# Project Report – Advanced Computer Graphics (CS-440)

Hugo Demule (SCIPER: 339960)

August 26, 2026

## 1. PROJECT GOAL AND REFERENCE SCENE

This project extends Nori [1] with participating media, textured materials, additional light sources, and depth of field to render a bathroom scene with visible light shafts through steam. The reference scene that served as a backbone for this project is the *Japanese Bathroom* by *Abhay Pratap* found on blenderkit.com [2].

**Figure 1.** Reference Scene: *Japanese Bathroom* by *Abhay Pratap* [2]

The following sections present the different features that were implemented to achieve the project goal. Everything except scene's meshes were added manually to the project using self-created or external data.

## 2. HOMOGENEOUS AND HETEROGENEOUS MEDIUM

### 2.1. `include/nori/medium.h` (new)

This header introduces a common medium abstraction used by the scene and the integrator, similar to the *media* chapter detailed in [3]. Implemented interface:

- Optical coefficients: `sigmaA()`, `sigmaS()`, `sigmaT()`, plus optional `emission()`.
- Phase anisotropy: `g()` for Henyey-Greenstein.
- Spatial query: `sigmaAt(p)` (constant by default, overridden for heterogeneous media).
- Sampling support: `majorant()` for tracking methods.

Utility methods centralize common volumetric formulas:

- Albedo (channel-wise): $\alpha = \frac{\sigma_s}{\sigma_t}$
- Homogeneous transmittance: $T_r(d) = \exp(-\sigma_t d)$
- Henyey-Greenstein phase function: $p(\cos\theta) = \frac{1}{4\pi}\frac{1-g^2}{(1+g^2-2g\cos\theta)^{3/2}}$
- Inverse-CDF phase sampling for HG (including isotropic fallback for $|g| \approx 0$).

Two concrete classes are declared:

- `HomogeneousMedium` with constant coefficients.
- `HeterogeneousMedium` backed by a density grid loaded from disk.

### 2.2. `src/homogeneous.cpp` (new, unused)

Implements `HomogeneousMedium` construction and validation:

- Reads `sigma_a`, `sigma_s`, `emission`, and `phase_g` from XML.
- Enforces physically valid ranges (non-negative coefficients, $|g| < 1$).
- Precomputes $\sigma_t = \sigma_a + \sigma_s$.

This file provides the robust base case used for smoke/steam experiments where density is spatially uniform.

### 2.3. `src/heterogeneous.cpp` (new)

Implements `HeterogeneousMedium` with a binary voxel grid.
Data loading and preprocessing:

1. Read optical parameters (`sigma_a`, `sigma_s`, `phase_g`, `density_scale`
2. Resolve and open a binary file.
3. Parse dimensions $n_x, n_y, n_z$ and world-space AABB.
4. Read density samples in x-major order.
5. Compute a global majorant: $\hat{\sigma}_t = \max_c(\sigma_{t,c}) \cdot \max(\rho)$.

The majorant must upper-bound the extinction coefficient everywhere in the medium. Taking the maximum extinction coefficient across color channels and multiplying it by the maximum density stored in the grid guarantees that this condition is satisfied for every point in the volume.
Runtime density query:

- Convert world point to normalized grid coordinates.
- Reject points outside the medium AABB.
- Perform trilinear interpolation over the 8 neighboring voxels.
- Return scaled density and derive local extinction with: $\sigma_t(\mathbf{x}) = \rho(\mathbf{x})\,\sigma_t$.

This directly enables localized steam concentrations and smooth transitions in the shower region.

> **Note**
>
> a supplementary python program `vdb_to_bin.py` have been created to allow converting external vdb files into bin files, easier to work with.

## 3. DISTANT LIGHT VOLUMETRIC PATH TRACER WITH MULTIPLE IMPORTANCE SAMPLING (MIS) AND NEXT EVENT ESTIMATION (NEE)

### 3.1. `src/volpath_distant_mis.cpp` (new)

This file implements the integrator `volpath_distant_mis`, designed to reduce volumetric noise in participating media under directional lighting, while keeping a structure that is straightforward to extend to non-distant emitters. An important note on this integrator is that a single distance proposal in the medium (e.g., only exponential/majorant sampling) produced high variance and visible fireflies in steam regions (see Figure 2). Even with a distant light, the single-scattering integrand along a camera segment remains highly non-uniform because it is shaped by:

- heterogeneous density $\sigma_t(\mathbf{x})$,
- transmittance from camera to sample and from sample to light,
- binary visibility/occlusion to the light.

