学号	姓名	论文规范性（10）	问题分析与调研（30）	方案创新性（20）	实验结果分析与讨论（40）	结课论文总成绩（100）
21301034	付家齐	9	28	19	39	95

北京交通大学

计算机图形学期末课程论文

基于 RSM 算法的全局光照实现与分析

学　　院：软件学院

专　　业：软件工程

学生姓名：付家齐

学　　号：21301034

北京交通大学

2024 年 6 月

中文摘要

摘要：全局光照（Global Illumination）是计算机图形学中的一个重要研究方向，它模拟了光在场景中多次反射和折射的复杂光照现象，从而使得渲染效果更加的真实和自然。现代图形学技术已经基本解决了间接光照的问题，尤其是 Lumen 算法。本文主要总结了当前全局光照的相关算法及其特点，并且基于 RSM（Reflective Shadow Map）算法实现了在自研游戏引擎中的全局关照效果。同时本文对比了自研游戏引擎中，不同环境下，不同 RSM 参数的全局光照效果。实验结果表明，在 RSM 算法中只有当直接光线不强烈的时候，全局光照的效果才比较明显，同时 RSM 对计算的开销较大，性能较差。

关键词：全局光照；实时渲染；RSM 算法

目　　录

中文摘要 I
目　　录 II
1　引言 1
1.1　研究背景 1
1.2　相关工作 1
1.2.1　蒙特卡洛光线追踪算法 1
1.2.2　光传播体积算法 1
1.2.3　Lumen 算法 2
1.3　本文工作 2
2　基于 RSM 算法的动态全局光照设计与实现 3
2.1　基本原理 3
2.2　实现细节 4
3　实验 5
3.1　不同场景下环境光对比实验 5
3.2　性能对比实验 6
3.3　实验发现 7
3.4　实验总结 7
4　总结 8
参考文献 9

1 引言

1.1 研究背景

全局光照（Global Illumination）[1]在计算机图形学中扮演着至关重要的角色，通过模拟光在场景中的多次反射和折射，显著提升了渲染的真实性和自然性。传统的光照模型主要处理直接光照，忽略了间接光照的影响，导致渲染效果缺乏真实感。随着计算能力和图形算法的不断进步，现代图形学技术逐步解决了间接光照的问题，使得全局光照的实时实现成为可能。

1.2 相关工作

线代图形学中关于全局光照技术的研究已经持续已久，并基本解决了这个问题，在这个过程中诞生了许多相关的算法。本章节根据《GAMES104》[2]和《Real-Time Rendering, Fourth Edition》[3]的内容简述关于全局光照相关算法。

1.2.1 蒙特卡洛光线追踪算法

蒙特卡洛光线追踪（Monte Carlo Ray Tracing）[4]是一种离线计算全局光照的算法。其主要思想是模拟真实的光路和成像过程，从相机出发向屏幕上的每一个像素放出射线，这些射线如果经过反弹射中了光源，那么我们就可以根据这条路径计算出对应的全局光照，不过通过每增加一层反弹，它的计算复杂度会进行指数级的扩张。这种方法最大的问题是采样，因为采样是随机的，所以没有办法保证相邻像素能够产生同样的结果，因此会在屏幕上看到很多的噪点。

1.2.2 光传播体积算法

光传播体积（Light Propagation Volumes, LPV）[5]算法基于这样的一个假设：光照辐射率在三维空间中沿直线传播且传播过程中辐射率保持不变。因此，LPV 的核心思路就是将整个场景的三维空间划分成一个均匀的体素网格，然后在这些体素格子之间传播光照的辐射率，最后每个着色点根据其三维位置找到相应的体素格子，从中取传播得到的

辐射率进行间接光照的计算。

1.2.3 Lumen 算法

Lumen[6]算法是当今图形学关于全局光照技术的集大成者，由 Epic Games 在其虚幻引擎 5 中引入。其综合使用了多种技术的结合体，而非单一技术的运用。Lumen 最核心的思想是以下三点：1. 在不用硬件 Ray Trace 的前提下，进行快速的 Ray Trace。2. 尽可能的在 sample 里做的优化。3. 放置的 Probe 尽可能贴着真实物体表面，使它的精度足够的高。

