用于光线追踪数据光栅化的辐射度纹理

Jakub
MaksymilianFobertalk@maxfober.space

arXiv:2301.01719v1 [cs.GR] 4 Jan 2023

摘要

介绍一种利用辐射度纹理对三维表面进行实时渲染的方法，可快速合成随入射角变化的复杂效果及环境交互。这些效果包括虹彩、视差遮蔽和内部映射；多次反弹的反射（镜面反射、常规反射、漫反射和全内反射）；折射、次表面散射、透明度等。该方法将纹理划分为辐射度桶矩阵，每个桶代表不同入射角下的数据。数据可以是最终像素颜色，也可以是延迟渲染中的环境光遮蔽、反射、阴影贴图等。最终合成输出的分辨率取决于辐射度桶矩阵的大小。该技术可通过简单的片段着色器实现。其计算开销相当于简单的仅漫反射图形，却能以存储内存占用为代价，达到复杂（离线）光线追踪渲染的视觉保真度。通过调整重复辐射度场景纹理的压缩比，可以轻松平衡计算开销与存储内存占用。

CCS 概念：• **计算方法 → 反射率建模**； **光栅化**； **纹理映射**；光线追踪。

关键词：三维图形、全息术、光场、全光、辐射场、光栅化、光线追踪、反射场

© 2023 Jakub Maksymilian Fober



本作品采用知识共享署名—非商业性使用—禁止演绎 3.0 许可协议。
https://creativecommons.org/licenses/by-nc-nd/3.0/ 如需用于其他用途（包括商业用途），请联系所有者/作者。

1 引言

辐射场和反射场渲染技术是一类用于计算机图形学的算法，可生成三维场景的图像。这些算法模拟光线与虚拟环境中表面相互作用的方式，从而生成逼真且细节丰富的图像。

这些技术一直是计算机图形学与渲染领域广泛研究的课题，因为它们提供了一种强大而灵活的方法，可用于生成高质量图像。辐射场和反射场算法应用广泛，包括电影和电子游戏制作、建筑可视化以及科学可视化。

本文介绍了一种技术，通过将辐射场纹理嵌入三维物体表面，来捕获和渲染复杂的预计算光照交互。

所提出的技术利用标准片段像素着色器和二维纹理查找，以极低的计算成本渲染动态、视点无关的照片级真实感图像，其计算成本仅为实时光线追踪、视差映射和动态阴影等效果的一小部分。

该技术非常适合在现代硬件上用于电子游戏、虚拟现实和虚拟制作环境中的实时执行。它能够利用第九代游戏系统的直接存储能力，提供高保真、高性能的图像。

该技术可以取代计算开销巨大的渲染管线链，同时保留经硬件加速、高度优化的光栅化元素。它还可以推动实时 GPU 光线追踪得到更广泛的应用，并能够将反弹光线与环境的预计算辐射度相结合。

1.1 先前工作

主流的辐射场渲染实现主要采用体积数据结构和球谐函数来渲染图像[Yu 等人 2021]。虽然体积数据可以采用稀疏表示，以排除空区域[Yu 等人 2021]，但从根本上说，最终目标理应是与所表示物体的几何形状完美匹配。而且，由于物体内部的体积通常并无意义，实际上只有辐射球的一半具有实用价值。因此，这类场实际上可以分布在物体表面。

一些研究者采用了这种方法，将神经反射场用作纹理基元[Baatz 等人 2022]，并渲染出了高保真结果。不过，虽然神经场能产生出色的结果，但它们在渲染时计算开销很大[Yu 等人 2021]，因此不适用于实时应用。

1.2 内容概述

在本文的初始版本中，您将看到对该主题的理论阐释和实现内容，以及相关公式和示意图。一些元素（例如映射函数）已经过测试，另一些则将在后续更新中介绍。

1.3 文档命名约定

本文档采用以下命名约定：

- 左手坐标系。

- 以列的形式原生呈现向量。
- 按行优先顺序排列的矩阵，记作“M行 列”。
- 矩阵乘法表示为“$[列]_a \cdot [行]_b = M_{ab}$”。
- 单竖线括起来的“$|u|$”表示标量绝对值。
- 单竖线括起来的“$|\vec{v}|$”表示向量长度。
- 带有或不带算术符号的向量均逐分量计算，并形成另一个向量。
- 居中的圆点“·”表示向量点积。
- 带逗号“$[f, c]$”的方括号表示区间。
- 内部留空的方括号“$[x\ y]$”表示向量和矩阵。

- “$^{-1}$”的幂表示该值的倒数。
- 证毕符号“□”表示最终结果或输出。

这种命名约定简化了将公式转换为着色器代码的过程。

2 方法论

模型纹理的每个像素都包含一个大小为 $n \times n$ 的离散辐射半球映射，称为“桶”。桶取代初始纹理的像素排列，使整体分辨率提升至 $w \cdot n \times h \cdot n$ 像素，其中 w 和 h 分别表示合成输出纹理的宽度和高度 。桶具有很高的重复性，彼此之间仅有细微差异。这种情况非常适合采用简单的压缩方法。