Using two complementary sampling techniques (truncated exponential + equi-angular [4]) with MIS gives better coverage of the segment and lower variance in practice (see Figure 3).

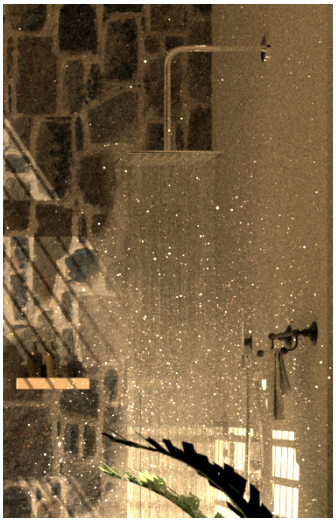

**Figure 2.** Render using only truncated exponential sampling, 256 samples per pixel (SPP).

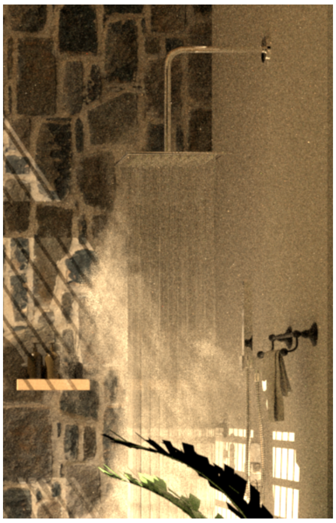

**Figure 3.** Render using multiple importance sampling from exponential and equi-angular sampling, 256 SPP.

#### 3.1.1. *Main Integrator Flow*

1. Intersect current ray with the scene to get `tSurface`.
2. If medium exists and `tSurface` is finite, estimate single-scatter direct lighting over $[0, tSurface]$ with `estimateSingleScatterDistantMIS(...)`.
3. Add medium emission over the same segment with `mediumEmissionIntegral(...)`.
4. Sample whether a real medium collision occurs before the surface with `deltaTracking(...)`.
5. If a medium collision occurs: sample phase direction (`samplePhase`), update throughput by albedo, and continue.
6. Otherwise, process surface terms: emitted radiance, direct distant light at surface, BSDF sampling, then continue.
7. Apply Russian roulette after a few bounces.

#### 3.1.2. *Radiative Transfer Equation (RTE).*

$$\frac{dL_i(\mathbf{x},\omega)}{dz} = \underbrace{\mu_s(\mathbf{x})\,L_s(\mathbf{x},\omega)}_{\text{In-scattering}} + \underbrace{\mu_a(\mathbf{x})\,L_e(\mathbf{x},\omega)}_{\text{Emission}} - \underbrace{\mu_t(\mathbf{x})\,L_i(\mathbf{x},\omega)}_{\text{Extinction}}$$

#### 3.1.3. *Where each RTE term is computed in code.*

- **In-scattering** $\mu_s(\mathbf{x})\,L_s(\mathbf{x},\omega)$: evaluated by the function `estimateSingleScatterDistantMIS(...)` that performs multiple importance sampling (see 3.1.4) with next event estimation. The per-sample contribution is assembled in `evaluateDistantLineSample(...)` from: `sigmaSAt(...)`, phase term `medium->phase(...)`, light radiance, and transmittance factors.
- **Emission** $\mu_a(\mathbf{x})\,L_e(\mathbf{x},\omega)$ (course notation): accumulated in `Li` via `radiance += throughput * mediumEmissionIntegral(...)`. The segment integral is computed in `mediumEmissionIntegral(...)`.
- **Extinction** $-\mu_t(\mathbf{x})\,L_i(\mathbf{x},\omega)$: not added as a standalone `radiance += ...` term. It is applied implicitly through transport attenuation and event statistics:
  - transmittance estimators in `ratioTrackingTr(...)`, used for `TrCam` and `TrLight`,
  - free-flight survival/collision sampling in `deltaTracking(...)` with majorant-based acceptance.

