

光线追踪

光线追踪是一种用于模拟光线与场景中物体的相互作用的计算技术。它模拟了光线从观察者（例如相机）发出，经过场景中的物体，并最终到达图像平面或传感器的过程。通过追踪光线的路径和相互作用，光线追踪可以生成高质量、逼真的图像和渲染效果。

光线追踪的基本原理是从相机发出一条射线，然后追踪该射线与场景中的物体是否相交。如果射线与物体相交，根据相交点处的材质属性、光照条件和相交点周围的几何信息，计算出光在该点上的反射、折射、遮挡等效果。然后，可以根据光线的反射和折射方向，继续追踪新的射线，以模拟光线在场景中的反射和折射。优点是能够模拟真实光线在场景中的传播和相互作用，因此可以生成高度逼真的图像。相比传统的渲染技术，如光栅化，光线追踪可以更准确地模拟光的行为，包括阴影、反射、折射、散射和全局光照等效果。

下面先简单介绍几种光线追踪的实现方法

1. 路径追踪 (Path Tracing)

路径追踪是一种用于逼真渲染的算法，它模拟了光线在场景中的传播和相互作用。与光线追踪不同，路径追踪考虑了光线的随机传播，通过追踪光线的路径来计算光线在场景中的能量传递，从而生成逼真的图像。路径追踪的基本原理是从相机位置发出一条射线，然后追踪该射线在场景中的路径。当光线与物体相交时，根据物体的材质属性，可以计算出光线在该点的反射、折射、散射等效果。然后，可以根据材质的反射和折射属性，以概率性地选择光线的下一个传播方向，继续追踪新的射线。通过迭代追踪和采样多条光线路径，最终可以估计出每个像素的颜色值，从而生成逼真的图像。优点是能够模拟真实光线的随机传播，包括全局光照和间接光照等效果。它可以处理复杂的光照场景，包括阴影、反射、折射、散射和光照的颜色混合等。然而，路径追踪也存在一些挑战，例如计算复杂度较高和需要大量的采样才能达到噪点较少的渲染结果。

2. 光子映射 (Photon Mapping)

光子映射是一种用于模拟光线与物体相互作用的渲染算法。与路径追踪不同，光子映射使用了两个阶段的过程：发射光子和通过光子映射重建图像。在光子映射的发射光子阶段，从光源发射大量的光子，并通过光线与物体相交来记录光子的路径和能量信息。这些光子在场景中传播并与物体相互作用，可能被吸收、反射或折射。当光子与物体相交时，可以根据物体的材质属性，计算光子的能量衰减和反射/折射方向。在光子映射的通过光子映射重建图像阶段，通过对光子的信息进行采样和聚合，估计每个像素的颜色值。光子映射使用了一种称为光子图的数据结构，它存储了场景中光子的位置、颜色和能量等信息。通过在光子图中搜索与像素相关的光子，并使用光子的信息来估计光照强度和颜色，最终可以重建出图像。光子映射的优点是可以有效地模拟

全局光照效果，包括间接光照和漫反射等。它对于处理复杂的光照场景、镜面反射和折射等效果表现出色。然而，光子映射也存在一些挑战，例如需要合理的光子密度和较长的计算时间来收集足够的光子样本。

辐射度量学

辐射度量学（Radiometry）涉及了光的辐射、传输和相互作用的测量和计算。它是用来描述和量化光线传播和能量传递的物理学分支。辐射度量学的目标是提供一套数量化的工具和概念，以便研究光的行为、进行光学设计和模拟光线传输过程。

在辐射度量学中，有一些重要的概念和度量参数：

Radiant Flux: 单位时间光源辐射的能量 $\phi = \frac{dQ}{dt}$

Radiant Intensity: 光源向单位立体角辐射的 Radiant Flux $I(\omega) = \frac{d\phi}{d\omega}$ 在球面上立体角为 $\frac{A}{r^2}$

Irradiance: 光源向单位面积辐射的 Radiant Flux $E = \frac{d\phi}{dA} = \frac{\phi}{4\pi r^2}$

Radiance: 光源向单位立体角和单位面积所辐射的 Radiant Flux $L = \frac{d^2\phi}{d\omega dA \cos \theta}$ 需要 $\cos \theta$