1.3 本文工作

反射阴影贴图（Reflective Shadow Map, RSM）作为一种有效的实时全局光照方法，通过捕捉光的反射信息，能够在复杂场景中实现逼真的光照效果。RSM 方法的核心在于记录光源发出的光在场景中的反射和漫射信息，从而计算出间接光照。

本文的研究旨在总结当前全局光照的相关算法及其特点，并基于 RSM 算法实现自研游戏引擎中的动态全局光照效果。通过对不同环境下、不同 RSM 参数设置的全局光照效果进行对比和分析，探讨 RSM 方法在实际应用中的性能表现和优化空间。希望通过这项研究，为实时渲染领域提供有价值的参考，进一步推动全局光照技术的发展和应用。

2 基于 RSM 算法的动态全局光照设计与实现

反射阴影贴图（Reflective Shadow Map, RSM）[7]是一种用于交互式渲染的间接光照算法。RSM 的主要思想是对标准阴影贴图（Shadow Map）的一种拓展，其将每个像素都视为一个间接光源，他解决的核心问题是怎么把光注入到场景中去。

2.1 基本原理

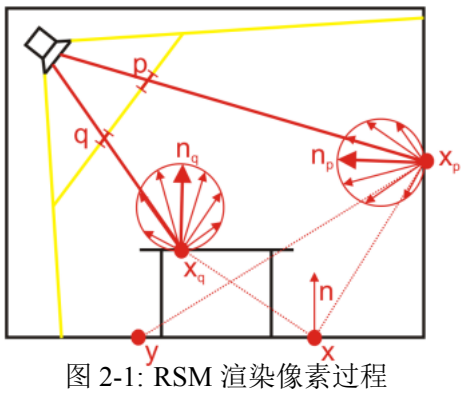

图 2-1: RSM 渲染像素过程

RSM（Reflective Shadow Maps）算法是基于 Shadow Map[8]机制的，同样是由两个阶段组成。第一个阶段中，RSM 算法记录哪些场景面元被直接光照照亮，将这些被直接光照照亮的面元几何信息和着色信息存储记录下来；在第二个阶段中，根据前面得到的直接光照照亮面元的信息，计算这些面元对其他场景物体的光照。

借用光子映射（Photon Mapping）[9]的思想，从光源的位置，可以能看到所有被它直接照亮的表面。所有被第一次被光源照亮的表面将被渲染到 Map 中，一个像素不会多，一个像素也不会少。相比与 Shadow Map 存储深度值，RSM 还会存储其他的一些信息。RSM 在每个像素 p 中存储深度值 d_p，世界空间位置 x_p，法线 n_p 以及可见表面点的反射辐射通量 Φ_p。为了便于存储和计算反射物到任意方向的辐射率，RSM 算法做了一个假设：反射物的材质均为漫反射材质，其任意方向的反射辐射率是一个某个固定的值。

在阶段二中，我们将要渲染每个像素的光照值。如图2-1中所示，假设我们要渲染 x 点的光照值，x 点因为被桌子挡住，光源无法直接照亮这个点，因此 x 点并没有直接光照。根据阶段一中得到的 Reflective Shadow Map，可以得知光源会射到 x_p 点，x_p 点会沿着它的法线方向进行散射，所以可以计算 x_p 点到 x 点的反射辐射率贡献，计算公式为：

$$E_p(x,n) = \Phi_p \frac{\max\{0, \langle n_p | x - x_p \rangle\} \max\{0, \langle n | x_p - x \rangle\}}{\|x - x_p\|^4}. \tag{2-1}$$