#### 3.1.4. *Distance MIS details.*

- Truncated exponential proposal on $[a, b]$:

$$p_{\text{dist}}(t) = \frac{\sigma\, e^{-\sigma(t-a)}}{1 - e^{-\sigma(b-a)}}.$$

- Equi-angular proposal on $[a, b]$:

$$p_{\text{equi}}(t) = \frac{D}{(\theta_b - \theta_a)\left(D^2 + (t - t_p)^2\right)}.$$

- Power-heuristic MIS combination:

$$w_i = \frac{p_i^2}{\sum_j p_j^2}.$$

#### 3.1.5. *Justification of equi-angular sampling here.*

Although the current implementation only samples a distant emitter, equi-angular distance sampling is still useful as a complementary proposal to majorant-exponential sampling. The two proposals fail in different regions of the segment; MIS mitigates extreme $f(t)/p(t)$ events and therefore reduces fireflies. This proposal-based structure is also naturally extensible: the same estimator layout can be reused with point/spot/area lights by changing the light-query geometry while keeping the same MIS structure.
Regarding the heuristic used, stable results were also obtained with the balance heuristic. However, the power heuristic was retained because low-PDF outliers are more aggressively downweighted by it, which provides a better robustness against rare high-variance spikes in general.

### 3.2. Validation Tests

#### 3.2.1. `test-medium-absorption.xml` *(new)*

Statistical t-test validating Beer-Lambert attenuation in a homogeneous, purely absorbing medium. This test validates medium transmittance estimation along camera-to-emitter paths. Because the expected transmittance is known analytically from the Beer-Lambert law, agreement between rendered and analytical values provides evidence that the absorption/transmittance implementation is unbiased in this setting. Scenes vary path length and/or absorption coefficient, with references matching

$$T = \exp(-\sigma_a d)$$

(including equivalent $\sigma_a d$ pairs that should produce identical transmittance).

```
Generating 100000 paths..
Sample mean = 0.36479 (reference value = 0.367879)
Sample variance = 0.231721
t-statistic = 2.02955 (d.o.f. = 99999)
Accepted the null hypothesis (p-value = 0.0424053, significance level = 0.00250943)

Passed 4/4 tests.
```

**Figure 4.** Medium Absorption Test

#### 3.2.2. `test-medium-emission.xml` *(new)*

This test verifies that radiance generated by medium emission along a camera ray matches the analytic single-segment solution:

$$L_e^{\text{int}} = \int_0^d e^{-\sigma_t t} L_e\, dt = \frac{L_e\left(1 - e^{-\sigma_t d}\right)}{\sigma_t}, \qquad \sigma_t = \sigma_a + \sigma_s.$$

In this setup, $(\sigma_s = 0)$, so $(\sigma_t = \sigma_a)$. The references also include the zero-extinction edge case, interpreted as the limit $(\sigma_t \to 0)$, which reduces to:

$$L_e^{\text{int}} = L_e d$$

```
Generating 100000 paths..
Sample mean = 2 (reference value = 2)
Sample variance = 0
t-statistic = 0 (d.o.f. = 99999)
Accepted the null hypothesis (p-value = 1, significance level = 0.00334451)

Passed 3/3 tests.
```

**Figure 5.** Medium Emission Test

#### 3.2.3. `test-medium-hetero-constant.xml` (new)

t-test validating that heterogeneous infrastructure reproduces homogeneous behavior when density is constant.
Using a constant-density `2x2x2` grid and the same optical coefficients, the measured attenuation matches homogeneous references, confirming correctness of:

- grid loading,
- trilinear lookup,
- density scaling,
- tracking estimators under spatial media API.

```
Generating 100000 paths..
Sample mean = 0.36672 (reference value = 0.367879)
Sample variance = 0.232239
t-statistic = 0.760825 (d.o.f. = 99999)
Accepted the null hypothesis (p-value = 0.446763, significance level = 0.00501256)

Passed 2/2 tests.
```