要为给定视点位置合成输出纹理，每个桶仅采样一次，从而得到正常分辨率的纹理输出。

模型的 u，v纹理坐标对应桶矩阵中的位置索引，而入射向量对应桶内部的 u，v位置。因此，辐射纹理采样算法可以描述为一个四维全光函数 $L(u,\ v,\ \theta,\ \phi)$，其中 u，v表示模型的纹理坐标，θ，ϕ表示入射角。

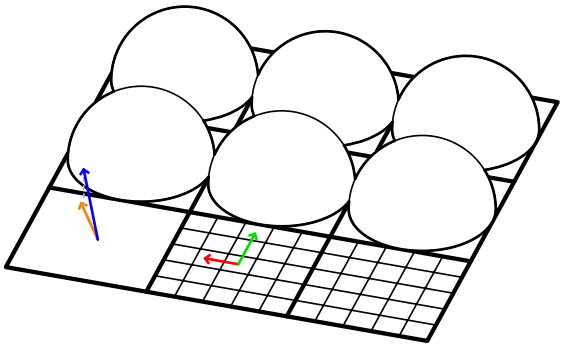

图1. 辐射度纹理采样模型：入射 $\mathbb{R}^3$ 向量（蓝色）经投影并转换为方形（橙色），得到 $\mathbb{R}^2$ 纹理坐标（红色和绿色），这些坐标映射到半球形辐射桶上，该桶以平面正方形表示。

每个辐亮度桶应表示反射率的一个半球。选择等面积方位投影来完成这项任务，因为它具有相应的特性：既能保持面积，又类似于球面镜反射[维基百科贡献者 2022]。在这种投影中，辐亮度桶的分辨率直接对应于 $\sin(\theta/2)\sqrt{2}$，其中 θ 为入射角。

为将信息高效分布到正方形桶中，我们还实现了圆盘到正方形映射函数，使正交方向和对角方向上的像素数量均匀。

等面积方位投影映射可在向量域中轻松实现，无需使用反三角函数，因为入射向量与法向量之和经正交投影和归一化后的长度为 $\sin(\theta/2)$。这样，全光函数中便不再需要 θ，ϕ 角度，从而得到新的 $L'(u,\ v,\ x,\ y,\ z)$，其中x，y，z对应于正交纹理空间中的入射单位向量分量。

2.1 将入射向量映射到辐射度桶

对于每个可见像素，都有一个入射向量 $\hat{I} \in \mathbb{R}^3$。该向量可通过平移和 $\mathbb{R}^{2\times3}$-矩阵变换进行映射和投影，得到 $\mathbb{R}^2$ 纹理坐标。

以下方程将入射向量映射到方位等距投影，其中 $r = 1$位于 $\Omega = 180°$处。

$$\begin{bmatrix} \vec{A}_x \\ \vec{A}_y \\ \sqrt{2}\cos\theta/2 \end{bmatrix} = \sqrt{2} \left\| \begin{bmatrix} \hat{I}_x + \hat{N}_x \\ \hat{I}_y + \hat{N}_y \\ \hat{I}_z + \hat{N}_z \end{bmatrix} \right\| \tag{1a}$$

$$\begin{bmatrix} \vec{A}_x \\ \vec{A}_y \end{bmatrix} = \frac{\sqrt{2}}{\left\| \begin{bmatrix} \hat{I}_x & \hat{I}_y & \hat{I}_z + 1 \end{bmatrix} \right\|} \begin{bmatrix} \hat{I}_x \\ \hat{I}_y \end{bmatrix} \quad \text{, if } \hat{N}_z = 1 \tag{1b}$$

逆映射：

$$\begin{bmatrix} \hat{A}'_x \\ \hat{A}'_y \\ \hat{A}'_z \end{bmatrix} = \begin{bmatrix} \vec{A}_x\sqrt{1/2} \\ \vec{A}_y\sqrt{1/2} \\ \sqrt{1 - \frac{\vec{A}_x^2}{2} - \frac{\vec{A}_y^2}{2}} \end{bmatrix} \tag{2a}$$

$$\begin{bmatrix} \hat{I}_x \\ \hat{I}_y \\ \hat{I}_z \end{bmatrix} = 2\left(\hat{A}' \cdot \hat{N} \begin{bmatrix} \hat{A}'_x \\ \hat{A}'_y \\ \hat{A}'_z \end{bmatrix} - \begin{bmatrix} \hat{N}_x \\ \hat{N}_y \\ \hat{N}_z \end{bmatrix} \right) + \begin{bmatrix} \hat{N}_x \\ \hat{N}_y \\ \hat{N}_z \end{bmatrix} \tag{2b}$$

$$= \begin{bmatrix} \vec{A}_x\sqrt{2 - \vec{A}_x^2 - \vec{A}_y^2} \\ \vec{A}_y\sqrt{2 - \vec{A}_x^2 - \vec{A}_y^2} \\ 1 - \vec{A}_x^2 - \vec{A}_y^2 \end{bmatrix} \quad \text{, if } \hat{N}_z = 1 \tag{2c}$$

