

北京交通大学

计算机图形学结课论文

基于光线追踪的渲染研究

A study of ray-tracing based rendering

学 院：	软件学院
专 业：	软件工程
学生姓名：	余旺
学 号：	22301056
指导教师：	吴雨婷

北京交通大学

2025 年 5 月

中文摘要

摘要：光线追踪技术（Ray Tracing）因其能够模拟真实光线的物理行为,如反射、折射、全局光照等，已成为影视、游戏和工业设计领域的高保真图像渲染核心方案。近年来，因 GPU 硬件加速架构的普及和相关技术飞速发展，受到了越来越多专家学者的关注与研究。为了深入对光线追踪技术的研究，本文以典型三维场景：红色漫反射球体、红黄棋盘格地面等为载体，系统性探索光线追踪模拟光线的反射折射效果。实验部分给出了不同视角下光线追踪渲染的水晶球图像，实验结果表明，光线追踪技术显著的提升了光照效果和真实感，为实时渲染提供可验证的技术参考。

关键词：光线追踪；图像渲染；反射折射；真实感

ABSTRACT

ABSTRACT: Ray Tracing has become a core solution for high-fidelity image rendering in film and television, games and industrial design because it can simulate the physical behaviour of real light, such as reflection, refraction, global illumination, etc. In recent years, due to the popularity of GPU hardware acceleration architecture and the rapid development of related technologies, it has received more and more attention and research from experts and scholars. In order to deepen the research on ray tracing technology, this paper takes typical 3D scenes : red diffuse reflective sphere, red and yellow checkerboard ground, etc. as the carrier, and systematically explores ray tracing to simulate the reflective and refractive effects of light. The experimental part gives the crystal ball images rendered by ray tracing under different viewpoints, and the experimental results show that ray tracing technology significantly improves the lighting effect and realism, and provides a verifiable technical reference for real-time rendering.

KEYWORDS: Ray tracing; Image rendering; Reflection refraction; Realism

目 录

中文摘要	I
ABSTRACT	II
目 录	III
1 引言	1
2 相关工作	2
3 方法描述	3
3.1 概述	3
3.2 光照	3
3.3 几何求交	4
3.3.1 球体求交	4
3.3.2 三角形求交	5
3.4 光线追踪算法	5
3.4.1 光线投射 (Ray Casting)	6
3.4.2 递归光线追踪 (Whitted-Style Ray Tracing)	6
4 实验	7
4.1 实验设置	7
4.1.1 实验环境与工具	7
4.1.2 场景与参数选择	7
4.2 实验结果	8
5 结论与展望	11
参考文献	12

1 引言

光线追踪 (Ray Tracing)^[1] 作为计算机图形学中最重要渲染技术之一, 通过模拟光线在虚拟场景中的物理传播行为, 能够生成具有高度真实感的图像效果。这项技术通过追踪从摄像机出发的每一条光线在场景中的传播路径, 精确计算光线与物体表面交互时产生的反射、折射、散射等现象, 从而实现对真实光照效果的物理模拟。

光线追踪的概念最早可以追溯到 16 世纪文艺复兴时期, 艺术家们通过”暗箱”技术研究光线传播规律。现代计算机图形学中的光线追踪算法则起源于 1968 年 Arthur Appel 提出的”光线投射”方法^[2]。1980 年, Turner Whitted 发表了开创性论文”An Improved Illumination Model for Shaded Display”, 首次提出了递归光线追踪算法, 奠定了现代光线追踪的理论基础^[1], 这一突破不仅为电影特效带来了革命性变化, 更为整个计算机图形学领域树立了新的技术标杆。

如今, 随着 GPU 硬件加速架构的出现, 光线追踪技术正从离线渲染领域走向实时应用^[3]。游戏引擎开始支持实时光线追踪效果, 建筑设计可视化能够即时呈现真实的光照效果, 甚至 AR/VR 设备也在探索如何利用这项技术提升沉浸感。这种转变不仅代表着计算能力的飞跃, 更反映了人们对真实感的追求。

本文作为《计算机图形学》的期末课程论文, 将从一束基础光线出发, 观察它如何与红色漫反射球体互动, 如何在透明水晶球内部折射, 又如何在棋盘格地面上投下阴影。通过这个过程, 我们不仅能够理解光线追踪的基本原理, 更能体会到这项技术如何通过精确的物理模拟, 实现令人惊叹的真实感。

