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）
21231303	张鲲鹏	6	25	14	33	78

实验结果分析比较单薄

北京交通大学

GPU 和实时光线追踪技术的发展

学　　院：软件学院

专　　业：软件工程

学生姓名：张鲲鹏

学　　号：21231303

指导教师：吴雨婷

北京交通大学

2024 年 6 月

摘要

实时渲染技术是计算机图形学的重要分支。随着技术的发展，GPU 的渲染能力不断提升，使得游戏、影视等领域的画面质量更加真实清晰。本文回顾了 GPU 的发展历程，并重点介绍了 NVDIA 的图灵架构 GPU 的实时光线追踪技术。实时光线追踪技术通过模拟光线在场景中的传播，可以更加真实地模拟光影、反射和折射效果，极大提高了图形渲染的逼真度。

关键词：实时渲染；显卡；GPU；光线追踪

1 引言

GPU 的发展历史可以追溯到上个世纪 80 年代，最初的 GPU 被用于简单的 2D 图形渲染，随后逐步发展到支持 3D 图形渲染的能力。在这个过程中，GPU 的渲染能力不断增强，从最初的几百万次每秒（MIPS）到目前主流显卡已经达到数万亿次每秒（TFLOPS）的级别。其中，NVIDIA 的图灵架构 GPU 是实时光线追踪技术的重要里程碑。实时光线追踪技术通过模拟光线在场景中的传播，可以更加真实地模拟光影、反射和折射效果，极大提高了图形渲染的逼真度。与传统的光栅化渲染相比，实时光线追踪技术在处理阴影、反射、折射等光学效果上更加出色，为计算机图形学的发展带来了新的突破。

2 GPU 发展历史

图形卡的历史可以追溯到 20 世纪 60 年代，当时计算机系统开始使用专门的硬件进行图形处理。早期的图形系统主要用于科学和工程应用，通常体积庞大、价格昂贵且功能有限。在 20 世纪 70 年代，微处理器和集成电路的发展促使了更经济、更紧凑的图形系统的诞生。第一批商用图形卡之一是 IBM 在 1981 年推出的黑白显示适配器（monochrome display adapter，MDA）。MDA 仅支持单色显示，提供 80 列×25 行的文本模式和 720×350 像素的分辨率。它不具备图形显示功能，无法显示位图或其他图形内容，仅仅提供了基本的文字显示能力。然而，MDA 作为早期个人电脑显示适配器的重要一步，为后续图形技术的发展奠定了基础。1981 年，IBM 推出了彩色图形适配器（CGA），这标志着图形卡演变中的一个重要里程碑。CGA 具有显示彩色图像和有限图形功能的能力，该适配器支持

640x200 像素的单色模式和 320x200 像素的 16 色模式，为用户提供了更丰富的视觉体验。1984 年，IBM 推出了增强图形适配器（EGA），与 CGA 相比，EGA 提供了更好的色深和分辨率。随后在 1987 年，视频图形阵列（VGA）问世，并在许多年里成为行业标准。VGA 提供了 256 种颜色和 640x480 像素的分辨率，使得图形更为详细、色彩更为丰富。VGA 在文字模式下可支持 720×400 分辨率，绘图模式下可支持 640×480×16 色和 20×200×256 色输出，VGA 标准一直沿用至今[1]。

20 世纪 90 年代图形技术快速发展，这一变革主要源于视频游戏的流行和对逼真的 3D 图形显示效果的需求。1992 年，S3 Graphics 推出了 S3 86C911，这是首款能够同时处理 2D 和 3D 图形的图形加速器，标志着专用图形处理单元（GPU）时代的开端。1995 年，Nvidia 公司成立，Nvidia 的首款产品 NV1 同样集成了 2D 和 3D 图形功能，并支持 Sega Saturn 游戏控制器，极大地推动了游戏显示技术的发展。与此同时，ATI Technologies 在 1995 年推出了 ATI Rage 系列显卡，不仅提供了 2D 和 3D 加速功能，还具备视频解码能力，标志着显卡解码视频的开端。到了 1999 年，Nvidia 发布了 GeForce 256，GeForce 256 是第一个被正式称为“图形处理单元”（GPU）的产品。传统上，3D 图形渲染中的几何变换和光照计算是由 CPU 来处理的，这不仅增加了 CPU 的负担，还限制了图形处理的效率。GeForce 256 通过在硬件中实现 T&L，使得这些计算能够在 GPU 上并行执行，从而大大提高了处理速度和效率。这一创新使得“GPU”这一术语被广泛接受和使用，标志着图形处理技术进入了一个新的时代。

