

图形绘制技术大作业报告：基于 Ray Marching 的实时体积雾

胡同学

2026 年 6 月 12 日

1 选题理解与目标

体积雾的目标是在渲染中模拟空气、烟雾、尘埃等参与介质对光的吸收与散射。与普通表面着色不同，体积效果并不只发生在物体表面，而是沿着相机光线在空间中持续积累。因此，体积雾可以表现远处物体逐渐被雾遮蔽、光源在空气中形成光束、点光源周围产生朦胧光晕等现象。

2 理论基础

2.1 参与介质与体渲染方程

在参与介质中，沿着一条观察光线传播的辐射亮度会受到吸收、外散射和内散射影响。设光线参数为 s ，观察方向为 ω_o ，则简化形式的辐射传输方程（Radiative Transfer Equation, RTE）可写为：

$$\frac{dL(s, \omega_o)}{ds} = -\sigma_t(s)L(s, \omega_o) + \sigma_s(s)L_i(s, \omega_o), \quad (1)$$

其中 $\sigma_t = \sigma_a + \sigma_s$ 为消光系数， σ_a 表示吸收， σ_s 表示散射。第一项表示原有光照在介质中被逐渐削弱，第二项表示其他方向的光在该点被散射进观察方向。

从相机到表面点距离为 D 时，体渲染方程的积分形式可写为：

$$L(0, \omega_o) = T(0, D)L_{\text{surface}} + \int_0^D T(0, s)\sigma_s L_i(s, \omega_o) ds, \quad (2)$$

其中 $T(0, s)$ 为相机到距离 s 处的透射率：

$$T(0, s) = \exp\left(-\int_0^s \sigma_t(u) du\right). \quad (3)$$

本项目采用均匀介质假设，因此 σ_t 为常数，透射率简化为：

$$T(0, s) = e^{-\sigma_t s}. \quad (4)$$

2.2 单次散射

完整的体积散射包含多次散射：光在介质中可能被散射多次后才进入相机。实时渲染中常用一次散射近似，即只考虑光源发出的光在某个介质点被散射一次后进入相机。对一个散射点 $\mathbf{x}$ ，入射光贡献可近似为：

$$L_i(\mathbf{x}, \omega_o) = \sum_l V_l(\mathbf{x}) L_l(\mathbf{x}) p(\omega_l, \omega_o), \quad (5)$$

其中 l 表示光源， $V_l(\mathbf{x})$ 是散射点到光源之间的可见性， $L_l(\mathbf{x})$ 是光源到达散射点的辐射亮度， p 是相函数（phase function），用于描述光从入射方向 ω_l 散射到观察方向 ω_o 的角度分布。

本项目使用 Henyey–Greenstein 相函数：

$$p(\cos \theta) = \frac{1 - g^2}{4\pi(1 + g^2 - 2g \cos \theta)^{3/2}}, \quad (6)$$

其中 $g \in [-1, 1]$ 是各向异性参数。当 $g > 0$ 时，介质更偏向前向散射，体积光束更加明显；当 $g = 0$ 时，散射是各向同性的。

2.3 Ray Marching 数值积分

由于屏幕空间后处理 Pass 中无法直接解析求解所有光源和阴影项，本项目使用 Ray Marching 进行数值积分。对每个像素，首先根据深度重建世界空间表面位置，得到从相机到表面的光线段。然后将该线段划分为 N 个小步长 Δs ，在每个采样点估计散射贡献：

$$L_{\text{fog}} \approx \sum_{i=0}^{N-1} T_i \sigma_s L_i(\mathbf{x}_i, \omega_o) \Delta s, \quad (7)$$

其中

$$T_{i+1} = T_i e^{-\sigma_t \Delta s}. \quad (8)$$

最终颜色由表面颜色和体积散射颜色混合得到：

$$L_{\text{final}} = T_N L_{\text{surface}} + L_{\text{fog}}. \quad (9)$$

3 系统设计与实现

3.1 管线位置

MoerEngine 的光栅化渲染器采用延迟渲染和后处理管线。体积雾被实现为一个新的全屏后处理 Pass，位于景深 Pass 之后、抗锯齿和 Bloom 之前。这样做的原因是：