**Figure 6.** Heterogeneous Medium Test

#### 3.2.4. `test-medium-furnace-noscatter.xml` (new)

Statistical t-test validating that the zero-scattering limit preserves radiance along the entire segment.
With $\sigma_a = \sigma_s = 0$, no attenuation, absorption, or in-scattering should be introduced by the medium, so the furnace reference remains constant at 1 for all tested path lengths. This checks the vacuum limit of both the homogeneous and heterogeneous code paths and confirms that the medium does not alter energy when extinction is zero.

```
Generating 100000 paths..
Sample mean = 1 (reference value = 1)
Sample variance = 0
t-statistic = 0 (d.o.f. = 99999)
Accepted the null hypothesis (p-value = 1, significance level = 0.00250943)

Passed 4/4 tests.
```

**Figure 7.** Furnace, No-Scatter Medium Test

#### 3.2.5. `test-medium-scattering-distant.xml` (new)

t-test validating non-zero single-scattering behavior under a side distant light source.
A weak but non-zero scattering medium is used so that the measured signal is dominated by the medium phase function and transmittance along the ray. The references are given by the single-scatter estimator The reference values are obtained from the analytical single-scattering solution under the assumptions of a homogeneous medium, isotropic phase function, and an unoccluded distant emitter. The probability of observing a scattering event before leaving a medium segment of length $d$ is

$$1 - e^{-\sigma_t d}.$$

Because isotropic scattering distributes energy uniformly over the sphere, only a fraction $1/(4\pi)$ contributes toward the viewing direction, yielding

$$L = L_e \frac{1}{4\pi} \frac{\sigma_s}{\sigma_t} (1 - e^{-\sigma_t d}).$$

which verifies that in-scattering is being accumulated correctly in both the homogeneous and heterogeneous constant-density cases.

```
Generating 100000 paths..
Sample mean = 0.0195694 (reference value = 0.01944)
Sample variance = 0.000428641
t-statistic = 1.97649 (d.o.f. = 99999)
Accepted the null hypothesis (p-value = 0.0481019, significance level = 0.00250943)

Passed 4/4 tests.
```

**Figure 8.** Non-Zero Single Scattering Test

## 4. TEXTURE HANDLING AND TEXTURED BSDF

### 4.1. `include/nori/texture.h` (new)

Defines `Texture` as a new Nori object type (`EClassType::ETexture`) with one essential operation:

- `eval(uv)` returning `Color3f`.

This abstraction allows materials to consume either constant colors or image-driven parameters through the same `addChild` pattern as other Nori components.

### 4.2. `src/imagetexture.cpp` (new)

Implements `ImageTexture`:

1. Reads image filename and options (`srgb`, `scale`).
2. Loads pixels via `stb_image`.
3. Stores RGB texels internally.
4. Evaluates with repeating UVs (`frac` behavior).
5. Flips the V coordinate to match common OBJ/Blender UV conventions [5].
6. Converts from sRGB to linear RGB when requested.

sRGB conversion uses the standard piecewise transfer [6]:

$$L = \begin{cases} \frac{s}{12.92}, & s \le 0.04045 \\ \left(\frac{s+0.055}{1.055}\right)^{2.4}, & s > 0.04045 \end{cases}$$

**Note**

converting texture colors from sRGB to linear space is usually required as in practice, most diffuse PNG/JPG textures found online are stored in sRGB (non-linear) rather than linear format. These textures are converted to linear RGB before participating in shading computations. Performing lighting calculations directly in sRGB space produces incorrect energy distributions and noticeably darker renders.

### 4.3. `src/diffuse_tex.cpp` (new)

Adds `diffuse_tex`, a Lambertian BSDF whose albedo is texture-driven:

- BRDF evaluation: $f_r = \frac{\rho(\mathbf{u},\mathbf{v})}{\pi}$
- PDF and sampling remain cosine-weighted hemisphere.
- If no child texture is attached, a fallback color is used.

This preserves all diffuse sampling behavior from base Nori while enabling detailed spatial variation from UV textures.

### 4.4. `src/microfacet_tex.cpp` (new)

Adds `microfacet_tex`, extending the microfacet BRDF with textured diffuse albedo `kd`.
Per shading sample:

1. Read $kd = texture(uv)$ (or fallback).
2. Set specular share: $k_s = 1 - \max(k_d)$.
3. Evaluate mixed model:
   - diffuse term $k_d/\pi$,
   - Beckmann microfacet specular with Fresnel and Smith-style masking-shadowing.