2 相关工作

有关于光线追踪的理论研究与实践历史悠久，并且也有许多的相关理论研究与相关技术实现。1979 年 Whitted^[1]利用光线的可逆性，提出了光线追踪算法。这个算法模拟真实的光照效果，在场景中的光线与物体相交后会因反射、折射和阴影等产生二次光线，二次光线再次到场景中进行相交测试。如此进行迭代计算，直到光线的反弹次数达到预期设置的值。此后，Cook 等人提出的分布式光线追踪^[4]和 Kajiya 提出的渲染方程^[5]进一步推动了这项技术的发展。

光线追踪的核心在于高效计算光线与物体的最近交点。早期采用空间网格（grid）划分方法^[6]，但存在划分不均、计算冗余等问题。后续研究提出了基于分层空间结构的加速方法，其思路是将场景中的图元划分到不同的分层空间结构中，在这种结构下，可以快速剔除那些不相关的空间，进而识别出与光线最接近的图元。而目前主流的加速结构包括 kd-tree^[7]和 BVH（包围体层次结构）^[8]，他们均采用轴对齐包围盒（AABB）来进行场景划分，从而简化运算。虽然 kd-tree 曾被认为是光线追踪的最佳选择^[9]，但由于其在相交检测等方面的局限，越来越多的研究倾向于使用 BVH。相对比与 kd-tree，BVH 内存可控、鲁棒性强、遍历高效^[10]，已成为离线和实时光线追踪的标准加速结构。

近年来，随着 GPU 并行计算架构的快速发展，光线追踪技术迎来了新的突破。NVIDIA 推出的 RTX 系列显卡首次在硬件层面实现了实时光线追踪加速，使得复杂场景的交互式渲染成为可能。在算法层面，Barrel 等人^[11]提出的混合渲染管线结合了传统光栅化与光线追踪的优势，显著提升了渲染效率。同时，深度学习技术的引入也为光线追踪带来了新的可能性，如 Zhu 等人^[12]总结的基于神经网络的降噪方法，能够在低计算开销下保持高质量的渲染效果。这些技术进步极大地拓展了光线追踪在游戏开发、影视制作和虚拟现实等领域的应用前景。

3 方法描述

3.1 概述

本文实现了一个基于 Whitted 风格光线追踪的三维渲染场景，该场景通过模拟光线与物体的物理交互来生成图像。首先从虚拟相机发射主光线穿过每个像素平面，利用几何求交算法检测光线与场景中物体如球体和三角形网格等的最近交点，并根据交点处的材质属性递归地追踪反射、折射等次级光线。通过定向光、点光源和环境光等多光源的 Phong 光照模型计算局部着色，并结合背景色形成最终图像。

3.2 光照

本文中使用了 Phong 光照模型，具体的公式如下3-1所示。Phong 光照模型是一种经典的局部光照模型，用于模拟物体表面的漫反射、镜面反射和环境光效果。该模型将最终的光照颜色分解为三个部分：环境光（Ambient）、漫反射（Diffuse）和镜面反射（Specular）。

$$I = I_{\text{ambient}} + I_{\text{diffuse}} + I_{\text{specular}} \quad (3-1)$$

环境光模拟场景中间接光照的均匀照明效果，它不受物体表面朝向或观察方向的影响。它确保物体即使在无直接光照的情况下仍能呈现基础亮度，避免完全黑暗的区域。环境光的强度由环境光反射系数 k_a 和环境光强度 I_a 共同决定。

$$I_{\text{ambient}} = k_a \cdot I_a \quad (3-2)$$

其中， k_a 为环境光反射系数和 I_a 为环境光强度。

漫反射光模拟粗糙表面对光线的均匀散射现象，遵循余弦定律：入射光强度与光线方向和表面法线夹角的余弦成正比。其特点是光照强度随入射角增大而减弱，但反射方向与观察者位置无关，因此表面呈现均匀的柔和明暗变化。漫反射光通过点积计算光源方向 \mathbf{l} 与表面法线 \mathbf{n} 的夹角，是表现物体体积感的核心分量。

$$I_{\text{diffuse}} = k_d (I/r^2) \max(0, \mathbf{n} \cdot \mathbf{l}) \quad (3-3)$$

