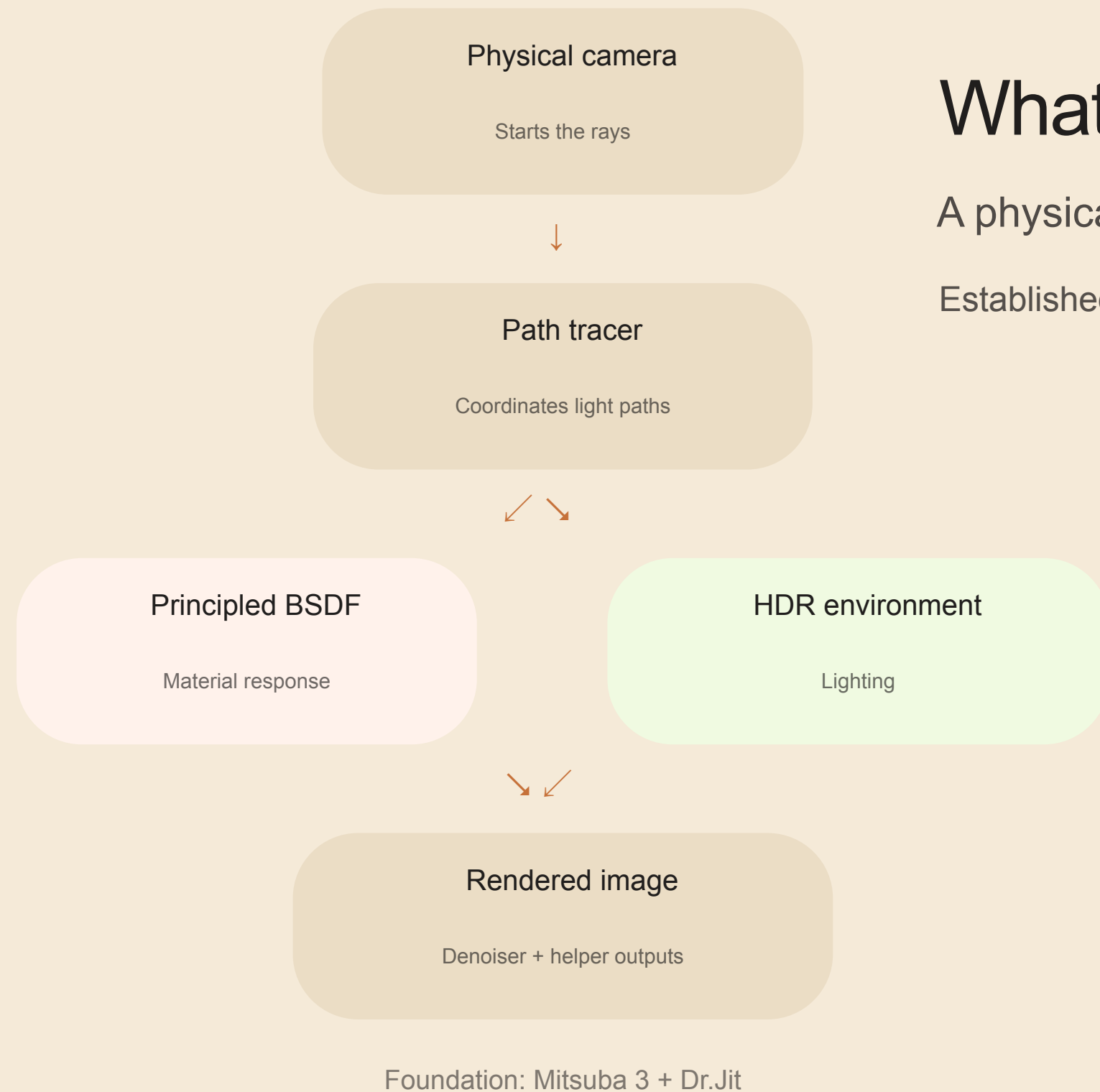


Implementing and Validating Physically Based Rendering Techniques in an Extensible Renderer

<https://github.com/Gabrielazmbr/Physically-Based-Renderer.git>



What was developed?

A physically based renderer built as custom Mitsuba plugins.

Established rendering techniques were implemented, integrated and validated as one complete pipeline.

- Mitsuba infrastructure + Python execution through Dr.Jit
- Camera, path tracer, materials and HDR lighting integrated
- Validation from mathematical tests to production-scale scenes
- Correctness, noise and visual quality evaluated
- End-to-end comparison with Blender Cycles

Mitsuba acts as the workshop, while the custom plugins form the specialised rendering team operating inside it.

Four questions guiding the project

1 — Correctness

Can established rendering techniques be implemented correctly as custom Mitsuba plugins?

2 — Sampling and noise

Which techniques reduce noise, and what accuracy or quality trade-offs do they introduce?

3 — Real-scene behaviour

What problems appear when moving from simple tests to realistic, production-scale scenes?