- 输入颜色仍是 HDR 颜色，方便积累体积光照；
- 后续 Bloom 可以自然放大高亮体积光；

- 后续 Tonemapping 统一处理表面颜色和体积雾颜色；
- 实现上只需要读取当前颜色、深度、光源 Buffer 和全局光照参数。

3.2 深度重建与积分区间

体积雾 Pass 对每个像素读取当前深度。MoerEngine 使用反向 Z，因此天空像素的深度接近 0。若该像素命中场景表面，则通过 `clip2world` 矩阵重建世界坐标，并将相机到表面的距离作为积分长度。若该像素是天空，则无法得到有限表面点，因此使用用户设置的 `Max Distance` 作为积分长度，使天空区域也能显示一定的雾气和体积光。

3.3 平行光散射与可见性

平行光方向和颜色来自 MoerEngine 的 GPU 光源 Buffer。对每个体积采样点 $\mathbf{x}_i$ ，平行光方向为常量，光照贡献为：

$$L_{\text{dir}}(\mathbf{x}_i) = C_{\text{dir}} I_{\text{dir}} p(\omega_{\text{dir}}, \omega_o) V_{\text{dir}}(\mathbf{x}_i). \quad (10)$$

其中 V_{dir} 使用已有 CSM 阴影系统计算。实现中调用现有的 CSM 阴影函数，将散射点的世界坐标变换到对应级联阴影贴图空间，采样 shadow map 并返回 $[0, 1]$ 的可见性。由于介质点没有表面法线，本项目在调用现有阴影代码时使用固定向上的法线作为偏置计算的近似。

3.4 点光源散射与可见性

点光源同样来自 GPU 光源 Buffer。对散射点 $\mathbf{x}_i$ ，点光源方向和距离为：

$$\omega_l = \frac{\mathbf{x}_{\text{light}} - \mathbf{x}_i}{\|\mathbf{x}_{\text{light}} - \mathbf{x}_i\|}, \quad r = \|\mathbf{x}_{\text{light}} - \mathbf{x}_i\|. \quad (11)$$

点光源贡献使用平方反比衰减：

$$L_{\text{point}}(\mathbf{x}_i) = C_{\text{point}} I_{\text{point}} \frac{1}{4\pi r^2} p(\omega_l, \omega_o). \quad (12)$$

本项目暂不计算点光源可见性，即：

$$V_{\text{point}}(\mathbf{x}_i) = 1. \quad (13)$$

原因是点光源阴影需要 cube shadow map，并且必须保证采样到的深度和散射点深度处于同一空间。MoerEngine 当前只为主点光源维护 cube shadow map，且阴影模式和 CSM 模式在配置上互斥；如果在所有点光源和每个 ray-marching sample 上计算 cube shadow visibility，开销也会显著增加。因此，在本实现中点光源用于表现局部体积辉光，而平行光负责展示带遮挡关系的体积光束。

3.5 参数与 UI

实现提供了以下 UI 参数：

表 1: 体积雾 UI 参数说明

参数	作用	影响
Enable	开关体积雾 Pass	用于对比启用/禁用效果
Directional Visibility	是否对平行光采样 CSM 可见性	控制是否出现遮挡光束
Point Light Scattering	是否启用点光源散射	控制局部光源雾状辉光
Sample Count	每像素步进次数	越高越平滑但越慢
Density	介质密度	越高雾越浓、远处衰减越强
Max Distance	最大积分距离	控制天空和远处雾影响范围
Directional Intensity	平行光散射强度	控制太阳光束亮度
Point Intensity	点光源散射强度	控制点光源辉光亮度
Anisotropy	HG 相函数参数 g	正值增强前向散射
Fog Color	介质颜色	控制雾气色调

4 实验结果

实验场景使用 MoerEngine 自带的 Sponza 场景。该场景包含一个平行光 Sun 和多个点光源，且具有柱廊、拱门、墙面等遮挡结构，适合展示带阴影的体积光束。