其中， k_d 表示漫反射系数， I/r^2 表示光到当前点的强度， \mathbf{l} 表示光源方向， \mathbf{n} 表示法线。

当光线照射到光滑表面时，会沿特定方向集中反射，形成高光（Specular Highlight）。镜面反射通过半程向量 \mathbf{h} 与法线 \mathbf{n} 的夹角，量化反射光的强度分布。

$$I_{\text{ambient}} = k_s(I/r^2) \max(0, \mathbf{n} \cdot \mathbf{h})^p \quad (3-4)$$

其中， k_s 表示镜面反射系数，决定材质反射光的能力， I/r^2 表示光源强度 I 随距离 r 的平方反比衰减， \mathbf{h} 表示半程向量， \mathbf{n} 表示法线， p 控制高光。

3.3 几何求交

3.3.1 球体求交

对于球体这类基本几何形状，我们采用解析求交法，通过建立光线方程3-5与球体方程的联立方程组3-6，求解二次方程3-7来快速确定交点位置。最终通过计算判别式 $\delta = b^2 - 4ac$ 来判断是否存在实数解，并选取最小的正实数解作为有效交点。这种方法的计算复杂度为常数时间 $O(1)$ ，特别适合处理场景中的简单几何体。

$$\mathbf{R}(t) = \mathbf{O} + t\mathbf{d} \quad (3-5)$$

其中 \mathbf{O} 为光线起点， \mathbf{d} 为方向向量。与球心 \mathbf{C} 、半径 r 的球体相交时，需满足：

$$\|\mathbf{R}(t) - \mathbf{C}\|^2 = r^2 \quad (3-6)$$

展开后得到二次方程：

$$at^2 + bt + c = 0 \quad (3-7)$$

其中， $a = \mathbf{d} \cdot \mathbf{d}$ ， $b = 2\mathbf{d} \cdot (\mathbf{O} - \mathbf{C})$ ， $c = (\mathbf{O} - \mathbf{C}) \cdot (\mathbf{O} - \mathbf{C}) - r^2$ 。

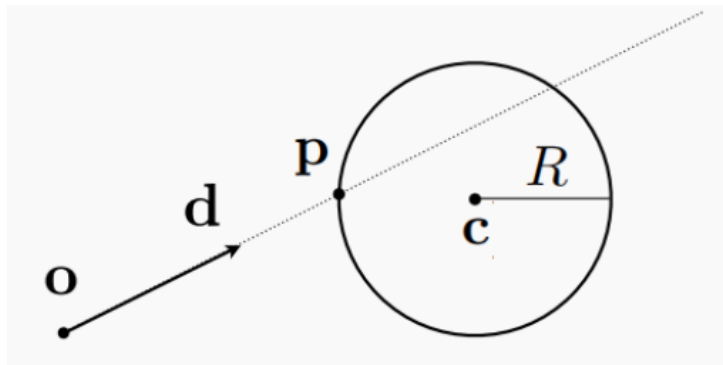


图 3-1 光线球体求交

3.3.2 三角形求交

对于更复杂的三角网格模型，我们实现了 Möller-Trumbore 算法，该算法利用重心坐标将交点表示为由三角形顶点线性组合的形式，通过求解一个 3×3 线性方程组来精确计算光线与三角形的交点参数。这种方法避免了传统平面求交法需要预计算平面方程的额外开销，在保持计算精度的同时显著提升了求交效率。

算法通过重心坐标直接求解，我们首先表示交点：

$$\mathbf{P} = \mathbf{v}_0 + u(\mathbf{v}_1 - \mathbf{v}_0) + v(\mathbf{v}_2 - \mathbf{v}_0) \quad (3-8)$$

其中 $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2$ 为三角形顶点， u, v 为重心坐标（需满足 $u \geq 0, v \geq 0, u + v \leq 1$ ）。

接着将光线方程代入，整理为矩阵形式，并进行求解：

$$\begin{bmatrix} -\mathbf{d} & \mathbf{v}_1 - \mathbf{v}_0 & \mathbf{v}_2 - \mathbf{v}_0 \end{bmatrix} \begin{bmatrix} t \\ u \\ v \end{bmatrix} = \mathbf{O} - \mathbf{v}_0 \quad (3-9)$$