在计算机图形学中，辐射度量学的概念和技术被广泛应用于渲染算法（如光线追踪和路径追踪）中，以模拟光线与场景中物体的相互作用。它可以帮助生成逼真的图像和动画。在光学设计中，辐射度量学用于分析和优化光学系统的性能，例如计算光学器件的传输效率和光学系统的照度分布。在遥感中，辐射度量学用于解释和处理遥感图像中的辐射数据，以提取地表特征和环境信息。

蒙特卡洛路径追踪

蒙特卡洛路径追踪（Monte Carlo Path Tracing）的基本思想是通过随机采样和统计平均来估计光线路径上的积分值。路径追踪算法从相机位置发射一条光线，然后追踪光线在场景中的路径，直到光线被吸收、超过最大反射次数或离开场景为止。在追踪的过程中，通过在光线路径上进行随机采样，并对采样点处的光线属性进行计算，来估计光线路径上的积分值，例如光线与物体相交处的表面颜色、辐亮度或光线的反射率等。

下面是蒙特卡洛路径追踪的基本步骤：

1. 发射光线：从相机位置发射一条光线，确定光线的起点和方向。
2. 光线追踪：追踪光线在场景中的路径，直到达到终止条件。在每次追踪中，需要判断光线与场景中的物体是否相交，并计算交点的属性，如交点处的表面法线、材质属性等。

3. 采样光线属性：在光线路径上进行随机采样，选择一些位置用于属性值的估计。这些位置可以是光线与物体相交处，也可以是光线经过的其他位置。
4. 计算属性值：在采样点处计算所需的属性值，如表面颜色、辐亮度或光线的反射率。这可以根据物体的材质属性、光照模型和光线与物体相交点的几何信息进行计算。
5. 累积属性值：对所有采样点的属性值进行累积，计算属性值的平均值。
6. 重复迭代：重复执行步骤2到步骤5，生成多个样本并累积平均值。通常需要生成足够多的样本以减小估计误差。
7. 估计像素颜色：最终的像素颜色值可以通过对累积的属性值进行平均来获得。

蒙特卡洛路径追踪的优点是能够处理复杂的光照效果和间接光照，而不需要对场景进行显式建模。它适用于各种光照模型和材质属性，并且可以通过增加样本数量来提高渲染结果的质量。然而，蒙特卡洛路径追踪也存在一些问题，如估计误差、噪声和收敛速度较慢。为了解决这些问题，可以使用一些技术，如重要性采样、路径重要性抽样和渐进路径追踪等。

重要性采样

在蒙特卡洛路径追踪中，重要性采样（Importance Sampling）是一种常用的技术，用于改善采样的效率和减少估计误差。它的作用是在采样过程中选择更有意义的样本，以提高路径积分的精度和收敛速度。在路径追踪算法中，光线路径上的每个点都需要对光线的属性进行采样，例如材质颜色、光照信息等。传统的均匀采样在每个采样点上平均分配样本，但这样的采样方式可能导致对某些重要的光线路径部分的采样不足，从而产生高方差和噪声。重要性采样通过根据采样点的重要性调整采样分布，更聚焦于对渲染方程中具有较大贡献的样本。具体而言，它根据场景的光照分布和材质属性等信息，调整样本的采样分布，使其更倾向于采样那些对最终结果有更大影响的光线路径。重要性采样的基本思想是使用一个称为重要性函数的概率密度函数来指导采样过程。这个函数将样本的分布调整为与渲染方程的贡献相关，以使得对于具有更大贡献的路径部分，样本更有可能被选中。

通过重要性采样，可以减少噪声和估计误差，提高渲染结果的质量。因为重要性采样将样本集中在对结果贡献较大的路径上，减少了采样在对结果贡献较小的路径上的浪费。这样可以在保持相同样本数量的情况下，更有效地估计路径积分的值，重要性采样并不能完全消除估计误差，而是通过合理选择采样点，使得误差的方差更均匀地分布在图像中，从而减少了噪声的出现。在实践中，常常需要根据场景的特点和需求，选择合适的重要性函数来进行采样，以获得最佳的渲染结果。