Sampling/PDF are mixture-based (diffuse or Beckmann lobe chosen with `ks`), then weighted consistently for unbiased estimation.
Together with `imagetexture.cpp`, this file is what makes bathroom assets (e.g. wooden floor) visually plausible.

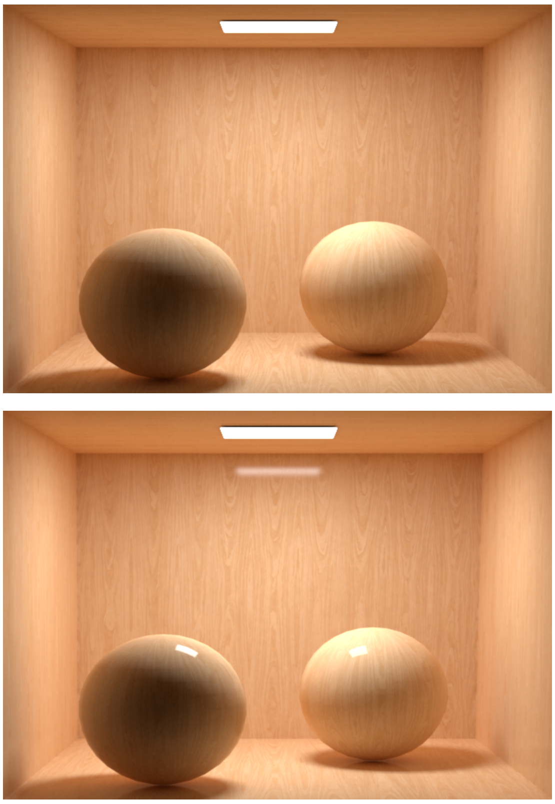

**Figure 9.** Comparison between diffuse wood texture (top) and microfacet with $\alpha = 0.01$ (bottom). Theses images match the expected behaviour align with their corresponding untextured materials.

## 5. DIRECTIONAL AND SPOT LIGHTS

### 5.1. `src/distant.cpp` (new)

Implements a directional emitter (`distant`) used as a sun-like source [7].
Characteristics:

- Constant incoming direction and radiance everywhere.
- Distant sampling API returns a deterministic direction with unit PDF.
- Surface-emitter methods (`sample/pdf/eval` on mesh points) return zero since this is not an area light.

This is directly consumed by `volpath_distant_mis.cpp` for both medium and surface direct-light terms.

### 5.2. `src/spot.cpp` (new, unused)

Implements an angularly-modulated area emitter (`spot`) attached to mesh geometry [8].
Algorithm:

1. Uniformly sample a point on emitter mesh area.
2. Build `EmitterQueryRecord` and convert PDF to requested measure (`EArea` or `ESolidAngle`).
3. Compute angular attenuation based on emission angle:
   - full intensity inside beam width,
   - smooth power falloff up to cutoff,
   - zero beyond cutoff.

With inner/outer cosine bounds:

$$f(\theta) = \begin{cases} 1, & \cos\theta \geq \cos\theta_{\text{inner}} \\ \left(\frac{\cos\theta - \cos\theta_{\text{outer}}}{\cos\theta_{\text{inner}} - \cos\theta_{\text{outer}}}\right)^n, & \cos\theta_{\text{outer}} < \cos\theta < \cos\theta_{\text{inner}} \\ 0, & \cos\theta \leq \cos\theta_{\text{outer}} \end{cases}$$

This is useful for controlled practical light sources in the interior setup. This emitter was introduced to provide a controllable directional light source that better highlights volumetric scattering effects than uniform area emitters. However, this emitter was discarded to only use distant light in the final scene.

## 6. CAMERA DEPTH OF FIELD

### 6.1. `src/perspective.cpp` (modified)

`PerspectiveCamera` is extended with thin-lens depth of field while preserving the pinhole mode.
New parameters:

- `apertureRadius`
- `focusDistance`

Ray generation logic:

1. Compute pinhole direction through sampled pixel.
2. If aperture radius is zero: keep original behavior.
3. Else:
   - intersect pinhole ray with focus plane at distance `focusDistance`,
   - sample lens point on a disk (`Warp::squareToUniformDisk`),
   - shoot ray from lens point to focus point.