其中，$\vec{A} \in [-1, 1]^2$ 是方位等距投影坐标。θ 是入射角。$\hat{N} \in \mathbb{R}^3$ 是表面法向量。当入射角$\hat{I} \in \mathbb{R}^3$ 映射到或从$\hat{}$正交纹理空间映射时，其中 $N_z = 1$，该变换可采用公式 1b 和 2c 的形式。

以下公式将方位投影向量变换为正方形坐标，用于辐射度桶采样。[1]

$$\begin{bmatrix}\vec{B}_x\\ \vec{B}_y\end{bmatrix} = \frac{\left|\left[\vec{A}_x \quad \vec{A}_y\right]\right|}{\max\left(|\vec{A}_x|,|\vec{A}_y|\right)}\begin{bmatrix}\vec{A}_x\\ \vec{A}_y\end{bmatrix} \quad \text{if } \vec{A}_x \text{ and } \vec{A}_y \neq 0 \tag{3}$$

其中，$\vec{B} \in [-1,1]^2$ 是桶的中心化纹理坐标，$\vec{A} \in [-1,1]^2$ 是方位投影向量。

注意。必须防止相邻桶边缘之间发生像素混合。可以将桶坐标限制在 $\vec{B} \in [B_{res}^{-1} - 1, 1 - B_{res}^{-1}]^2$ 范围内。

将桶的中心化坐标$\vec{B} \in \mathbb{R}^2$ 逆变换为方位投影坐标 $\hat{} A \in \mathbb{R}^2$，可采用相同但反向的方法。

$$\begin{bmatrix}\vec{A}_x\\ \vec{A}_y\end{bmatrix} = \frac{\max\left(|\vec{B}_x|,|\vec{B}_y|\right)}{\sqrt{\vec{B}_x^2 + \vec{B}_y^2}}\begin{bmatrix}\vec{B}_x\\ \vec{B}_y\end{bmatrix} \tag{4a}$$

$$\begin{bmatrix}\hat{R}_x\\ \hat{R}_y\\ \hat{R}_z\end{bmatrix} = \begin{bmatrix}-\vec{A}_x\sqrt{2 - \vec{A}_x^2 - \vec{A}_y^2}\\ -\vec{A}_y\sqrt{2 - \vec{A}_x^2 - \vec{A}_y^2}\\ 1 - \vec{A}_x^2 - \vec{A}_y^2\end{bmatrix} \tag{4b}$$

其中，$\hat{} R \in \mathbb{R}^3$ 表示等立体反射向量。该向量用于将光线追踪数据采样到辐射场纹理上。它是沿法线镜像后的向量，见前一页的公式 2c。

3 结果

TBA

4 结论

我探讨了利用现代硬件着色能力实现辐射场纹理化的可能方案，并提出了实现这一概念的数学解决方案。

注意。论文更新后将补充更多结论。

5 可能的应用

辐射场纹理采样可以取代着色管线，也可以为其补充增强效果。这些效果包括：

视差室内映射。这种效果可模拟透过窗户看到的室内景象，也可模拟通往另一个地方的传送门。

带视差映射的代理网格。带阿尔法蒙版的辐射度纹理可以模拟绑定在代理网格内部的更复杂或毛茸茸的物体。与神经辐射场对基元进行纹理映射类似[Baatzetal. 2022]。

[1]视觉参考请见图 2。



(a) 一枚美国一美分硬币的图片。



(b) 使用公式 3 将一美分硬币映射到矩形。

图 2。使用公式 3 中的表达式进行圆盘到正方形映射的示例。

反射。辐射度纹理贴图的单个像素中可以合并多次光线反弹。动态物体随后可以采样此类辐射场，以获得环境反射。此外，近实时光线追踪可以将动态生成的反射累积到此类纹理贴图中，从而更新并增强环境纹理贴图。

阴影。1 位辐射场纹理图可表示静态物体的阴影。此处，在阴影遮挡采样中，入射向量由光方向向量取代。它适用于平行光源和点光源。每个桶使用多个采样时，还可以生成区域阴影。

次表面散射。这种计算开销很大的效果可以编码到辐射度纹理贴图中，采样时，该贴图便会取代入射向量，并根据视点位置使用光方向向量。

参考文献

H. Baatz、J. Granskog、M. Papas、F. Rousselle 和 J. Novák。2022年。《NeRF-Tex：神经反射场纹理》。*Computer Graphics Forum* 41, 6（2022年3月），287–301。https://doi.org/10.1111/cgf.14449

维基百科贡献者。2022年。鱼眼镜头：映射函数。维基百科，自由的百科全书。https://en.wikipedia.org/w/index.php?title= Fisheye_lens&oldid=1124809304#Mapping_function［Online］。Alex Yu、Sara Fridovich-Keil、Matthew Tancik、Qinhong Chen、Benjamin Recht 和 Angjoo Kanazawa。2021年。Plenoxels：无需神经网络。*arXiv*（2021年12月）。https://doi.org/10.48550/ARXIV.2112.05131

收稿于2023年1月