

# 可视化实时渲染引擎的设计

勾炜焯, 孙金萍\*, 马小龙, 张晨阳

徐州工程学院信息工程学院(大数据学院), 江苏 徐州

收稿日期: 2022年3月16日; 录用日期: 2022年6月10日; 发布日期: 2022年6月17日

## 摘要

渲染引擎封装了图像渲染等相关技术, 为开发者提供统一接口来进行图像程序开发。通过对三维实时渲染进行研究, 运用基于物理的渲染、多重采样抗锯齿等技术, 开发了一款快捷高效的可视化渲染引擎。实现了自动模型加载、多线程渲染和实时调节渲染画面设置等功能。可以应用于电影特效制作, 三维动画渲染等领域。

## 关键词

可视化, 自动模型, 渲染引擎

# Design of Visual Real-Time Rendering Engine

Weixuan Gou, Jinping Sun\*, Xiaolong Ma, Chenyang Zhang

School of Information Engineering (School of Big Data), Xuzhou University of Technology, Xuzhou Jiangsu

Received: Mar. 16<sup>th</sup>, 2022; accepted: Jun. 10<sup>th</sup>, 2022; published: Jun. 17<sup>th</sup>, 2022

## Abstract

The rendering engine encapsulates image rendering and other related technologies, and provides a unified interface for developers to develop image programs. Through the research of 3D real-time rendering, a fast and efficient visual rendering engine is developed by using physics based rendering, multi sampling anti aliasing and other technologies. It realizes the functions of automatic model loading, multi-threaded rendering, real-time adjustment of rendering picture settings and so on. The rendering engine can be applied to film special effects production, 3D animation rendering and other fields.

## Keywords

Visualization, Automatic Model, Render Engine

\*通讯作者。

Copyright © 2022 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

随着计算机的发展, 三维渲染技术也正在突飞猛进的发展。快速、高质量的渲染[1]能提供接近物理真实的场景画面, 大大扩展了显示系统的应用。3D 光场显示和平面显示的成像机理不同, 平面显示的渲染方法在 3D 光场渲染上并不适用。

在渲染引擎发展较早的国外, 有一款名叫 UNREAL ENGINE 的引擎, 已经更新换代到第五代。Nanite 虚拟几何体的出现意味着由数以亿计的多边形组成的影视级美术作品可以被直接导入虚幻引擎——无论是来自 Zbrush 的雕塑还是用摄影测量法扫描的 CAD 数据。Nanite 几何体可以被实时流送和缩放, 因此无需再考虑多边形数量预算、多边形内存预算或绘制次数预算; 也不用再将细节烘焙到法线贴图或手动编辑, 画面质量不会再有丝毫损失[2] [3]。

如今渲染引擎的应用已经不仅仅只在游戏行业有所运用, 在建筑工程、影视动画等等领域都在大展身手[4] [5]。国外的渲染技术如雨后春笋般生长, 国内的渲染技术不断进步, 但如何实现高效的渲染引擎仍需要进一步研究。

通过对 3D 渲染引擎的功能研究分析, 设计了一款可视化渲染引擎。采用基于物理的渲染方式, 这种渲染方式使用了更先进的法线分布函数, 拥有了更宽尾部和更高峰值的 *trowbridge-Reitz* 分布。图像的光源照明使用了基于物理的渲染技术中的基于图像的照明, 还使用了点光源、聚光灯源、定向光, 抗锯齿采用了多重采样抗锯齿的方式。引擎对每个功能模块进行了详细分析, 在设计中以模块的效率为主要目标, 兼顾易开发性, 并用微软提供的 *DirectX12* 库实现了功能模块, 模块包括了纹理、材质、光照、场景管理、天空盒、摄像机。

## 2. 抗锯齿的渲染引擎

### 2.1. 辐射度量学

为了渲染出物理上正确的图像, 需要从能量的角度上去理解光的传播、光在空间中的分布, 而这也刚好是辐射度量学的范畴。

在三维中, 使用球面上的面积  $A$  与所队员的半径平方之比来定义一个立体角, 而且这个立体角是与球体半径大小无关的, 如图 1 所示。

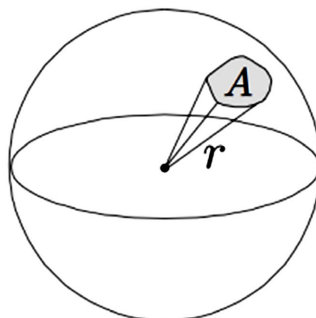


Figure 1. Solid angle

图 1. 立体角

考虑图 2 所示空间中的点光源，以球心为顶点的无限长锥体内，任何截面接收到的光子能量应该是一样的。换句话说，近处较小面积的表面和远处较大面积的表面所接收到的能量应该相同。立体角可以表示为类似比例关系。