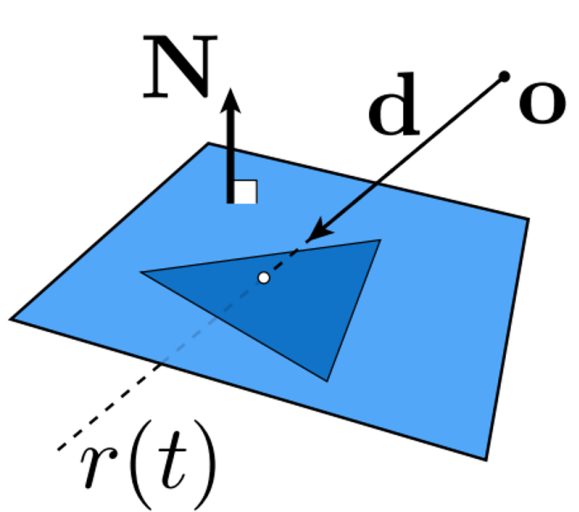


图 3-2 光线三角形求交

3.4 光线追踪算法

光线追踪听名字就知道，讨论的核心是光线，因此我们首先对光线做一些假设。

- 1 光线一定沿着直线传播
- 2 光线之间无法碰撞

3 光线路径可逆，即从 A 发出的到 B 的光线，一定也可以从 B 发出到 A（中途可发生反射和折射）

基于假设，我们可以阐述我们的光线追踪算法。

3.4.1 光线投射 (Ray Casting)

光线追踪的起点是从人眼或摄像机向虚拟成像平面的每个像素发射一条主光线 (Primary Ray)。这条光线从相机出发，穿过成像平面的像素点，进入三维场景中。在代码中，我们首先遍历所有物体，计算光线与每个物体的相交点。若存在多个交点，仅保留最近的交点，并将其他交点因被遮挡而忽略。接着，从交点向场景中的光源发射阴影光线 (Shadow Ray)，检测该点是否位于阴影中。若阴影光线在到达光源前与其他物体相交，则认为该点被遮挡，无法直接接受光源照射。最后，对于未被遮挡的交点，使用光照模型计算其颜色，生成初步的像素颜色。

3.4.2 递归光线追踪 (Whitted-Style Ray Tracing)

为模拟全局光照效果，我们对光线在物体表面的反射与折射行为进行递归追踪。首先，若交点位于镜面或光滑表面，我们根据表面法线方向计算反射光线，并向新方向发射次级光线 (Secondary Ray)。次级光线继续与场景物体相交，重复执行交点检测、阴影计算和着色流程。其颜色贡献按反射率加权累加到原始交点的最终颜色中。其次，若交点位于透明或半透明物体，我们根据材质折射率计算光线折射方向，发射次级光线进入物体内部。折射光线在物体另一侧穿出时可能再次发生方向偏折，继续追踪直至逃逸出场景或达到递归深度限制。最后，我设置了最大递归深度来限制光线反射或者折射次数，避免无限循环。

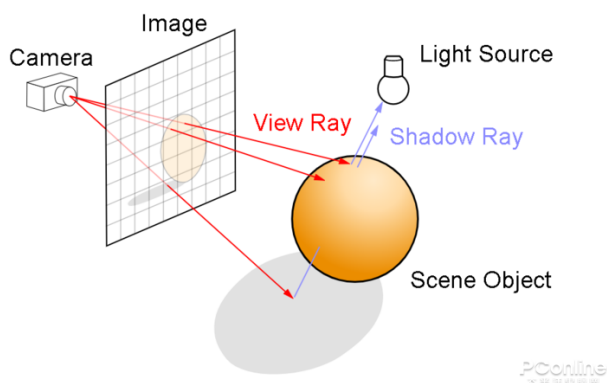


图 3-3 光线追踪算法示意图

4 实验

4.1 实验设置

4.1.1 实验环境与工具

本实验基于 Windows 11 操作系统搭建开发环境，采用 Clion 2024.1 作为集成开发工具，主要使用 C++ 编程语言实现核心算法。图形渲染部分依托 OpenGL 图形库实现三维场景的绘制与光照计算。