由此 x 点的间接辐照度可以通过所有像素光源的照明之和来近似：

$$E(x,n) = \sum_{\text{pixels } p} E_p(x,n) \tag{2-2}$$

最后一个问题是如何解决上式中的的求和范围。最简单直接的方法就是将 RSM 中的每个像素都当成是一个小光源，把 RSM 上的每一个点都进行一次渲染，但这个方法有个问题就是过于粗暴，计算复杂度太高。RSM 又做了一个非常大胆的假设：空间上相邻的点投影到 Shadow Map 上也是邻近的。因此解决方式就是如果某个方向上采样点距离比较远，我们采样的密度低一些，如果比较近，我们采样的密度会高一些。

2.2 实现细节

具体 RSM 代码实现在 https://github.com/YXHXianYu/BJTU-Game-Engine 仓库中 feat-GI 分支下的 shader/glsl/shading.frag 的 line 139-175。以下展示的是伪代码。

Algorithm 1 RSM 光照计算

for each i, j in $[0.0, 1.0]$ with steps di, dj **do**
  $rsm\_position \leftarrow \text{texture}(u\_rsm\_position\_texture, (i, j))$
  $attenuation \leftarrow 1/(1 + k \times \text{distance}^2(rsm\_position, frag\_pos))$
  if $attenuation < \varepsilon$ **then**
    continue
  end if
  $rsm\_normal \leftarrow \text{texture}(u\_rsm\_normal\_texture, (i, j))$
  if $\text{dot}(rsm\_normal, normal) \geq \theta$ **then**
    continue
  end if
  $rsm\_light\_dir \leftarrow \text{normalize}(rsm\_position - frag\_pos)$
  if $\text{dot}(rsm\_normal, -rsm\_light\_dir) \leq \theta$ or $\text{dot}(normal, rsm\_light\_dir) \leq \theta$ **then**
    continue
  end if
  $rsm\_color \leftarrow \text{texture}(u\_rsm\_color\_texture, (i, j)) \times light\_color \times attenuation$
  $rsm\_color \leftarrow rsm\_color \times \max(\text{dot}(rsm\_normal, light\_dir), 0.0)$
  $\text{calc\_light}(frag\_pos, normal, rsm\_light\_dir, rsm\_color, diffuse\_light, specular\_light)$
  $rsm\_ambient \leftarrow rsm\_ambient + diffuse\_light$
end for
$ambient \leftarrow ambient + rsm\_ambient \times RSM\_STRENGTH$