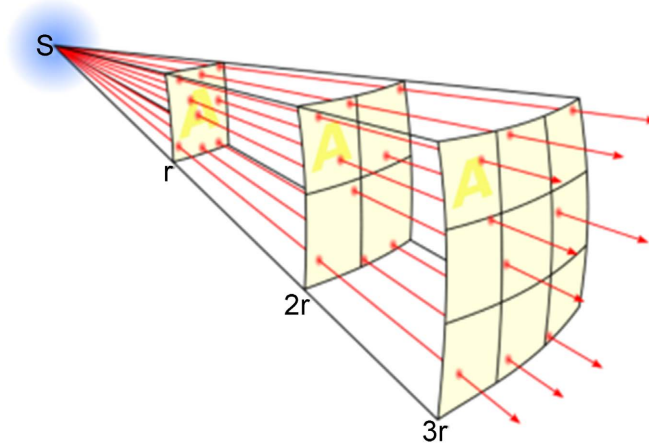


Figure 2. Radiation proportion at different distances  
图 2. 不同距离受辐射比例

### 2.2. 双向反射分布函数(Bidirectional Reflectance Distribution Function, BRDF)

光的漫反射是认为表面足够粗糙以至于光打到该表面时会往各个方向均匀发散，在 BRDF 中表现为：

$$k_d \frac{c}{\pi} \tag{1}$$

其中， $c$  为纹理颜色， $k_d$  为漫反射系数。

微平面模型中一个重要特性就是微平面法线的统计分布。法线分布函数  $D$  描述了平面法线分布的概率，即具有正确朝向的微表面法线浓度。

当法线分布比较集中时，物体表面材质会更容易出现高光；当法线分布比较散开时，物体表面将非常暗淡。如图 3 所示。

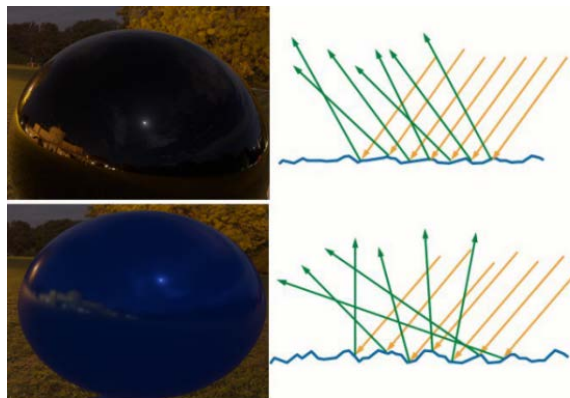


Figure 3. Contrast of different normal concentration  
图 3. 不同法线集中度对比

由于物体的表面不会反射所有的光线，在一部分光线被反射之后，还会有一小部分光线折射进入物

体的内部与物体之间的微小粒子不断碰撞并散射到随机方向,即漫反射。双向反射分布函数(Bidirectional Reflection Distribution function, BRDF)定义如下:

$$f_r = \kappa_d f_{\text{lambert}} + \kappa_s f_{\text{cook-torrance}} \quad (2)$$

其中,  $\kappa_d$  和  $\kappa_s$  表示射入光纤的折射比率和被反射比率, 而  $f_{\text{lambert}}$  指漫反射的 BRDF,  $f_{\text{cook-torrance}}$  指镜面反射的 BRDF。

$f_r = \frac{1}{\pi}$ 。在考虑上反射率之后, 即可得到最终的漫反射 BRDF:

$$f_{\text{lambert}} = \frac{\rho}{\pi} \quad (3)$$

还有一个镜面反射相对于漫反射, 要更加复杂一些:

$$f_{\text{cook-torrance}} = \frac{F(l, h)G(l, v)D(h)}{4(n \cdot l)(n \cdot v)} \quad (4)$$

其中  $D$  为法线分布函数, 可以理解为微观下镜面法线的分布情况。

函数  $G$  为几何函数, 描述的是微观下平面的遮挡属性。函数  $F$  为菲涅尔方程, 描述的是物体表面在不同光角度下反射光线所占的比率。且  $G$  和  $F$  满足:

$$\begin{aligned} 0 \leq G(l, v) \leq 1 \\ 0 \leq F(l, h) \leq 1 \end{aligned} \quad (5)$$