4.1.2 场景与参数选择

几何图形

- 1，红色漫反射球体：位置 (-1,0,-12)，半径 2，使用混合材质，漫反射系数 $K_d=0.8$ 主导表面着色, 高光指数为 25，镜面反射系数 $K_s=0.2$ ，基础色为朱红色。
- 2，玻璃球体：位置 (0.5,-0.5,-8)，半径 1.5，为折射率 1.5。
- 3，四边形地板：顶点坐标 (-5,-3,-6) 到 (-5,-3,-16) 构成的平面，使用红黄相间的棋盘格纹理

光照配置

我们采用三点布光法构建场景照明体系，各光源参数如表所示：

表 4-1 光源配置参数表				
光源类型	坐标 (x,y,z)	颜色 (R,G,B)	辐射度 (lx)	衰减模型
主光源	(-30,100,30)	(1.0,1.0,0.9)	1.2×10^5	平方反比衰减
辅助光	(40,60,-15)	(0.6,0.7,1.0)	3.6×10^4	线性衰减
背光	(0,50,-30)	(0.3,0.3,0.4)	1.1×10^4	恒定衰减

同时，采用蓝灰色的背景色。蓝灰色环境光为阴影注入自然冷调，中和主光源的同时，通过色彩对比增强物体轮廓的立体感与真实感。

4.2 实验结果

通过不同视角的切换，我们渲染上下左右远近，正视角共七张图片。为保持场景的简洁性，在实验过程中并未添加更多的玻璃球，漫反射球体等，这样有助于实验的探究。

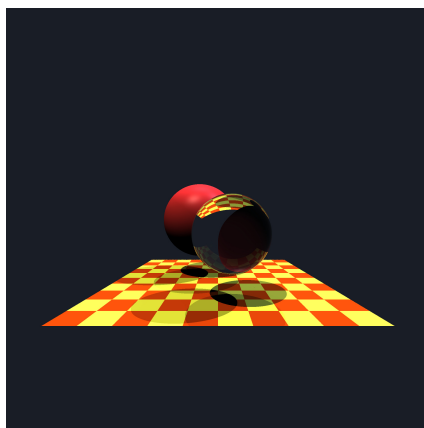
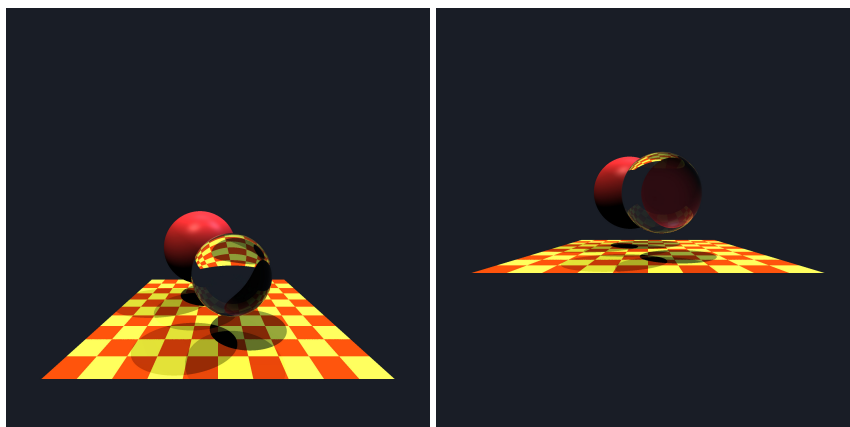


图 4-1 正视图

从正视图，图4-1可以看出，水晶球中心区域展现了典型的折射特性，后方红球通过球体时发生显著形变，棋盘格线条的弯曲轨迹体现了斯涅尔定律对光线路径偏折的模拟。球体顶部的高光区域位置严格遵循 Phong 模型中的镜面反射规律——当光源方向与观察方向重合时，半程向量与法线方向一致，导致反射光强度达到峰值。棋盘表面的阴影特征进一步验证了算法效能：透明球阴影边缘因全局光照的多次散射呈现柔和过渡，而红球阴影的边界则反映了直射光被物体完全遮挡时的几何特性，这证明光线-物体相交检测算法的正确性。此外，图像整体色彩过渡自然，表明递归算法不仅完整计算了直射光，还通过反射与折射路径有效捕捉了间接光照的叠加效应。

