失效，如图 4b 所示.

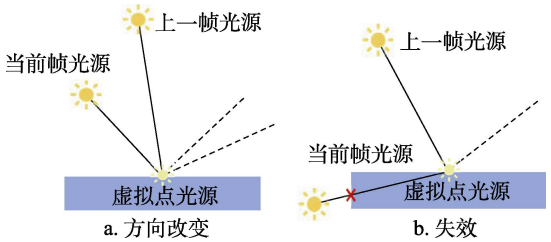


图 4　光源移动造成的误差

为了充分地考虑光源移动轨迹的变化，本文提出自适应的复用策略. 根据相邻 2 帧之间光源位置、VPLs 位置以及其法线所形成的角度变化来调整混合权重. 即在角度变化较小时，提高历史帧的权重；在角度变化太大时，提高当前帧的权重. 纹素中第 i 个 VPLs 的混合权重计算公式为

$$\alpha_i=\left(1-\Theta\left(\boldsymbol{\omega}_v,\boldsymbol{\omega}_v',\boldsymbol{n}_v\right)\right)\times\alpha+\Theta\left(\boldsymbol{\omega}_v,\boldsymbol{\omega}_v',\boldsymbol{n}_v\right).$$

其中，$\boldsymbol{\omega}_v$ 和 $\boldsymbol{\omega}_v'$ 分别表示当前帧光源、上一帧光源到 VPLs 的入射方向；$\boldsymbol{n}_v$ 表示 VPLs 处的法线；$\Theta\left(\boldsymbol{\omega}_v,\boldsymbol{\omega}_v',\boldsymbol{n}_v\right)$ 表示角度变化函数，考虑 VPLs 失效的情况，其计算公式为

$\Theta(\boldsymbol{\omega}_v,\boldsymbol{\omega}'_v,\boldsymbol{n}_v)=\left(1-\max\left(\langle\boldsymbol{\omega}'_v,\boldsymbol{\omega}_v\rangle,0\right)\right)\times H\left(\langle\boldsymbol{\omega}_v,\boldsymbol{n}_v\rangle\right)$，$H\left(\langle\boldsymbol{\omega}_v,\boldsymbol{n}_v\rangle\right)$ 表示赫维赛德函数，即

$$H\left(\langle\boldsymbol{\omega}_v,\boldsymbol{n}_v\rangle\right)=\begin{cases}0, & \text{if}\ \langle\boldsymbol{\omega}_v,\boldsymbol{n}_v\rangle\leqslant 0\\ 1, & \text{otherwise}\end{cases}.$$

因此，引入自适应混合权重后，式(2)纹素中第 i 个 VPLs 的重要性值计算修改为 $E_i=\alpha_i E_i+(1-\alpha_i)E'$．至此，IDDM 计算完成.

2.4 基于 Mipmap 多级映射技术生成 VPLs

在 IDDM 的指导下，通过光线追踪技术引导 VPLs 的生成方向，从而间接控制其在场景中的位置.

2.4.1 Mipmap 多级纹理映射技术

根据 IDDM 相邻方向块的重要性比值，对初始随机均匀分布的 VPLs 的方向进行调整，使其符合重要性分布.

Mipmap 是 1 棵多级的四叉树，每个内部节点都指向 4 个子节点，而节点中的重要性值表示 4 个缩放因子．根据缩放因子对节点中样本的方向进行线性调整，可得到指定密度的样本，单个节点调整流程如图 5 所示．首先采样得到 4 个子节点的缩放因子，同时初始化样本；然后计算垂直划分轴和水平划分轴；最后根据划分轴调整样本.