4.1 启用与禁用体积雾对比

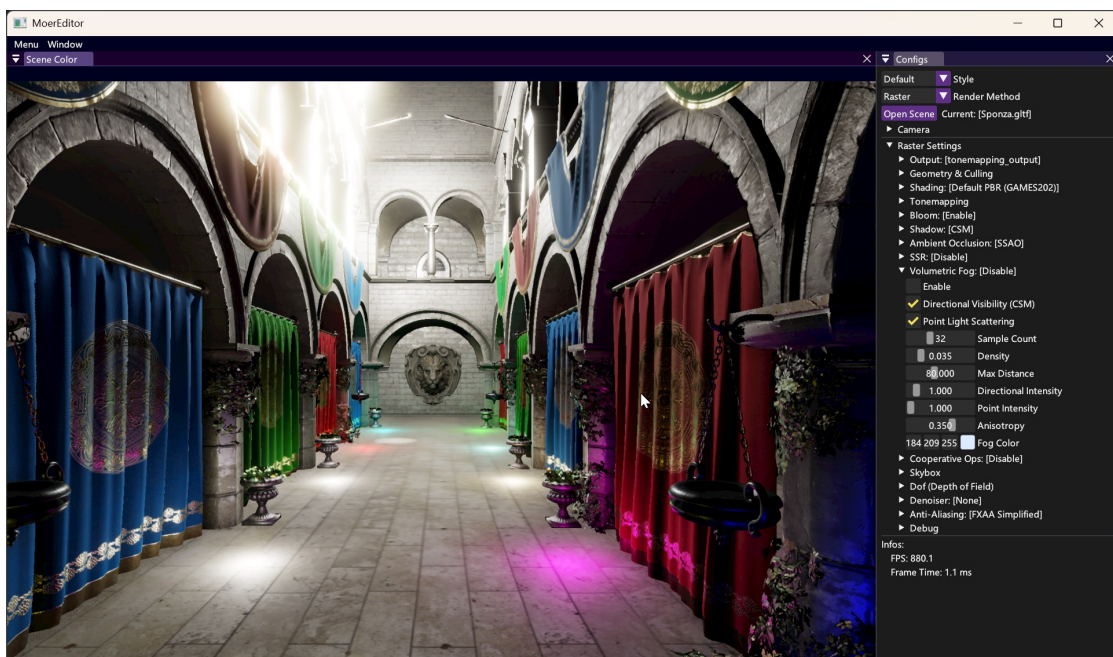


图 1: 禁用体积雾时的 Sponza 场景。

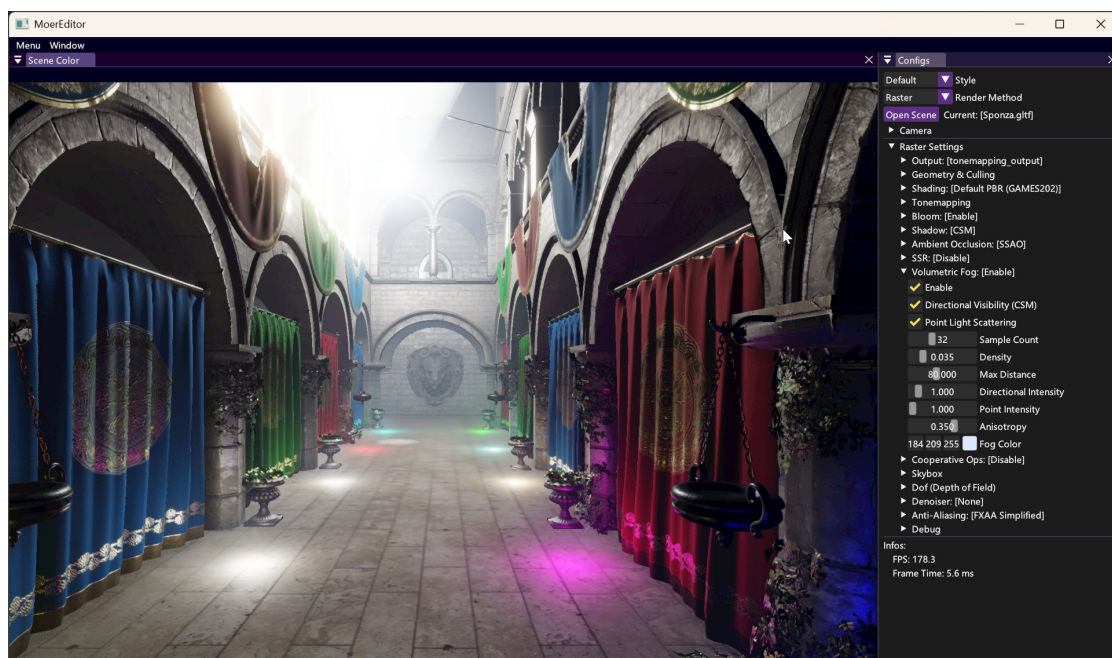