This produces physically meaningful blur for out-of-focus geometry while keeping in-focus surfaces sharp.

**Validation.** The thin-lens camera model was validated by rendering scenes with objects placed at different depths relative to the focal plane (see Figure 10). Objects located at the focus distance remain sharp, while objects located in front of or behind the focal plane become increasingly blurred, as expected from the thin-lens model.

**Note**

In the final shot, a foreground plant is seen through a background mirror. With this basic thin-lens depth-of-field model, focus is decided from direct camera depth along primary rays. This is too simple for mirror reflections: the reflected plant is treated at a different effective depth than the mirror surface itself.
As a result, even when the mirror region is in focus, the reflected plant can still appear out of focus, because the camera model does not account for the reflected optical path the way we artistically expect in this composition.
Therefore, a mask hack has been added to enforce one-sided blur behavior and keep the reflected content readable. This is an intentional artistic workaround, not a physically accurate camera model.

(a) Glass in focus, High aperture

(b) Glass in focus, Low aperture

(c) No depth of field

(d) Bowl in focus, Low aperture

(e) Bowl in focus, High aperture

**Figure 10.** Validation of the thin-lens depth-of-field implementation. The center image corresponds to a pinhole camera with no depth of field. Moving away from the center, changing the focus distance shifts the sharp region from the foreground glass to the background bowl, while increasing the aperture produces a stronger blur for out-of-focus objects. These results match the expected behavior of the thin-lens camera model.

## 7. EXTRA MATERIAL

### 7.1. `src/conductor.cpp` (new, unused)

Implements a simplified Cook–Torrance rough conductor microfacet BSDF inspired by the PBRT conductor model [9].
Key aspects:

- No diffuse term.
- Colored conductor Fresnel from complex IOR $\eta, k$ per RGB channel.
- Beckmann microfacet normal distribution with geometric masking-shadowing attenuation.

The implementation uses the Beckmann distribution instead of the Trowbridge–Reitz (GGX) distribution used in PBRT, since Beckmann importance sampling and evaluation had already been implemented in previous assignments and integrates naturally with the existing Nori microfacet framework.
BRDF form:

$$f_r(\omega_i, \omega_o) = \frac{D(\omega_h)\, G(\omega_i, \omega_o)\, F_c(\omega_i \cdot \omega_h)}{4|\cos\theta_i||\cos\theta_o|}$$

Sampling follows microfacet normal sampling and reflection mapping, with the corresponding Jacobian-adjusted PDF.

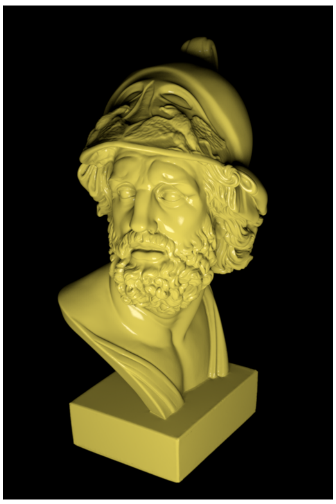

**Figure 11.** Yellow microfacet material using $\alpha = 0.03$, intIOR $= 1.5$, $kd = (0.6, 0.5, 0.08)$ and 2048 SPP.

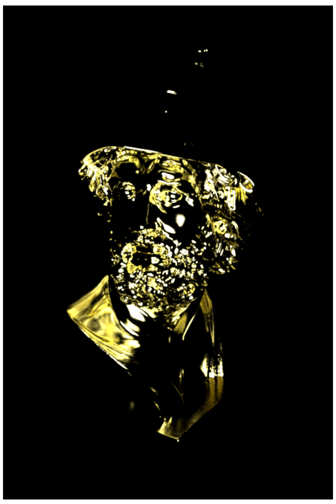

**Figure 12.** Gold conductor material using $\alpha = 0.03$, $\eta = (0.17, 0.35, 1.50)$, $k = (3.14, 2.98, 1.94)$ and 2048 SPP.