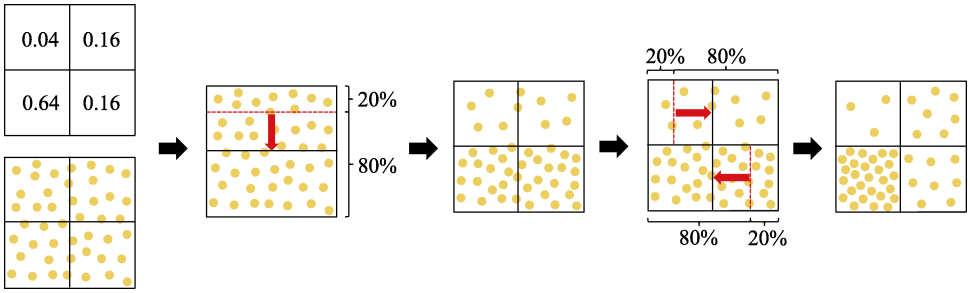


图 5 单个节点调整流程

当自上而下地遍历 Mipmap 时，可以快速地迭代调整方向，使得样本符合重要性分布．对指定层级进行调整时，需要对 2 个纹理坐标进行处理．假设 1 个样本 s_t 的纹理坐标为 (u_t,v_t)，经过调整之后，得到样本 s_{t+1} 的纹理坐标为 (u_{t+1},v_{t+1})，计算公式为

$$\begin{cases}u_{t+1}=\\ \quad\begin{cases}\left(1-\dfrac{u_{\text{mid}}-u_{\min}}{u_s-u_{\min}}\right)u_{\min}+\dfrac{u_{\text{mid}}-u_{\min}}{u_s-u_{\min}}u_t, & \text{if}\ \ u_t<u_s\\ \left(1-\dfrac{u_{\max}-u_{\text{mid}}}{u_{\max}-u_s}\right)u_{\max}+\dfrac{u_{\max}-u_{\text{mid}}}{u_{\max}-u_s}u_t, & \text{otherwise}\end{cases}\\ v_{t+1}=\\ \quad\begin{cases}\left(1-\dfrac{v_{\text{mid}}-v_{\min}}{v_s-v_{\min}}\right)v_{\min}+\dfrac{v_{\text{mid}}-v_{\min}}{v_s-v_{\min}}v_t, & \text{if}\ \ v_t<v_s\\ \left(1-\dfrac{v_{\max}-v_{\text{mid}}}{v_{\max}-v_s}\right)v_{\max}+\dfrac{v_{\max}-v_{\text{mid}}}{v_{\max}-v_s}v_t, & \text{otherwise}\end{cases}\end{cases}\tag{3}$$

其中，$u_{\max}$ 和 $u_{\min}$ 分别表示该内部节点 u 坐标的最大和最小值；$v_{\max}$ 和 $v_{\min}$ 分别表示该内部节点 v 坐标的最大和最小值；u_{mid} 和 v_{mid} 分别表示 u 坐标和 v 坐标的中点；u_s 和 v_s 分别表示该内部节点 u 坐标和 v 坐标划分轴的值．假设 Mipmap 中 4 个子节点的缩放因子分别为 γ_{u^-,v^-}，γ_{u^+,v^-}，γ_{u^-,v^+} 和 γ_{u^+,v^+}，则划分轴的计算公式为

$$\begin{cases}v_s=v_{\min}+\dfrac{\gamma_{u^-,v^-}+\gamma_{u^+,v^-}}{\gamma_{u^-,v^-}+\gamma_{u^+,v^-}+\gamma_{u^-,v^+}+\gamma_{u^+,v^+}}(v_{\max}-v_{\min})\\ u_s=\begin{cases}u_{\min}+\dfrac{\gamma_{u^-,v^-}}{\gamma_{u^-,v^-}+\gamma_{u^+,v^-}}(u_{\max}-u_{\min}), & \text{if}\ \ v_t<v_s\\ u_{\min}+\dfrac{\gamma_{u^-,v^+}}{\gamma_{u^-,v^+}+\gamma_{u^+,v^+}}(u_{\max}-u_{\min}), & \text{otherwise}\end{cases}\end{cases}.$$

根据 IDDM 的 Mipmap 树逐层自上而下地对 N 个样本进行线性迭代调整之后，结合光线追踪技术，即可在场景的指定区域生成 VPLs.