图 2: 启用体积雾后的 Sponza 场景。

4.2 平行光可见性效果

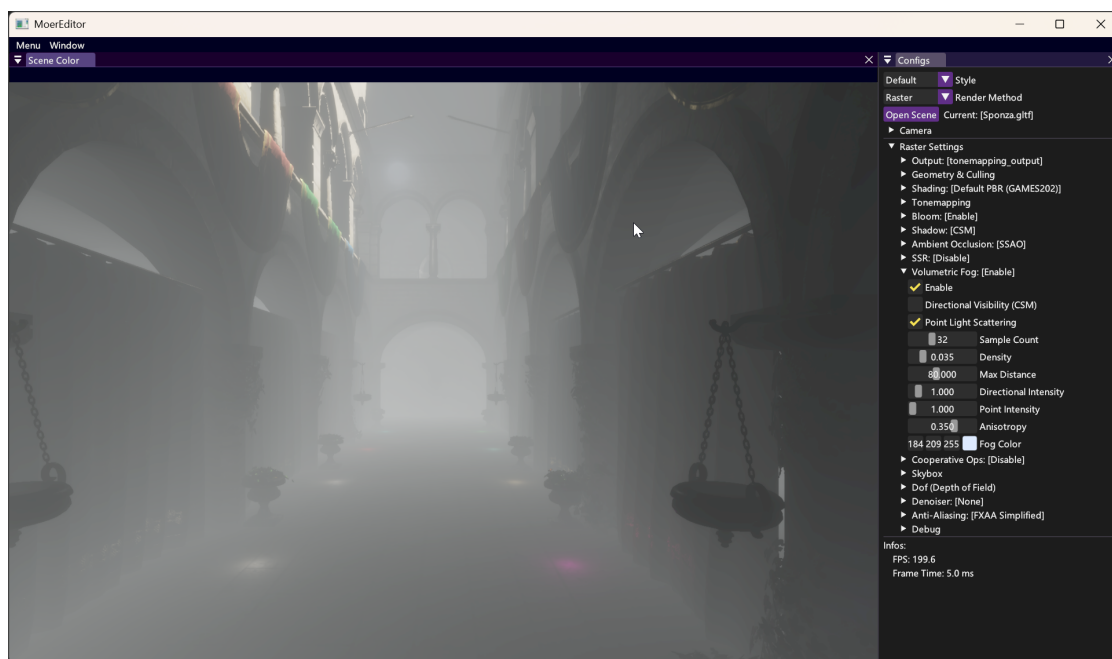


图 3: 关闭平行光可见性