从水晶球顶部观察，如图4-2a，红黄棋盘更为突出，其弯曲轨迹的线条曲线区域明显增大，同时红球的形变程度也更加明显，充分展现了折射和反射的效果，真实准确。



(a) 上移视图

(b) 下移视图

图 4-2 上下视图

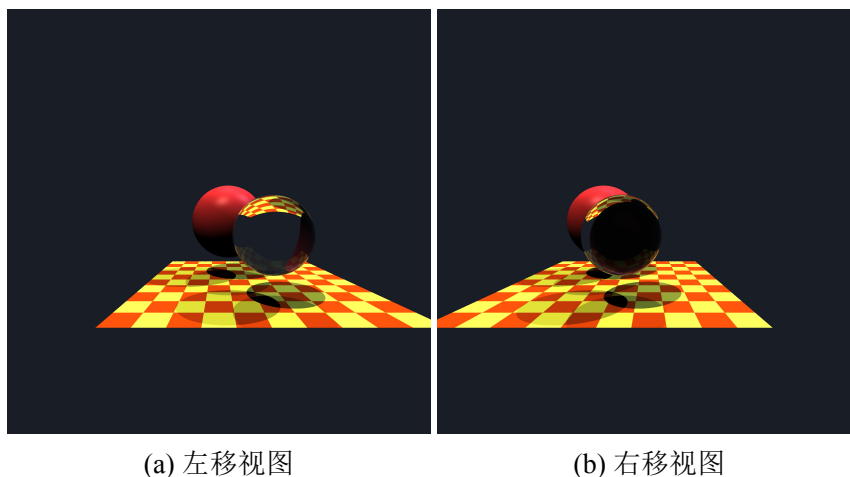


图 4-3 左右视图

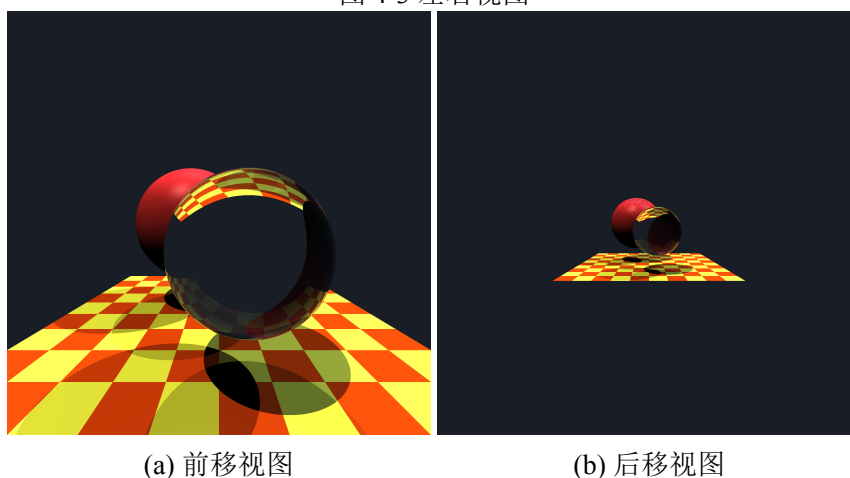


图 4-4 前后视图

从下方观察时,如图4-2b,同样的,红黄棋盘被压缩,其弯曲轨迹的线条曲线区域明显减小反之红球的形变程度减小,但区域更大,这充分展现摄像机视角不同下的反射折射效果。棋盘上的阴影因视角的下移,而变得扁平,呈椭圆状,呈现出因光线被球体自身遮挡而表现的真实效果。

当光线从左侧进入水晶球,折射路径向右侧偏折,形成向右延伸的复杂光带。左侧区域折射光斑亮度较高,符合的强度衰减规律。右移视图,光线从右侧进入,折射路径向左侧偏折,呈现镜像对称性。继续分析,左移视图高光区域集中在球体右侧表面,右移视图,高光转移至球体左侧表面,这很好的表现了 Phong 模型对视角依赖性。深入分析,左移视图环境光通过球体左侧表面折射后,在右侧背景形成渐变光晕,同样的右移视图,环境光从右侧进入,间接光照向左扩散,阴影分布与左视图镜像对称,验证了光线追踪在复杂遮挡模拟中的精度。