### 3. 实验结果与分析

从抗锯齿、天空盒、基于物理和光照渲染四个方面进行实验, 并进行分析。

#### 3.1. 抗锯齿渲染效果

锯齿是几何体走样(几何物体的边缘有锯齿), 由于对几何边缘采样不足导致的。由于渲染精度的问题, 现代渲染引擎难免出现走样, 在物体边缘会出现锯齿状的线条, 这显然不是想得到的结果, 于是就有了抗锯齿的出现, 抗锯齿的方法有很多种, 超采样抗锯齿, 多重采样抗锯齿等, 本引擎采用的是多重采样抗锯齿。

若采样点在红色直线之外, 那么这个像素点就不会被采样, 即不会被着色。但区别与计算机的我们能明显的知道, 这个像素点有大约 1/4 的区域被划在直线内, 但由于采样点不在直线内所以没有被采样, 所以系统则判断其不在三角形内, 所以不以着色, 最后呈现的是背景色。而下面那个像素点的采样点在直线内, 所以最终呈现的就是红色, 如图 4 所示。图 5 是抗锯齿开启后的渲染效果, 边缘清晰且光滑。

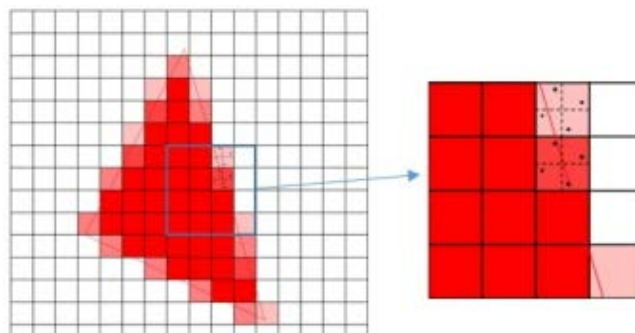


Figure 4. Anti aliasing example  
图 4. 抗锯齿示例

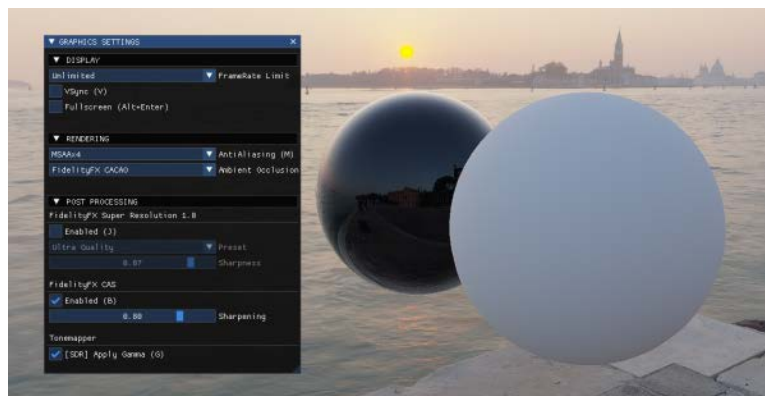


Figure 5. Anti aliasing module

图 5. 抗锯齿模块

### 3.2. 天空盒渲染

在渲染引擎中, 如果想要立即搭建一个基础的场景, 且这个场景距离当前的视角非常的远, 即使当前的视角怎么移动, 都不会发生什么变化, 这时候运用天空盒技术是非常方便的。天空盒就是将一个立方体展开, 然后分别在六个面上贴上贴图然后将视角也就是摄像机至于天空盒的正中间, 天空盒将摄像机包围, 最后根据摄像机视线与立方体的交点的坐标(x, y, z)来确定要对那个面进行采样。取 x, y, z 中绝对值最大的那个分量, 再根据它的符号来判断对哪个面进行采样。最后让其他两个值都除以最大的那个值, 这样最后得到的值就在[0, 1]内, 就可以直接在对应的贴图上进行映射, 效果如图 6 所示。