4.3 点光源散射效果

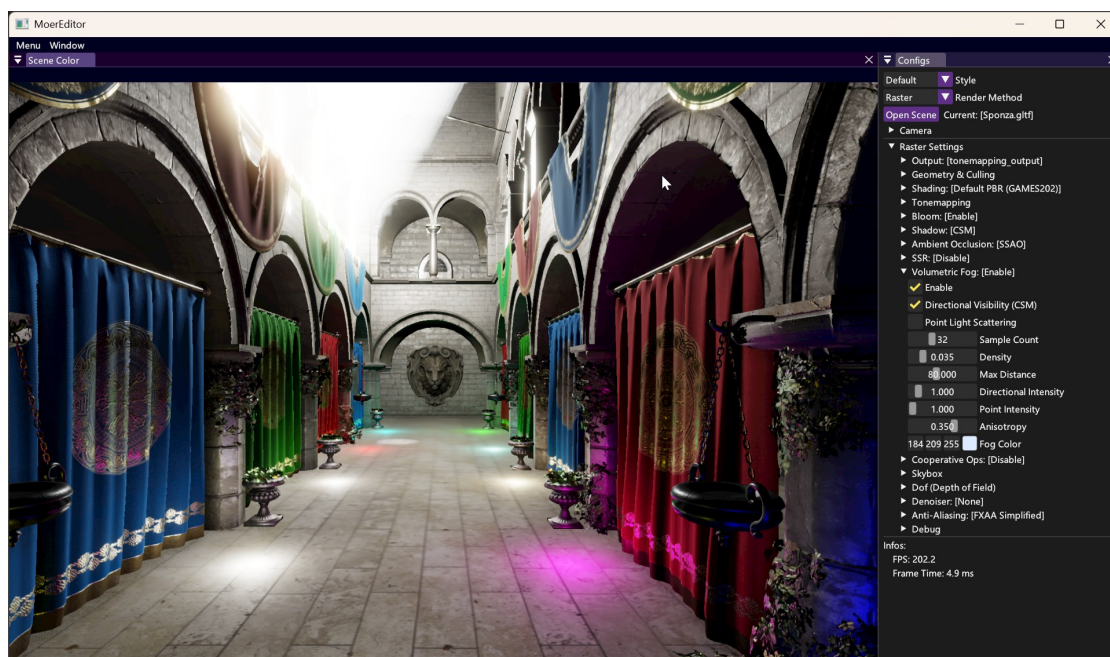


图 4: 关闭点光源对体积雾的影响

4.4 参数影响