3 实验

本章在渲染器 BJTU Game Engine 中进行了不同场景，不同 RSM 参数下的全局光照效果对比。RSM 参数包括强度、角度阈值和距离衰减程度。

3.1 不同场景下环境光对比实验

在本节实验中，选取了开阔地与室内分别作为实验场景，对比有无环境光的效果。以下是实验结果。

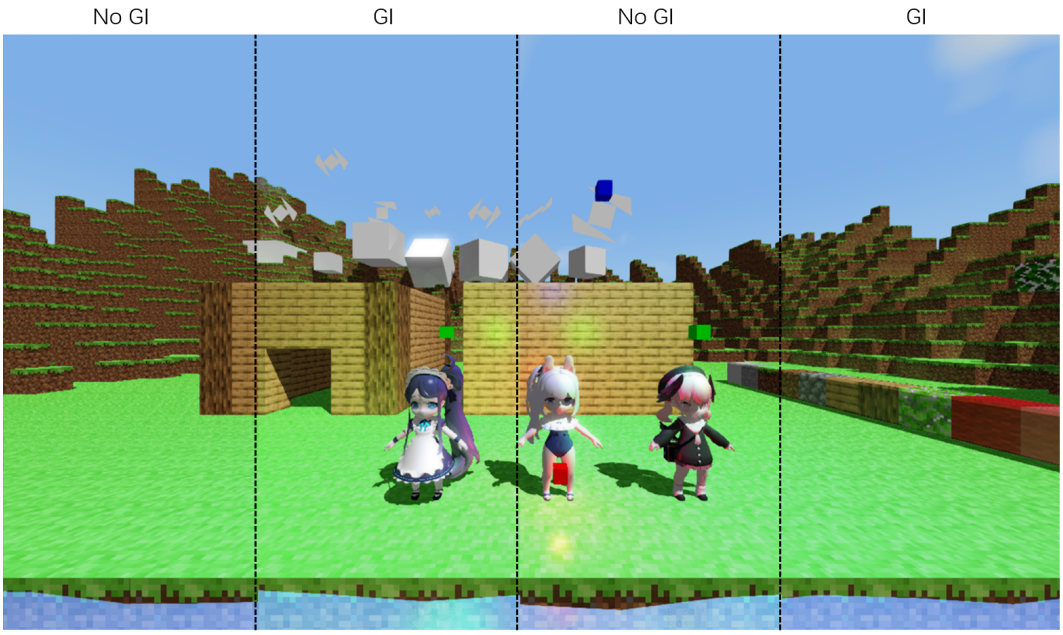

图 3-1: 室外有无全局光照对比

在图3-1中，从左往右依次为无全局光照、有全局光照、无全局光照、有全局光照。通过对比左侧半张图中小屋墙体部分的颜色，可以看到有全局光照一侧的墙体颜色呈现出绿色，为草地的颜色反射效果。这表明全局光照有效地捕捉到了草地的间接光照，并反射到墙体上。对比右侧半张图，可以发现在远处山体部分的阴影处，全局光照的效果更加明显。无全局光照的一侧阴影过渡较为生硬，而有全局光照的一侧阴影过渡更加自然，显示出更多的细节和真实感。

在图3-2中，可以看出无全局光照的部分显示出墙壁和地板的光照效果较为单一，光照分布不均匀，缺乏细节。而有全局光照的部分则表现出光照更加均匀，光线过渡自然，尤其是地板上的红光反射到墙壁上，使墙体呈现出淡淡的红色，增加了场景的真实感。