2.4.2 优化效率

根据观察，逐层进行线性调整是不必要的操作，只需要计算其内部节点的变换矩阵，并在叶节点中进行最终计算即可．对于式(3)，假设一个位于左下子节点中的样本 s_t，即 $(u_t<u_s,v_t<v_s)$，其 v 坐标的调整操作为

$$\begin{cases}C_v^{\text{scale}}=\dfrac{v_{\text{mid}}-v_{\min}}{v_s-v_{\min}}\\ C_v^{\text{add}}=\left(1-\dfrac{v_{\text{mid}}-v_{\min}}{v_s-v_{\min}}\right)v_{\min}\\ v_{t+1}=C_v^{\text{add}}+C_v^{\text{scale}}v_t\end{cases}.$$

那么，从样本 s_t 变换到 s_{t+1} 的过程就可以用三阶矩

阵 $\boldsymbol{M}$ 描述为

$$\begin{bmatrix} u_{t+1} \\ v_{t+1} \\ 1 \end{bmatrix} = \begin{bmatrix} C_u^{\text{scale}} & 0 & C_u^{\text{add}} \\ 0 & C_v^{\text{scale}} & C_v^{\text{add}} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_t \\ v_t \\ 1 \end{bmatrix}.$$

在 n 层的 Mipmap 树中，方向样本 s_0 从根节点开始，直至叶子节点的迭代调整过程，就可以用一组累积的矩阵表示为 $s_n = \boldsymbol{M}_n \cdots \boldsymbol{M}_1 s_0$.

在对样本进行调整时，只需要判断样本属于哪个方向块，然后进行一次矩阵乘法，这样将显著提高线性调整的计算效率.

3　实验结果与分析

通过在多个存在间接光泽反射效果的场景中进行实验，对本文提出的重要性驱动的实时间接光泽反射算法进行分析. 实验环境为 NVIDIA RTX 2060, Intel Xeon E3-1230, 16 GB 内存.

实验中，采用实时渲染领域常见的金属戒指、康奈尔盒和斯坦福兔子场景；图片输出参数为 1 024×1 024 像素，默认 VPLs 的数量为 16 384，IDDM 分辨率为 64×64 像素，Tile 数量为 16×16；使用 262 144 个 VPLs 计算的间接光照作为 Ground Truth 的渲染结果；使用均方根误差(root mean square error, RMSE)作为渲染质量的衡量标准.

3.1　渲染效果和效率对比

为了验证本文算法的渲染效果和渲染效率，将其与 RSM 算法[3]和 SLC 算法[1]进行对比实验.

3.1.1　渲染效果比较

为了验证本文算法的渲染结果在间接光泽反射的表现以及误差上具备更高的正确性，在上述场景中使用相同数量的 VPLs 渲染间接光照，效果如表 1 所示. 可以看出，与 RSM 算法[3]和 SLC 算法[1]相比，本文算法渲染效果更优，间接光照的表现更接近 Ground Truth 的结果, RMSE 不超过 2%. 这是由于 RSM 算法只考虑间接漫反射，从而缺乏间接光泽反射效果；SLC 算法生成的 VPLs 均匀分布在光源空间中，导致光泽材质处的 VPLs 分布稀疏，渲染质量较差；本文算法根据上一帧 VPLs 的光照贡献指导当前帧 VPLs 的分布，使得其分布更加合理，从而保证了焦散等特殊的间接光泽反射效果. 康奈尔盒场景和斯坦福兔子场景由于场景几何比较复杂，影响视觉效果，因此无法直接从视觉上看到本文算法和 SLC 算法的显著差异，但是 RMSE 存在较大差别.

3.1.2　渲染效率比较

本文算法在整个渲染流程中增加了 VPLs 的光照贡献计算过程、IDDM 生成以及 VPLs 生成等，因此，在计算过程中会带来一定的计算开销. 通过对比本文算法和 SLC 算法[1]在相似误差渲染的场景中时间上的差异，验证本文算法的高效性，实验结果如表 2 所示. 可以看出，金属戒指场景下本文算法的渲染效率比 SLC 算法低，这是因为金属戒指场景中表面较为平整，通过增加一定数量的 VPLs, SLC 算法就可以较大地提升渲染质量；而对于康奈尔盒和斯坦福兔子场景，本文算法具有更高的间接光照计算效率，这是因为在这 2 个场景中，光泽材质处的几何形状较为复杂，本文算法通过控制 VPLs 的生成方向保证其在光泽材质表面的密度，从而提高了计算效率. 因此，与 SLC 算法相比，本文算法具有更高的自适应性，可以更好地调整 VPLs 在场景中的分布. 从康奈尔盒场景的结果可以看出，在对面墙壁和地面区域中，虽然本文算法比 SLC 算法误差大，但是 SLC 算法的误差主要集中在龙身上，会形成显著亮斑现象，影响视觉效果；而本文算法的误差被分布到了更大的区域，对视觉效果的影响较小，并且本文算法的 RMSE 小于 SLC 算法.