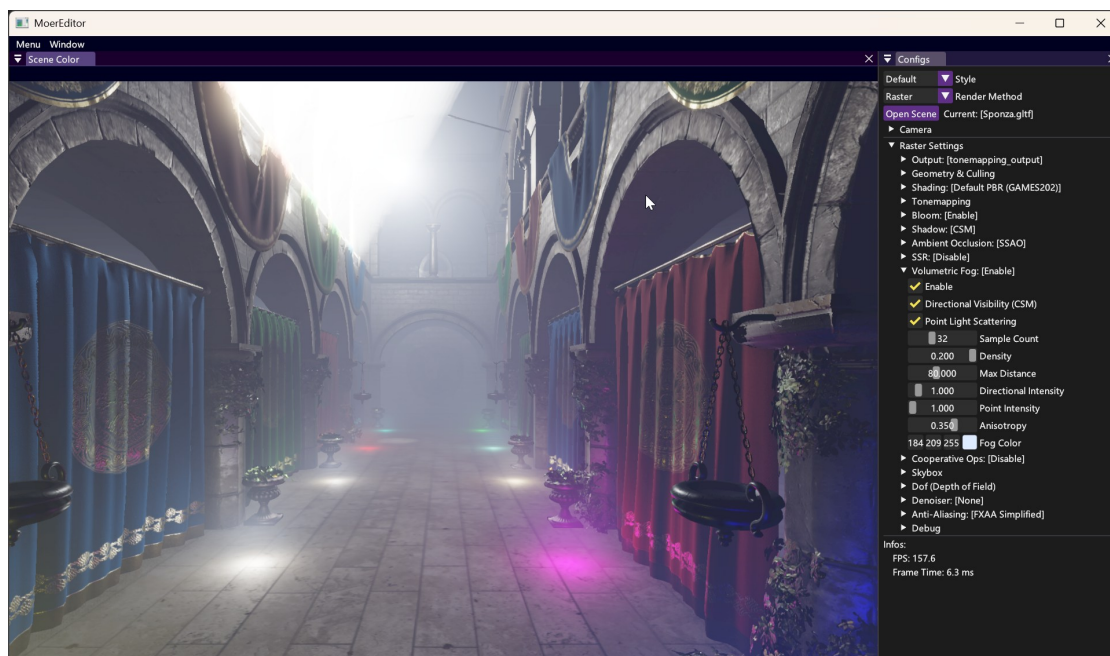
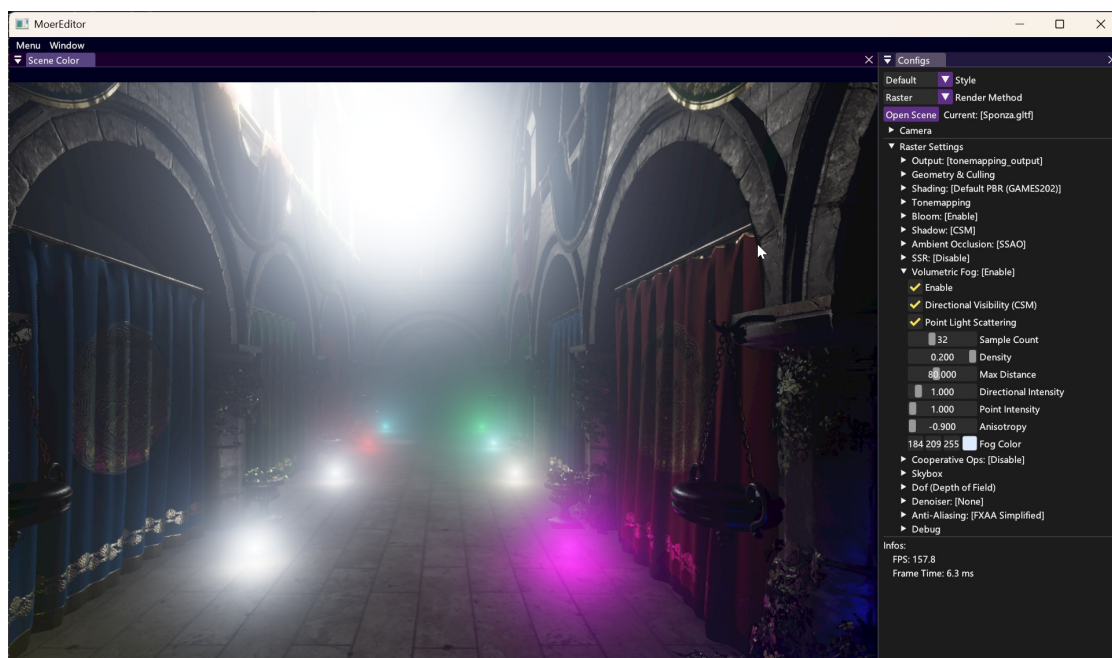
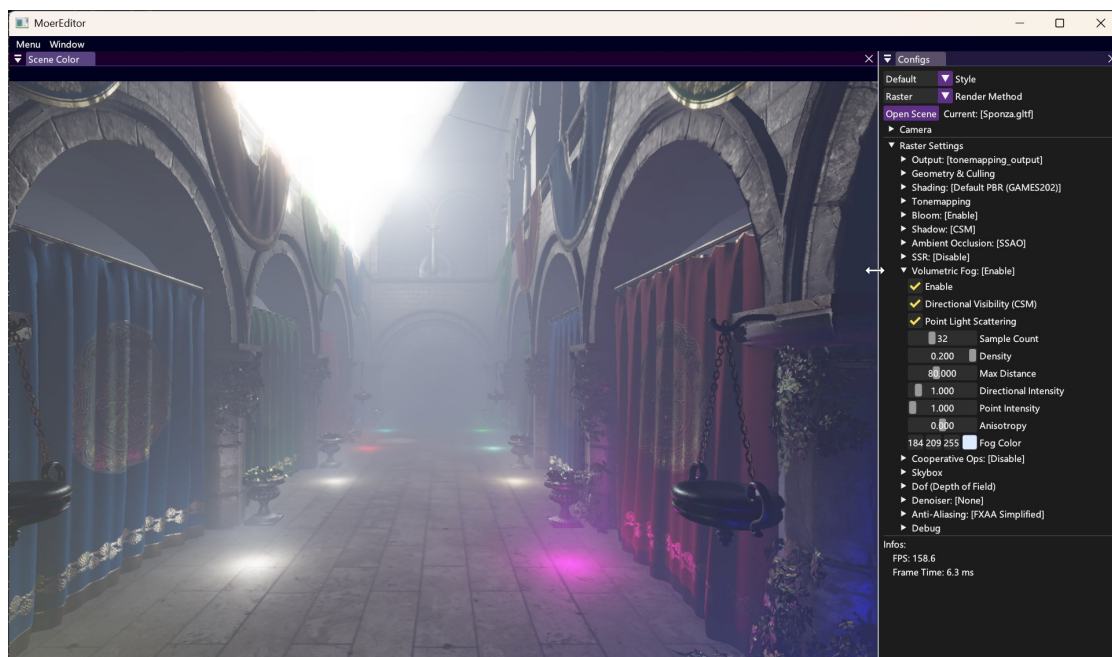