从前视图4-4a,光线从摄像机正前方垂直入射水晶球,高光区域集中在球体中心附近,这符合 Phong 模型的镜面反射特性,同时环境光多次反射叠加,全局光照效果显著。从后视图4-4b,光线从球体后方入射,折射路径向观察方向偏折。由于光线需穿透整个

球体，可能触发更多次内部反射，导致光线能量集中在球体边缘。前后视图的光路呈现互补特性，前视图强调中心对称的折射与反射，后视图则凸显光线穿透与能量衰减的动态过程，验证了算法对光线传播物理规律的模拟。同时，前视图环境光通过球体正面折射后，形成渐变光晕，与直射光共同作用生成细腻的光照层次。而后视图的环境光需穿透球体背面，经过多次内部反射后，间接光照的传播路径更长，可能导致光能衰减更显著，背景光晕的亮度与范围可能减小。前后视图的间接光照差异反映了光线追踪算法对能量衰减与路径复杂度的精确模拟，进一步验证了其物理准确性。

于是，我们可以总结如下：首先，我们的实验很好的实现了折射下的模拟，其次我们的 Phong 光照模型在实验中展现出对视角变化的敏感捕捉，接着，实验成功区分了直射光与间接光对阴影的贡献，很好的模拟了阴影效果，最后，我们的递归算法通过追踪反射与折射路径，完整捕捉了间接光照的叠加效应。

5 结论与展望

本文通过实验，系统地研究了光线追踪技术在静态场景渲染中的应用，通过构建基于 Phong 光照模型与递归算法的光线追踪框架，实现了对折射、反射、阴影及全局光照的物理精确模拟。实验部分通过上下左右前后多视角中渲染水晶球图像，验证了光线追踪技术在实际运用中显著的光照效果和真实感。

尽管如此，其计算效率与噪点控制仍需进一步优化。在未来可以考虑引入 GPU 并行计算，典型的像 NVIDIA RTX 架构，利用光追核心加速光线-物体相交检测与递归渲染，提升计算效率。同时，结合时空降噪技术或双向路径追踪（Bidirectional Path Tracing），可以减少复杂光路导致的图像噪点。

参考文献

- [1] WHITTED T. An improved illumination model for shaded display[C]//Proceedings of the 6th annual conference on Computer graphics and interactive techniques. 1979: 14.
- [2] APPEL A. Some techniques for shading machine renderings of solids[C]//Proceedings of the April 30–May 2, 1968, spring joint computer conference. 1968: 37-45.
- [3] 徐松. 三维动画设计中的实时渲染技术应用[J]. 电视技术, 2024, 48(8): 79-81.
- [4] COOK R L, PORTER T, CARPENTER L. Distributed ray tracing[C]//Proceedings of the 11th annual conference on Computer graphics and interactive techniques. 1984: 137-145.
- [5] KAJIYA J T. The rendering equation[C]//Proceedings of the 13th annual conference on Computer graphics and interactive techniques. 1986: 143-150.
- [6] FUJIMOTO A, TANAKA T, IWATA K. Arts: Accelerated ray-tracing system[J]. IEEE Computer Graphics and Applications, 2007, 6(4): 16-26.
- [7] BENTLEY J L. Multidimensional binary search trees used for associative searching[J]. Communications of the ACM, 1975, 18(9): 509-517.
- [8] CLARK J H. Hierarchical geometric models for visible surface algorithms[J]. Communications of the ACM, 1976, 19(10): 547-554.
- [9] HAVRAN V. Heuristic ray shooting algorithms[D]. Ph. d. thesis, Department of Computer Science, 2000.
- [10] MEISTER D, OGAKI S, BENTHIN C, et al. A survey on bounding volume hierarchies for ray tracing[C]//Computer Graphics Forum: vol. 40: 2. 2021: 683-712.
- [11] BARRÉ-BRISEBOIS C, HALÉN H, WIHLIDAL G, et al. Hybrid rendering for real-time ray tracing[J]. Ray Tracing Gems: High-Quality and Real-Time Rendering with DXR and Other APIs, 2019: 437-473.
- [12] ZHU S. Survey: Machine Learning in Production Rendering[J]. arXiv preprint arXiv:2005.12518, 2020.