## 8. LLM STATEMENT

Large Language Models (LLMs), especially *Github Copilot* [10], was used throughout this project for the following tasks:

1. **Code Completion**: When starting to define functions or any other code block, *GitHub Copilot* helps speed up development by completing the most likely code to be written next. This tool is a great way to be more efficient while still maintaining a fine-grained understanding of the code.
2. **Code Verification**: After implementing an entire source file, human errors often remain. The *GitHub Copilot* chat feature makes it possible to ask for potential issues that could make the code physically incorrect, for example inconsistencies in mathematical formulas or implementation details.
3. **File Conversion Script**: The script used to convert `.vdb` files into `.bin` files was generated using *GitHub Copilot*. This made it possible to focus on the core aspects of the project rather than spending time on auxiliary tasks.
4. **Refactoring**: *GitHub Copilot* is also a great tool for refactoring existing code by suggesting more specialized and reusable functions. This was especially useful in `volpath_distant_mis.cpp`, where the integrator can quickly become verbose and difficult to read.

All generated suggestions were manually reviewed, tested, and modified when necessary.

## 9. FINAL RENDER

The following picture (Figure 13) shows the final render of the scene. The video showing steps of the process can be found using the following link: https://www.youtube.com/watch?v=UHACSvas2-A.

The image can be downloaded here: https://raw.githubusercontent.com/HugoDemRey/AdvancedComputerGraphicFinalRender/27136ebdfd68f190406375055d246c07c597fb49/scene_focus_sky.png

**Figure 13.** Final Render, 4096 SPP.

## REFERENCES


[1] W. Jakob, *Nori: An educational ray tracer*, https://wjakob.github.io/nori/, Accessed: 2026-05-21, 2024.

[2] A. PRATAP, *Japanese bathroom*, https://www.blenderkit.com/asset-gallery-detail/4e817ddf-d50b-45f3-a276-e595f6e005e7/, ArtStation: https://www.artstation.com/arsenal-05, accessed 2026-05-19.

[3] W. J. Matt Pharr and G. Humphreys, *Media*, https://www.pbr-book.org/4ed/Volume_Scattering/Media, In *Physically Based Rendering: From Theory to Implementation*, 4th ed. Accessed: 2026-05-19.

[4] C. Kulla and M. Fajardo, *Importance sampling techniques for path tracing in participating media. cgf (proc. of eurographics symposium on rendering) 31, 4 (june 2012), 1519–1528*, 2012.

[5] Wikipedia contributors, *Wavefront .obj file*, https://en.wikipedia.org/wiki/Wavefront_.obj_file, Accessed: 2026-05-22, 2026.

[6] International Electrotechnical Commission, *IEC 61966-2-1:1999 – sRGB Colour Space Specification*, https://www.color.org/sRGB.pdf, Accessed: 2026-05-22, 1999.

[7] W. J. Matt Pharr and G. Humphreys, *Distant lights*, https://pbr-book.org/4ed/Light_Sources/Distant_Lights, In *Physically Based Rendering: From Theory to Implementation*, 4th ed. Accessed: 2026-05-19.

[8] Oracle, *SpotLight (JavaFX 25 API Documentation)*, https://download.java.net/java/GA/javafx25/docs/api/javafx.graphics/javafx/scene/SpotLight.html, Accessed: 2026-05-22, 2025.

[9] W. J. Matt Pharr and G. Humphreys, *Conductor brdf*, https://pbr-book.org/4ed/Reflection_Models/Conductor_BRDF, In *Physically Based Rendering: From Theory to Implementation*, 4th ed. Accessed: 2026-05-19.

[10] GitHub, *Github copilot*, https://github.com/features/copilot, Accessed: 2026-05-29, 2025.


### External Assets

1. **Steam VDB Volume**: Grenade Dust Impact found at https://jangafx.com/software/embergen/download/free-vdb-animations
2. **Textures**:
   - Light Wood: https://polyhaven.com/a/oak_veneer_01
   - Rocky Wall: https://texturelabs.org/textures/brick_123/
   
   *All other textures were self-made (e.g. simple colours) or found in the reference scene [2].*