图 5: 增加体积雾浓度

图 6: HG-phase function中 $g=-0.9$ 图 7: HG-phase function中 $g=0$

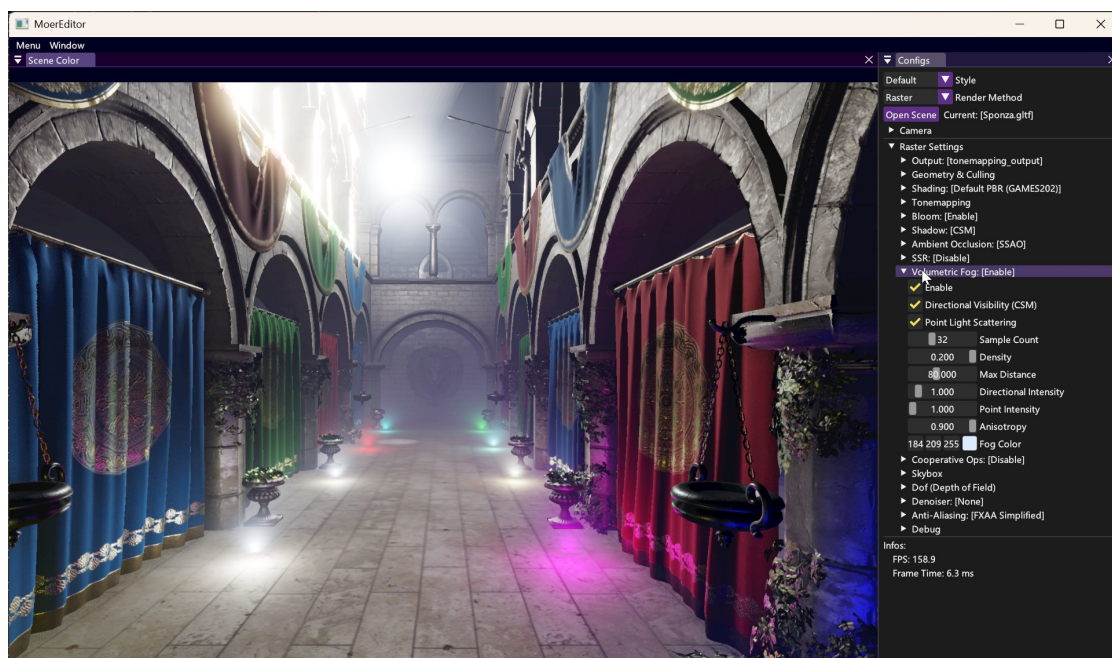
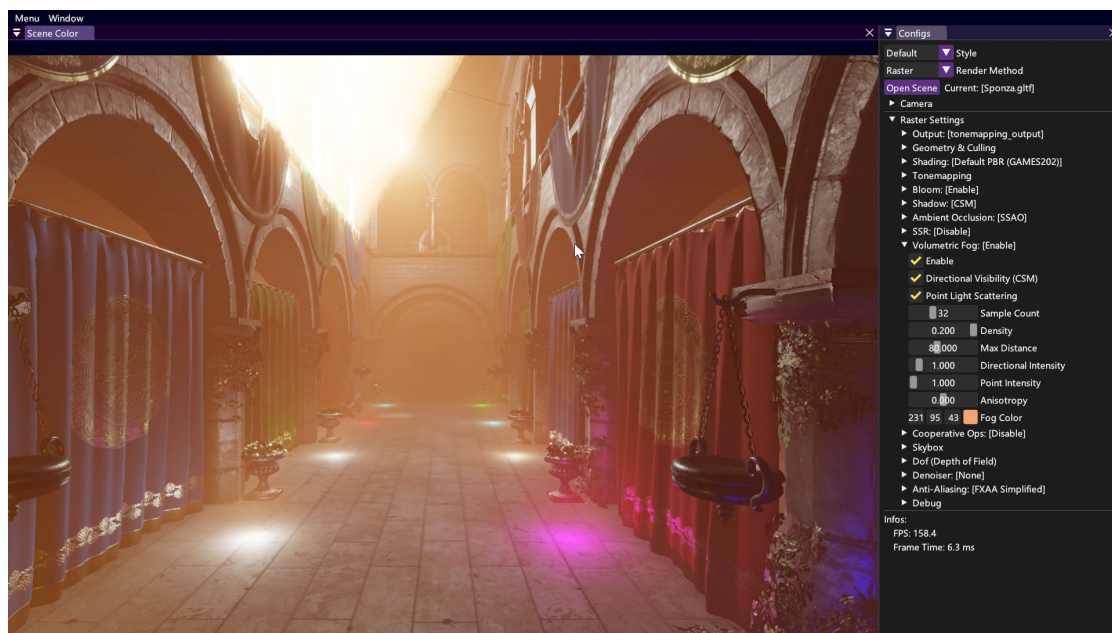
图 8: HG-phase function中 $g=0.9$ 