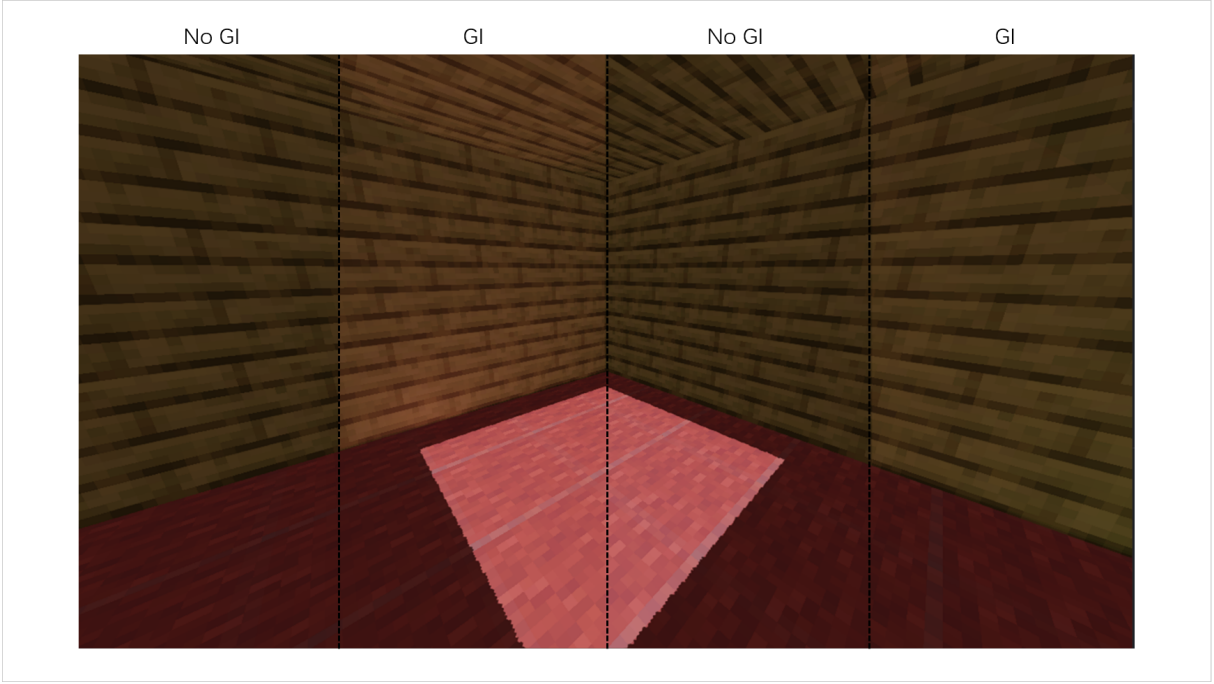

图 3-2: 室内有无全局光照对比

通过这些对比，可以看出全局光照在提高光照均匀性和真实感方面的显著优势。

通过对比室内室外两张效果图，我们可以看出在缺乏直接光照的室内场景体现了更好的全局光照效果。室外效果中全局光照较为明显部分同样是山的阴影部分。由此可以得出当直接光线不强烈的时候，全局光照才比较明显。

3.2 性能对比实验

表 3-1: 实验机器基本参数

操作系统	CPU	GPU	内存	显存
Windows 10	Intel i7-11800H	NVIDIA RTX 3070	32GB	8GB

本节实验中我对比了在室外和室内两个场景下，开启与关闭全局光照的 FPS 数。实验选用的电脑配置如表3-1所示。FPS 数取在环境中静止一分钟的 FPS 平均数。实验结果如表3-2所示，可以看出所实现的 RSM 算法性能还是有所缺陷的，在室内和室外测试中 FPS 都有着显著的下降。这表明当前 RSM 算法需要处理大量的间接光照计算，尤其是在复杂场景中，对计算资源的需求较高，影响了渲染性能。