3.2　基于 Tile 的光照贡献估算方法的分析

3.2.1　验证拟合算法的准确性

对拟合的位置和法线进行误差计算，位置的误差 e_{p} 和法线偏差 e_{n} 计算公式分别为

$$\begin{cases} e_{\text{p}} = \sqrt{\dfrac{\sum_{i=1}^{m} (\hat{p} - p_i)^2}{m}} \Big/ (1+\sigma_{\text{p}}) \\ e_{\text{n}} = \dfrac{1}{m} \sum_{i=1}^{m} \langle \hat{\boldsymbol{n}}, \boldsymbol{n}_i \rangle / (1+\sigma_{\text{n}}) \end{cases}.$$

其中，m 表示该分块中着色点数量；p_i 和 $\boldsymbol{n}_i$ 分别表示分块中第 i 个着色点的位置和法线；$\hat{p}$ 和 $\hat{\boldsymbol{n}}$ 分别表示分块拟合的代表点的位置和法线；σ_{p} 和 σ_{n} 分别表示分块拟合结果的位置方差和法线方差. 实验结果如表 3 所示. 可以看出，本文算法能够快速拟合；此外，由于拟合算法是基于 Tile 的算法，与具体的场景无关，因此计算开销相对稳定.

3.2.2　验证光照贡献估算方法的高效性

将本文提出的光照贡献估算方法与准确的光照贡献计算结果进行比较. 估算结果 C' 与准确结果 C 之间的误差 e_{c} 计算公式为