这是我实现的一个简单的重要性采样的蒙特卡洛路径追踪渲染器

ReSTIR (Recursive Sampled Importance Resampling)

ReSTIR是一种用于实时渲染的高效光线追踪算法。它结合了光线追踪和重要性采样的思想，通过递归采样和重采样的方式，在实时渲染中提供高质量的全局光照效果。该算法的核心思想是使用一种称为Virtual Shadow Maps (VSM) 的数据结构来存储场景的光照信息。VSM是一种基于阴影贴图的数据结构，用于在每个像素上存储光照的累积贡献。这些累积贡献可以用于生成重要性采样的分布，以指导光线的采样过程。

下面是ReSTIR算法的基本步骤：

1. Virtual Shadow Maps构建：首先，使用传统的光线追踪算法，计算场景中每个像素的VSM。这涉及追踪多条光线，计算它们与场景中物体的交点，并根据光照模型和材质属性计算光线的贡献。
2. 初始采样：在每个像素上，使用VSM的累积贡献作为重要性分布，对初始样本进行采样。这些初始样本代表了每个像素上的光线路径。
3. 递归采样：从初始样本出发，通过递归地追踪光线，生成更多的样本。在每次追踪中，根据重要性分布对采样点进行采样，以选择具有较大贡献的路径部分。
4. 重要性评估：在递归采样过程中，对光线路径上的每个样本，根据场景的光照信息和材质属性，计算路径的重要性贡献。这些贡献用于更新VSM的累积贡献。
5. 重采样：根据重要性贡献，对生成的样本进行重采样。重采样的目的是根据样本的贡献，选择更有意义的样本，以减少估计误差。
6. 累积结果：对重采样后的样本进行累积，计算最终的像素颜色。

ReSTIR通过递归采样和重采样的方式，充分利用了VSM存储的光照信息，以高效地生成具有全局光照效果的图像。它在实时渲染中具有较低的延迟，并且能够处理复杂的光照效果和间接光照。相比于传统的路径追踪算法，ReSTIR在保持渲染质量的同时，显著降低了计算成本，使得实时渲染更加可行。

NRC (Neural Radiance Caching)

NRC 是一种用于实时渲染的光线追踪技术，它通过神经网络的辅助来提高渲染效率。NRC结合了光线追踪和神经渲染的思想，通过缓存和重用光线路径上的信息，以加速渲染过程。

下面是Neural Radiance Caching的基本步骤：

1. 初始光线追踪：使用传统的光线追踪算法，从相机位置发射光线，与场景中的物体进行交点计算，并根据光照模型和材质属性计算光线的贡献。
2. 几何缓存：在每个像素上，将光线路径的几何信息（例如交点位置、法线等）缓存起来。这些几何信息将在后续的渲染中被重复使用。
3. 辐射缓存：对于每个像素，使用神经网络（通常是基于深度学习的）来估计光线路径的辐射（Radiance）。这个神经网络可以是预先训练好的，也可以是实时训练得到

的。

4. 辐射缓存重用：在渲染过程中，对于新的光线路径，通过查询辐射缓存来重用之前计算的辐射值。如果缓存中存在匹配的几何信息，可以直接返回之前计算的辐射值，避免重复计算。
5. 重新渲染缺失的路径：对于无法从辐射缓存中获得辐射值的光线路径，使用传统的光线追踪算法进行重新渲染。这些路径通常是由于缓存未命中或者缓存过期导致的。

通过上述步骤，NRC 能够高效地重用之前计算的光线路径信息，减少了重复计算的开销，从而提高了渲染效率。同时，通过神经网络的辅助，可以对辐射进行更精确的估计，提升了渲染质量。关键在于几何缓存和辐射缓存的设计和管理。几何缓存用于存储光线路径的几何信息，需要考虑缓存的大小和命中率。辐射缓存用于存储光线路径的辐射信息，需要选择合适的神经网络结构和训练方法，以实现高质量的辐射估计。

NRC 已经在实时渲染中具有广泛的应用，可以用于生成高质量的全局光照效果和间接光照。它在提高渲染效率的同时，能够处理复杂的光照场景和材质属性，并且具备一定的实时交互性。