在 2000 年代，Nvidia 和 AMD 在 GPU 市场展开了激烈竞争。Nvidia 推出了 GeForce 3 系列（2001 年），首次引入可编程顶点和像素着色器，推动了更复杂逼真的图形效果的开发。随后的 GeForce 4 系列（2002 年）提供了更高性能和多显示器支持。AMD 则在 2002 年发布了 Radeon 9700 Pro，首款支持 DirectX 9 的 GPU，带来高级像素和顶点着色器，以及 HDR 渲染。整个 2000 年代，Nvidia 和 AMD 相继推出了 GeForce 6 系列（2004 年）、Radeon X1000 系列（2005 年）、GeForce 8 系列（2006 年）和 Radeon HD 2000 系列（2007 年），这些产品引入了着色器模型和 DirectX 版本的新支持，推动了图形处理效率和功能的进一步提升。两家公司通过技术革新和竞争，极大地丰富了用户的视觉体验，推动了图形应用和游戏技术的快速发展。

在 2010 年代，GPU 不再仅用于图形处理，而是开始被应用于通用计算任务，推动了科学模拟、机器学习和数据分析等领域的快速发展。Nvidia 在 2006 年推出了 CUDA，成为通用 GPU 计算领域的关键技术，允许开发者使用 C、C++或 Fortran 编写代码并利用 GPU 的并行处理能力。AMD 在 2007 年推出了 ATI Stream

技术（后更名为 AMD APP Acceleration），采用 OpenCL 编程模型将 GPU 用于通用计算任务。2012 年，Nvidia 发布了 Kepler 架构，随后相继推出了 Maxwell（2014）、Pascal（2016）和 Turing（2018）架构，每一代都在性能、能效和 GPU 计算能力方面有显著提升。另一方面，AMD 在 2011 年引入了 Graphics Core Next（GCN）架构，之后推出了 RDNA（2019）和 RDNA 2（2020）架构，进一步提升了性能和能效。

3 实时光线追踪技术

光线追踪最早由 Arthur Appel 在 1968 年提出，最初用于生成阴影。这种方法通过追踪从光源到物体的光线来确定哪些部分在阴影中。1979 年，Turner Whitted 扩展了光线追踪技术，提出了一种能够生成反射和折射效果的算法，这一突破性的发展使得光线追踪可以模拟更复杂的光影效果。光线追踪（Ray Tracing）算法属于三维图形渲染算法，其基本出发点就是追踪光线，模拟真实的光路和成像过程。传统的光栅化渲染，只是将三维场景投影到二维屏幕上，并通过像素来表示场景，而光线追踪考虑了光线与物体的相互作用，可以产生逼真的阴影、反射、折射等效果，但与此带来了更长的渲染时间。

在图灵架构之前，现有的 GPU 主要针对光栅化渲染进行了优化，而光线追踪的需求与之不同。传统 GPU 缺乏专门用于加速光线追踪计算的硬件单元，这使得实时光线追踪的实现变得更加复杂和低效[2]。2018 年，NVIDIA 推出的图灵架构显卡（如 RTX 2080 和 RTX 2080 Ti）内置了专门的 RT Cores（光线追踪核心），这些核心专门用于加速光线追踪计算，从而显著提高了光线追踪的效率。