图 9: 红色的体积雾

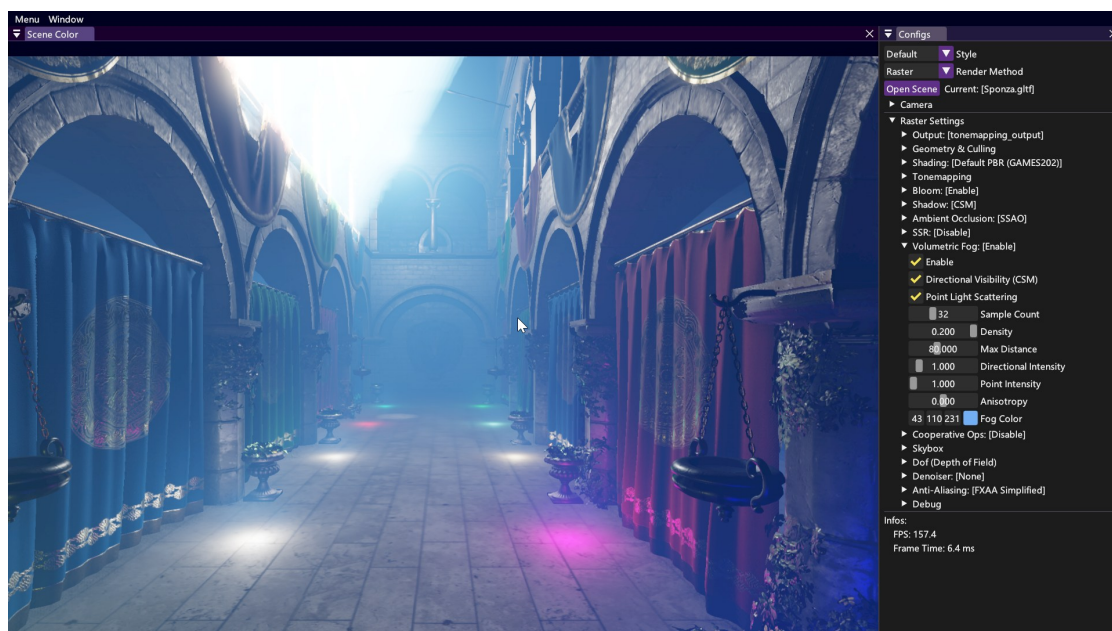


图 10: 蓝色的体积雾

5 局限性与改进方向

当前实现是一个工程上较轻量的屏幕空间体积雾方案，仍存在以下限制：

- 介质密度是全局均匀的，不能表达局部烟雾、云团或高度雾；
- Ray Marching 在全分辨率逐像素执行，采样数较高时开销明显；
- 点光源暂不计算可见性，因此会出现穿墙辉光；
- 平行光可见性复用了表面阴影函数，偏置模型并非专门为体积采样点设计；
- 没有加入时间重投影、蓝噪声抖动和双边上采样，因此低采样数下可能出现条带或噪声。

后续可以从以下方向改进：

- 使用 Foxel 或 Voxel Volume 存储体积光照，在低分辨率体积网格中积分后再上采样；
- 引入高度雾或三维噪声纹理，让密度随空间变化；
- 为点光源实现 cube shadow map 可见性，并限制参与体积雾的光源数量；
- 使用蓝噪声 jitter 和 temporal accumulation 降低采样数；