$$e_{\text{c}} = \frac{|C - C'|}{C} \times 100\%.$$

表 1 3 种算法使用相同数量的 VPLs 渲染效果对比

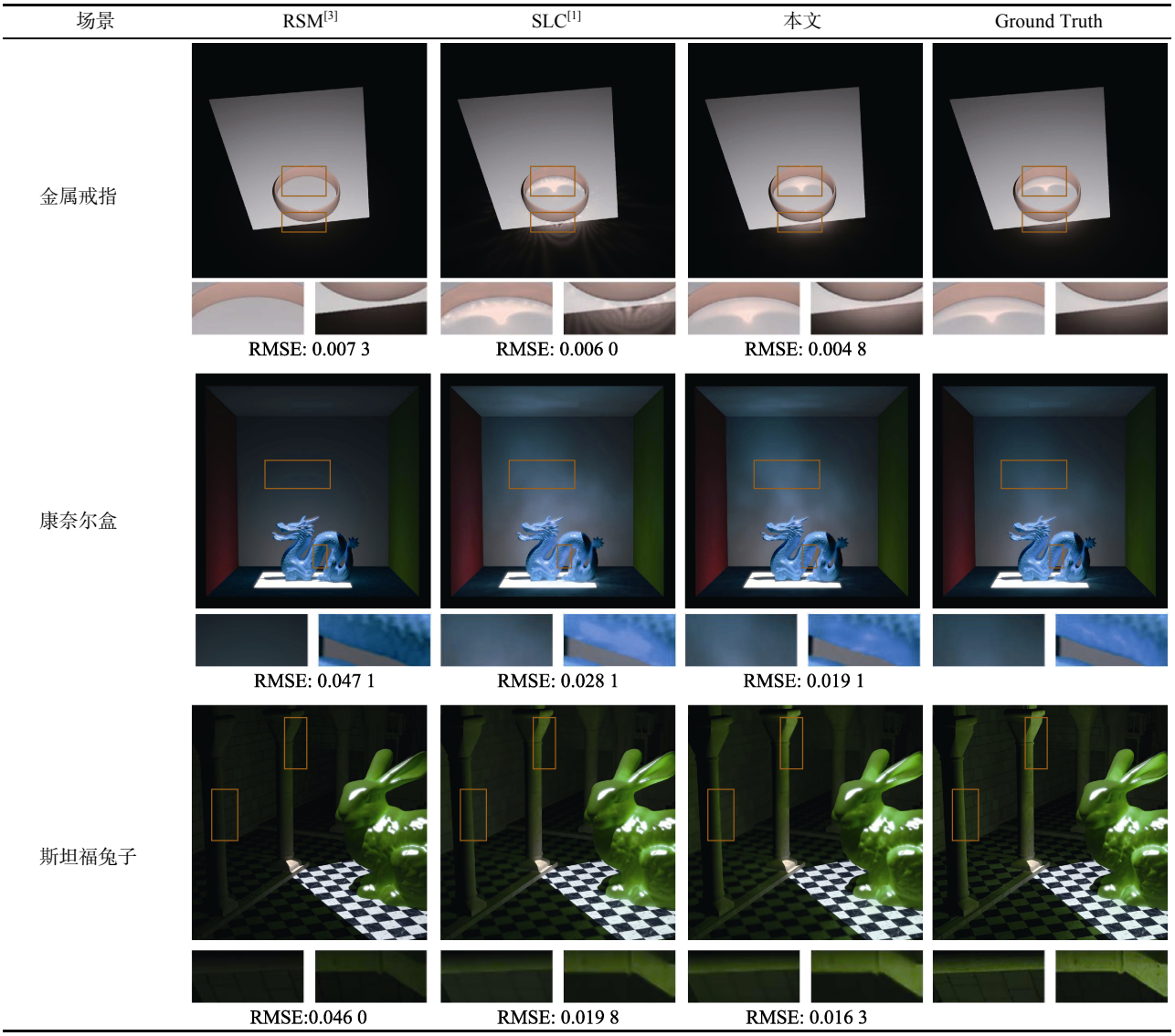


表 2 相似误差下的对比

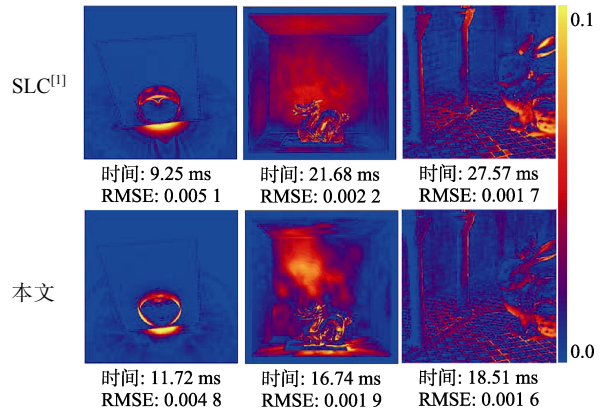


表 3 屏幕空间拟合算法的误差和时间

场景	位置误差	法线误差	拟合时间/ms
金属戒指	0.007 20	0.000 18	1.04
康奈尔盒	0.019 19	0.000 90	0.92
斯坦福兔子	0.219 30	0.003 29	1.06

表 4 VPLs 光照贡献估算方法的时间和误差

场景	时间/ms		误差/%	
	估算	准确	平均	最大
金属戒指	1.71	90.43	0.187 6	1.348 1
康奈尔盒	1.72	91.40	3.557 0	4.886 7
斯坦福兔子	1.78	90.46	1.300 4	5.235 3

实验结果如表 4 所示. 可以看出, 本文算法将计算效率提高约 50 倍; 同时, 在 3 个场景中最大误差不超过 6%. 这是因为本文只需要根据有效分块来计算 VPLs 的光照贡献, 而不需要遍历每个着色点. 因此, 本文提出的基于 Tile 的光照贡献估算方法能够在合理的误差范围内极大地提高计算效率.