Turing 架构配备了名为 RT Core 的专用光线追踪处理器，能够以高达每秒 10 Giga Rays 的速度对光线和声音在 3D 环境中的传播进行加速计算。RT Core 是单独用来加速光线追踪 BVH 算法的硬件逻辑电路[3]。BVH（Bounding Volume Hierarchies）算法是一种广泛应用于光线追踪的高效算法。它通过递归地将场景中的几何对象组织成层次化的包围体树结构，显著减少了射线与几何对象求交测试的计算量。这种方法不仅能处理复杂场景中的大量三角形，还能优化动态场景中的实时更新，是现代光线追踪技术中不可或缺的核心算法。与传统的 CUDA Core 通用算法相比，RT Core 在 BVH 计算效率上实现了几何数量级的提升，使得实时光线追踪渲染成为可能。这种技术的进步极大地提高了游戏和影视行业的画面质量，实现了更加逼真和细腻的光影效果。RT Core 能够以极高的速度和并行处理能力处理复杂的光线与几何体交互，显著减少了渲染时间，为实时应用提供了强大的支持。

与硬件级光线追踪相辅相成的是Microsoft推出的DirectX Ray Tracing (DXR)技术。DXR扩展了DirectX 12 API，通过调用显卡中的光线追踪硬件来加速光线与物体的交互计算。这种方法相比传统的光栅化算法，显著提高了光线求交的速度和效率。DXR 不仅简化了开发者在游戏和图形应用中实现光线追踪的过程，还为实时渲染提供了强大的技术支持[4]。

综上所述，光线追踪技术自提出以来，经历了从软件实现到硬件加速的重大变革，特别是NVIDIA 图灵架构引入的RT Cores 和 Microsoft 的 DXR 技术，极大地提高了光线追踪的效率和实时渲染的可行性，使得光线追踪在游戏、电影、虚拟现实等领域展现出巨大发展潜力和广泛应用前景，实现了更加逼真和细腻的光影效果，标志着计算机图形学的又一次革命性进步。

4 光线追踪实验

4.1 实验目的

使用 DirectX 12 Raytracing (DXR) API 实现的简单光照渲染示例,了解 DirectX 12 中光线追踪技术的基本实现和工作原理。

4.2 实验环境

4.3 实验流程

1) 初始化

首先初始化 DirectX 12 和 DirectX Raytracing (DXR)的配置，并设置场景的参数，包括材质、摄像机、光照等。

设置摄像机的各项参数：

- m_eye：摄像机的位置，初始化为 (0.0f, 2.0f, -5.0f, 1.0f)。
- m_at：摄像机的目标位置，初始化为 (0.0f, 0.0f, 0.0f, 1.0f)。
- 计算摄像机的方向 direction 和上向量 m_up。
- 旋转摄像机，使其绕 Y 轴旋转 45 度。

设置光源参数：

- lightPosition：光源位置，设置为 (0.0f, 1.8f, -3.0f, 0.0f)。
- lightAmbientColor：环境光颜色，设置为灰色 (0.5f, 0.5f, 0.5f, 1.0f)。
- lightDiffuseColor：漫反射光颜色，设置为红色 (0.5f, 0.0f, 0.0f, 1.0f)。

2) 着色器配置

着色器从常量缓冲区和缓冲资源中访问所有输入数据。常量缓冲区包括：

- CubeConstantBuffer：包含立方体的颜色，通过着色器记录传递给着色器。
- SceneConstantBuffer：存储全场景的相机和光源参数，通过全局根签名提供。

三角形法线从索引和顶点缓冲区中访问，这些缓冲区显式传递给最近命中着色器。首先，从 16 位索引缓冲区加载三个命中三角形的顶点索引。然后，使用这些索引索引到顶点缓冲区并加载重复存储在每个三角形顶点上的三角形法线。

3) 加速结构配置

光线追踪需要使用加速结构 (Acceleration Structures)，包括底层加速结构 (Bottom-Level Acceleration Structures, BLAS) 和顶层加速结构 (Top-Level Acceleration Structures, TLAS)。