表 3-2: 室外和室内场景下开启与关闭全局光照的 FPS 对比

场景	关闭 GI (FPS)	开启 GI (FPS)
室外	64	32
室内	59	28

3.3 实验发现

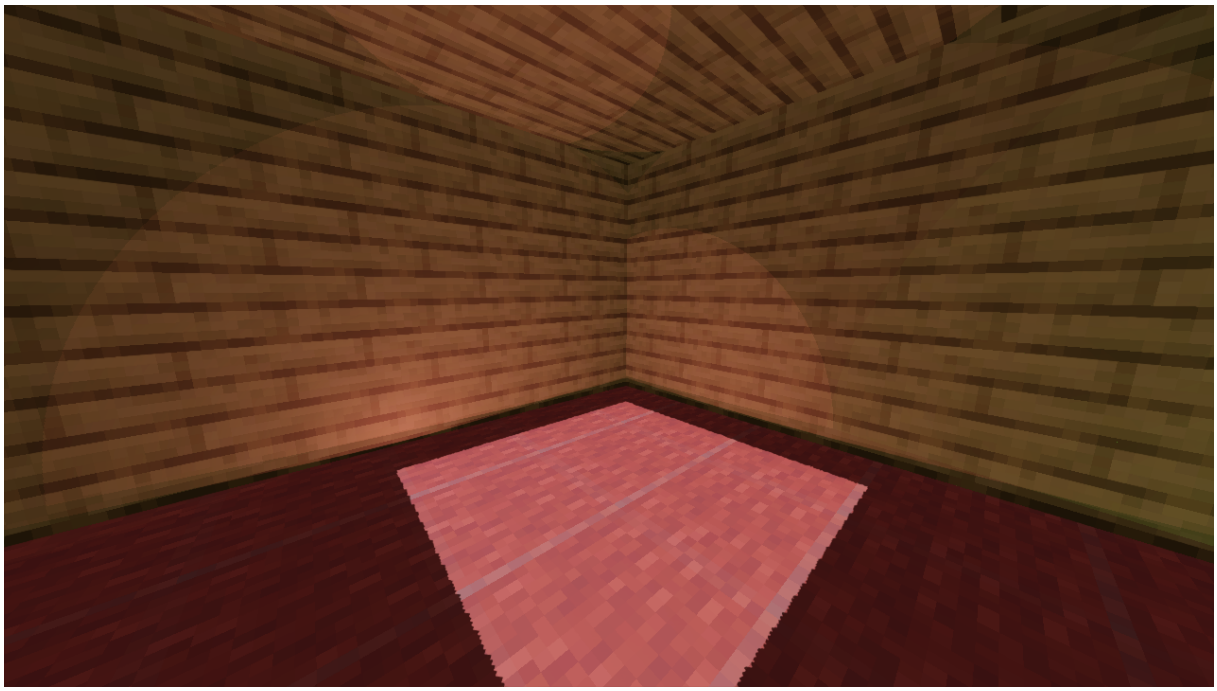

图 3-3: GI 与环境产生明显分割

本节实验讲述了在实验过程中的发现。在实验过程中，为了测试全局关照的效果，对于三个参数进行了调参，上述实验结果均是在强度为 0.3，角度阈值为 0.01，距离衰减程度为 1.25 的条件下产生的结果。但是在测试过程中发现了当强度设定为 0.6，角度阈值设定为 0.05 时，全局光照与环境产生了明显的分割线，如图3-3所示。

3.4 实验总结

本节实验中尝试了 RSM 算法的效果。相比于蒙特卡洛光线追踪算法，RSM 算法可以在线执行，可以应用到游戏引擎当中，实现动态全局光照。实验结果显示，开启全局光照后的效果显著提升了场景的真实感和视觉效果，特别是在光照均匀性和阴影过渡方面。然而，我也发现了 RSM 算法的局限性，RSM 算法的计算复杂度较高，尤其在复杂场景中，导致 FPS 显著下降。

4 总结

本文研究了基于反射阴影贴图（RSM）算法的全局光照相关工作调研，RSM 算法实现与分析。相比于传统的蒙特卡洛光线追踪算法，RSM 算法具有实时执行的优势，适用于动态场景中的全局光照渲染。通过在自研游戏引擎中实现 RSM 算法，并在不同场景下进行测试，实验结果表明，RSM 算法显著提升了场景的光照效果，特别是在光照均匀性和阴影过渡方面。但是也发现了 RSM 算法的局限性，RSM 算法计算复杂度较高，同时也只考虑了光线单次反弹，没有考虑间接光照的遮挡。但是毫无疑问 RSM 是一个非常具有启发性的算法，为后来全局光照领域的发展做出了巨大贡献。

参考文献

[1] ARVO J. Transfer equations in global illumination[J]. Global Illumination, SIGGRAPH‘93 Course Notes, 1993, 2: 6.

[2] GAMES-Webinar. GAMES104-现代游戏引擎入门必修课[Z]. 2022.

[3] AKENINE-MOELLER T, HAINES E, HOFFMAN N. Real-Time Rendering, Fourth Edition[M]. 2018.

[4] JENSEN H W, ARVO J, DUTRE P, et al. Monte Carlo ray tracing[C]//ACM SIGGRAPH: vol. 5. 2003: 340769537.

[5] KAPLANYAN A, DACHSBACHER C. Cascaded light propagation volumes for real-time indirect illumination[C]//Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games. 2010: 99-107.

[6] VENTER H, OGTEROP W. Unreal Engine 5 Character Creation, Animation, and Cinematics: Create custom 3D assets and bring them to life in Unreal Engine 5 using MetaHuman, Lumen, and Nanite [M]. Packt Publishing Ltd, 2022.

[7] DACHSBACHER C, STAMMINGER M. Reflective shadow maps[C]//I3D ’05: Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games. Washington, District of Columbia: Association for Computing Machinery, 2005: 203-231. DOI: 10.1145/1053427.1053460.

[8] STAMMINGER M, DRETTAKIS G. Perspective shadow maps[C]//Proceedings of the 29th annual conference on Computer graphics and interactive techniques. 2002: 557-562.

[9] JENSEN H W. Global illumination using photon maps[C]//Rendering Techniques’96: Proceedings of the Eurographics Workshop in Porto, Portugal, June 17–19, 1996 7. 1996: 21-30.