3.3 自适应的时域复用算法的分析

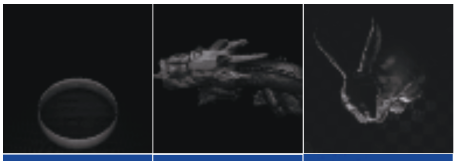
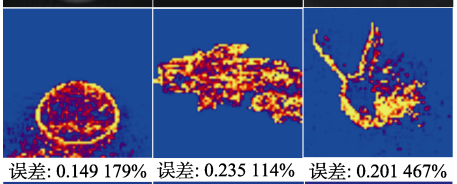
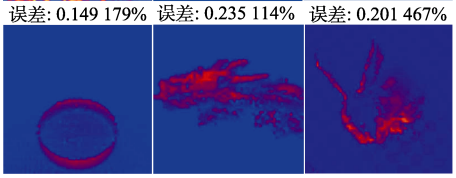
验证使用自适应的时域复用算法生成的 IDDM

能够减少间接光照闪烁. 对光源添加随机抖动模拟光源的移动，将使用时域复用生成的 IDDM 和未使用时域复用生成的IDDM，在连续帧之间的变化情况作为衡量标准. 误差计算公式为

$$e_t = \frac{1}{n-1}\sum_{i=2}^{n}\left|E_i - E_{i-1}\right|.$$

其中，n 表示帧数，实验中 $n=10$；E_i 和 E_{i-1} 分别表示当前帧和上一帧生成的 IDDM 中的重要性值. IDDM 可视化为灰度图，颜色偏白的区域是 VPLs 高贡献区域，颜色偏黑的区域是VPLs低贡献区域，实验结果如表 5 所示. 可以看出，本文提出的自适应的时域复用算法，使得相邻帧之间生成的IDDM变化较小，从而可以有效地减轻间接光照闪烁的问题.

表 5　动态光源导致的 IDDM 变化示意图

算法	金属戒指	康奈尔盒	斯坦福兔子
IDDM			
未使用时域复用			
	误差: 0.149 179%	误差: 0.235 114%	误差: 0.201 467%
本文			
	误差: 0.074 446%	误差: 0.097 925%	误差: 0.132 27%



3.4　Mipmap 多级映射技术生成 VPLs 的分析

为了验证基于 Mipmap 多级映射技术生成 VPLs 的高效性，统计了不同场景在该阶段的时间开销，如表 6 所示. 可以看出，基于 Mipmap 多级映射技术能够快速基于 IDDM 生成 VPLs，整个阶段不超过 0.4 ms.

表 6　生成 VPLs 的时间开销　　ms

阶段	金属戒指	康奈尔盒	斯坦福兔子
生成多级 Mipmap	0.317	0.318	0.320
初始化样本	0.005	0.005	0.005
线性调整	0.072	0.072	0.072
合计	0.394	0.395	0.397

4　结　语

本文针对间接光泽反射对 VPLs 的分布有更高要求的问题，提出一种重要性驱动的实时间接光泽反射算法. 该算法基于历史帧中 VPLs 的光照贡献，在光源空间中生成 IDDM，指导 VPLs 在场景中的分布；同时，基于 IDDM，使用 Mipmap 技术快速生成 VPLs；提出自适应的时域复用算法，以减轻场景闪烁的情况；提出基于 Tile 的光照贡献估算方法，以加速计算. 本文仍需在 2 个方面进行改进:

(1) 实现多次反射的间接光照效果. 当前算法只支持一次反射的间接光照，为了渲染更加真实的全局光照效果，多次反射的间接光照效果是未来工作的重点.

(2) 添加更多光源类型. 对于面光源等具有空间体积的光源，光线的起始位置处于其面上的任意区域，使得 IDDM 难以较好地表达光照贡献的分布. 因此，改进 IDDM 使其支持更多的光源类型，将是未来工作的研究方向之一.

参考文献(References):

[1] Tokuyoshi Y, Harada T. Stochastic light culling for VPLs on GGX microsurfaces[J]. Computer Graphics Forum, 2017, 36(4): 55-63

[2] Keller A. Instant radiosity[C] //Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques. New York: ACM Press, 1997: 49-56

[3] Dachsbacher C, Stamminger M. Reflective shadow maps[C] //Proceedings of the Symposium on Interactive 3D Graphics and Games. New York: ACM Press, 2005: 203-231

[4] Lensing P. LightSkin: globale echtzeitbeleuchtung für virtual und augmented reality[D]. Ilmenau: Technische Universität Ilmenau, 2014

[5] Bischoff D, Schwandt T, Broll W. A real-time global illumination approach for high resolution reflective shadow maps in open world scenes[C] //Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications. Setúbal: Science and Technology Publications, 2017: 116-126