4) 创建着色器表

着色器表用于存储着色器记录，它包含着色器的入口点和相关资源。首先分配 Shader Table 的内存，用于存储着色器记录。然后填充 Shader Table，将着色器记录填充到表中，每个记录包含着色器标识符和资源指针。

5) 构建光线追踪管线

光线追踪总线将多个着色器集成到一个统一的管线中，它定义了光线追踪过程中的各种着色器和状态，包括如何生成、跟踪和处理光线，以生成最终的图像。光线追踪管线的工作流程：首先，应用程序创建加速结构来表示场景，然后，定义和编译光线追踪所需的着色器，并创建着色器表来组织这些着色器及其参数。接着，创建 RTPSO 来配置光线追踪管线。最后，应用程序提交光线追踪命令，GPU 执行光线追踪计算，并将结果输出到目标资源。

6) 执行光线追踪

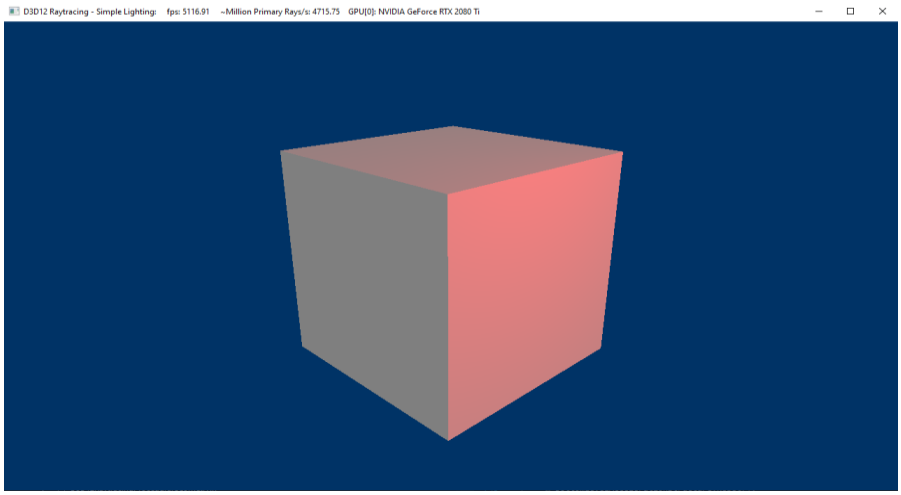
执行光线追踪以生成最终图像。光线追踪的执行包括调度光线追踪工作，并

将结果写入输出缓冲区。

- **绑定管线**：绑定之前创建的光线追踪管线。
- **调度光线**：调度光线以执行光线追踪。
- **复制结果**：将光线追踪结果复制到输出缓冲区。

4.4 实验结果

通过上述流程，可以在 DirectX 12 和 DXR 中实现简单的光线追踪场景。此实验的关键在于正确配置材质、摄像机和光源参数，并构建加速结构、着色器表和光线追踪管线，以提高光线追踪的效率。可以见到，渲染得到的立方体有着真实的阴影效果，通过配置立方体的材质参数，可以见到不同的光线反射与折射效果，体现了光线追踪技术带来的真实感。同时，渲染速度相比传统的光栅化要慢，GPU 的利用率的显存占用都比较突出。



参考文献

主要参考文献：

[1] 熊庭刚.GPU 的发展历程、未来趋势及研制实践[J].微纳电子与智能制造,2020,2(02):36-40.DOI:10.19816/j.cnki.10-1594/tn.2020.02.036.

[2] 郭笑孜. 基于光线跟踪的实时全局光照算法研究与实现[D].陕西师范大学,2022.DOI:10.27292/d.cnki.gsxfu.2021.002386.

[3] 姚晔. 基于光线追踪的阴影和反射渲染算法研究[D].安徽理工大学,2024.DOI:10.26918/d.cnki.ghngc.2023.000237.

[4] 黎小玉,田泽,郭亮,等.符合 OpenGL 标准的国产化显卡研究与实现[J].计算机技术与发展,2014,24(05):173-175+179.