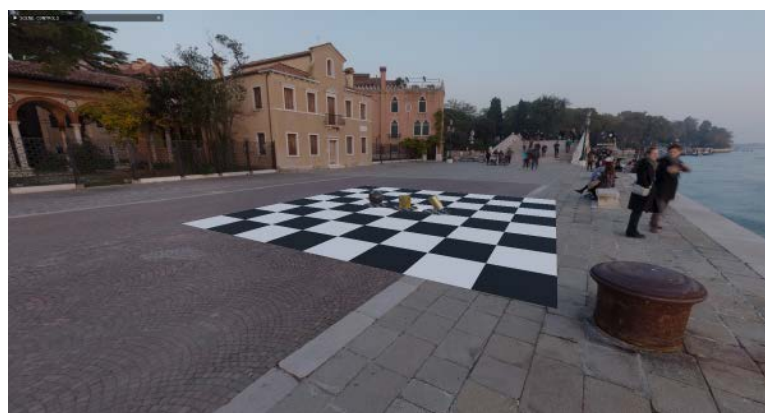


Figure 6. Sky box renderings

图 6. 天空盒效果图

### 3.3. 基于图像的光照渲染

如果想获得非常好的光源渲染, 则需要分析巨大的数据来进行计算, 并设置多个光源, 这显然要耗费太多的资源了。

但如果把天空盒贴图上的颜色信息作为光源, 再通过反射方程的计算, 会大大减少计算量, 而且最终效果也非常不错。当采用了微平面理论和基于物理的双向反射分布函数之后, 具体效果如图 7 所示。

### 3.4. 基于物理的渲染

当控制不同的法线集中度时, 即可使渲染的物体表面反射度产生变化, 具体变化如图 8 所示。

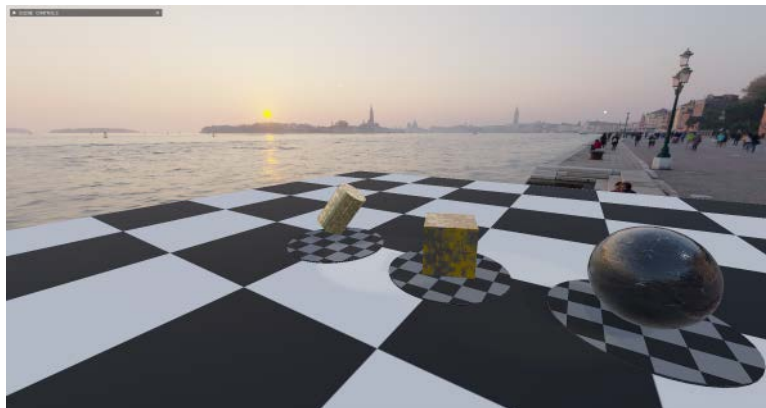


Figure 7. Image based lighting example

图 7. 基于图像的光照示例

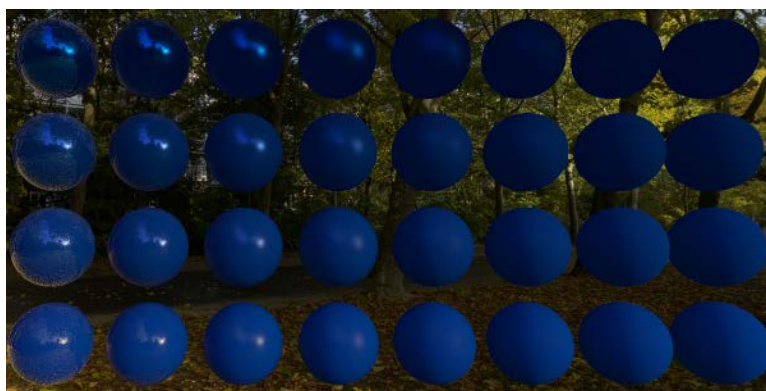


Figure 8. Rendering of different normal distribution functions

图 8. 不同法线分布函数的渲染

#### 4. 结论

首先分析国内外渲染引擎和其发展状况, 对其中涉及到的几何变换、光照、纹理映射处理以及面临的问题做了研究。实现了纹理、材质、光照、场景管理、天空盒、摄像机等功能。

由于时间、资源、技术能力有限, 引擎还不是很完善, 在很多地方还需要做进一步的改进和扩充, 比如现在的 UI (User Interface, UI) 是用的一款开源 UI, 和引擎匹配不好, 还需要开发配套的 UI。

#### 基金项目

江苏省大学生创新训练计划项目, 可视化渲染引擎开发与前端优化(xcx2021315)。

#### 参考文献

- [1] 李远航. 三维光场显示实时编码与渲染方法研究[D]: [博士学位论文]. 北京: 北京邮电大学, 2021.
- [2] 孟令增, 王中训. 基于 PBR 的渲染算法研究[J]. 电声技术, 2021, 45(11): 58-60.
- [3] 曹天池, 李秀实, 李丹, 陈嘉南, 刘爽, 向未来, 胡迎松. 基于图像分解的光照估计算法[J]. 计算机工程与科学, 2021, 43(8): 1422-1428.
- [4] 张宇剑. 图形处理器三角形光栅化和抗锯齿算法研究与实现[D]: [硕士学位论文]. 西安: 西安电子科技大学, 2018.
- [5] 景志敏. 基于 GPU 的矢量地图连续 LOD 渲染方法研究[D]: [硕士学位论文]. 南京: 南京师范大学, 2021.