[6] Sriwasansak J, Gruson A, Hachisuka T. Efficient energy- compensated VPLs using photon splatting[C] //Procee- dings of the ACM on Computer Graphics and Interactive Techniques. New York: ACM Press, 2018: Article No.16

[7] Schwandt T, Kunert C, Broll W. Glossy reflections for mixed reality environments on mobile devices[C] //Proceedings of International Conference on Cyberworlds. Los Alamitos: IEEE Computer Society Press, 2018: 138-143

[8] Laine S, Saransaari H, Kontkanen J, *et al*. Incremental instant radiosity for real-time indirect illumination[C] //Proceedings of the 18th Eurographics Conference on Rendering Techniques. Aire-la-Ville: Eurographics Association Press, 2007: 277-286

[9] Veach E, Guibas L J. Metropolis light transport[C] //Proceed-

ings of the 24th Annual Conference on Computer Graphics and Interactive Techniques. New York: ACM Press, 1997: 65-76

[10] Simon F, Hanika J, Dachsbacher C. Rich-VPLs for improving the versatility of many-light methods[J]. Computer Graphics Forum, 2015, 34(2): 575-584

[11] Georgiev I, Slusallek P. Simple and robust iterative importance sampling of virtual point lights[C] //Proceedings of Eurographics (Short Papers). Aire-la-Ville: Eurographics Association Press, 2010: 57-60

[12] Ritschel T, Eisemann E, Ha I, *et al*. Making imperfect shadow maps view-adaptive: high-quality global illumination in large dynamic scenes[J]. Computer Graphics Forum, 2011, 30(8): 2258-2269

[13] Davidovič T, Křivánek J, Hašan M, *et al*. Combining global and local virtual lights for detailed glossy illumination[J]. ACM Transactions on Graphics, 2010, 29(6): Article No.143

[14] Hedman P, Karras T, Lehtinen J. Sequential Monte Carlo instant radiosity[C] //Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games. New York: ACM Press, 2016: 121-128

[15] Babahenini D, Gruson A, Babahenini M C, *et al*. Efficient inverse transform methods for VPL selection in global illumination[J]. Multimedia Tools and Applications, 2018, 77(11): 13571-13595

[16] Jin S H, Wang R, Zheng W T, *et al*. A virtual point light generation method in close-range area[OL]. [2022-06-08]. https://doi.org/10.48550/arXiv.2203.11484

[17] Walter B, Fernandez S, Arbree A, *et al*. Lightcuts: a scalable approach to illumination[J]. ACM Transactions on Graphics, 2005, 24(3): 1098-1107

[18] Walter B, Arbree A, Bala K, *et al*. Multidimensional lightcuts[J]. ACM Transactions on Graphics, 2006, 25(3): 1081-1088

[19] Huo Y C, Jin S H, Liu T, *et al*. Spherical Gaussian‐based lightcuts for glossy interreflections[J]. Computer Graphics Forum, 2020, 39(6): 192-203

[20] Harada T. A 2.5D culling for forward+[C] //Proceedings of SIGGRAPH Asia 2012 Technical Briefs. New York: ACM Press, 2012: Article No.18

[21] O'Donnell Y, Chajdas M G. Tiled light trees[C] //Proceedings of the 21st ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games. New York: ACM Press, 2017: Article No.1

[22] Tokuyoshi Y, Harada T. Stochastic light culling[J]. Journal of Computer Graphics Techniques, 2016, 5(1): 35-60

[23] Yuan Qing, Lou Lizhi, Chen Weixian. Applying weighted total least-squares to the plane point cloud fitting of terrestrial laser scanning[J]. Bulletin of Surveying and Mapping, 2011(3): 1-3(in Chinese)
(袁庆, 楼立志, 陈玮娴. 基于加权总体最小二乘的平面点云拟合方法[J]. 测绘通报, 2011(3): 1-3)

[24] Deledalle C A, Denis L, Tabti S, *et al*. Closed-form expressions of the eigen decomposition of 2×2 and 3×3 Hermitian matrices[OL]. [2022-06-08]. https://hal.archives-ouvertes.fr/hal-01501221