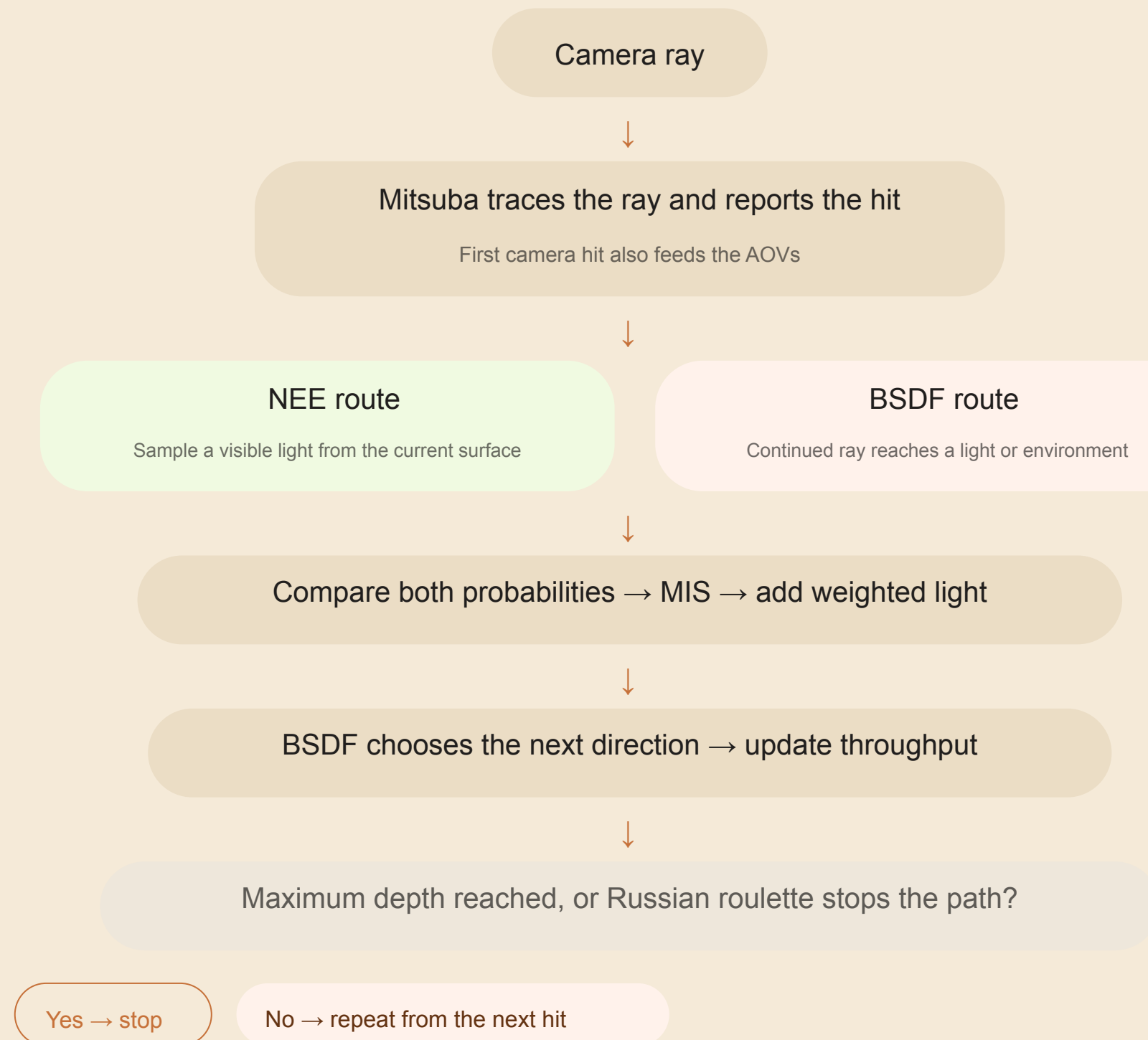
4 — Cycles comparison

How closely can the completed renderer reproduce Blender Cycles, and can the remaining differences be explained?

The first component required to answer these questions was the path tracer, which coordinates the complete light-transport process.

The path tracer coordinates every light bounce

At each surface, it collects available light, chooses the next direction and decides whether the path should continue.



Key terms

Throughput

The remaining light weight carried by the path.

NEE

Explicitly selects a light and checks whether it is visible.

BSDF sampling

Asks the material to choose the next ray direction.

MIS

Balances light sampling and material sampling without overcounting.

Maximum depth

Places a fixed limit on the number of bounces.

Russian roulette

Probabilistically stops weak paths while preserving the expected average.

What Mitsuba provides

The path tracer is a path coordinator: Mitsuba provides the map and reports collisions, while the path tracer decides where to search for light, which path to follow and when the path ends.

Mitsuba provides

Scene and geometry information
Accelerated ray–geometry intersections
Standard BSDF and emitter interfaces
Random samplers and safe ray creation
Film storage and image reconstruction

The custom path tracer controls

Path continuation and accumulated light
NEE and MIS weighting
Termination
Optional output channels and visibility behaviour

Important optional features — disabled unless requested

First-hit AOVs

Albedo, shading normal and depth.

Transparent shadows

Allow supported glass surfaces to pass direct light.

Firefly clamping

Limits extreme samples but introduces bias.

Camera visibility controls

Hide selected geometry from primary rays.

The Principled BSDF combines several material behaviours

Material controls change which light behaviours are active and how likely each one is to be selected.

Evaluation, probability and sampling must agree

sample() **chooses** a direction, pdf() **reports** how likely that direction was to be chosen, and eval() **calculates** how much light the material sends in that direction.

eval()

Returns the combined material response in a direction.

pdf()

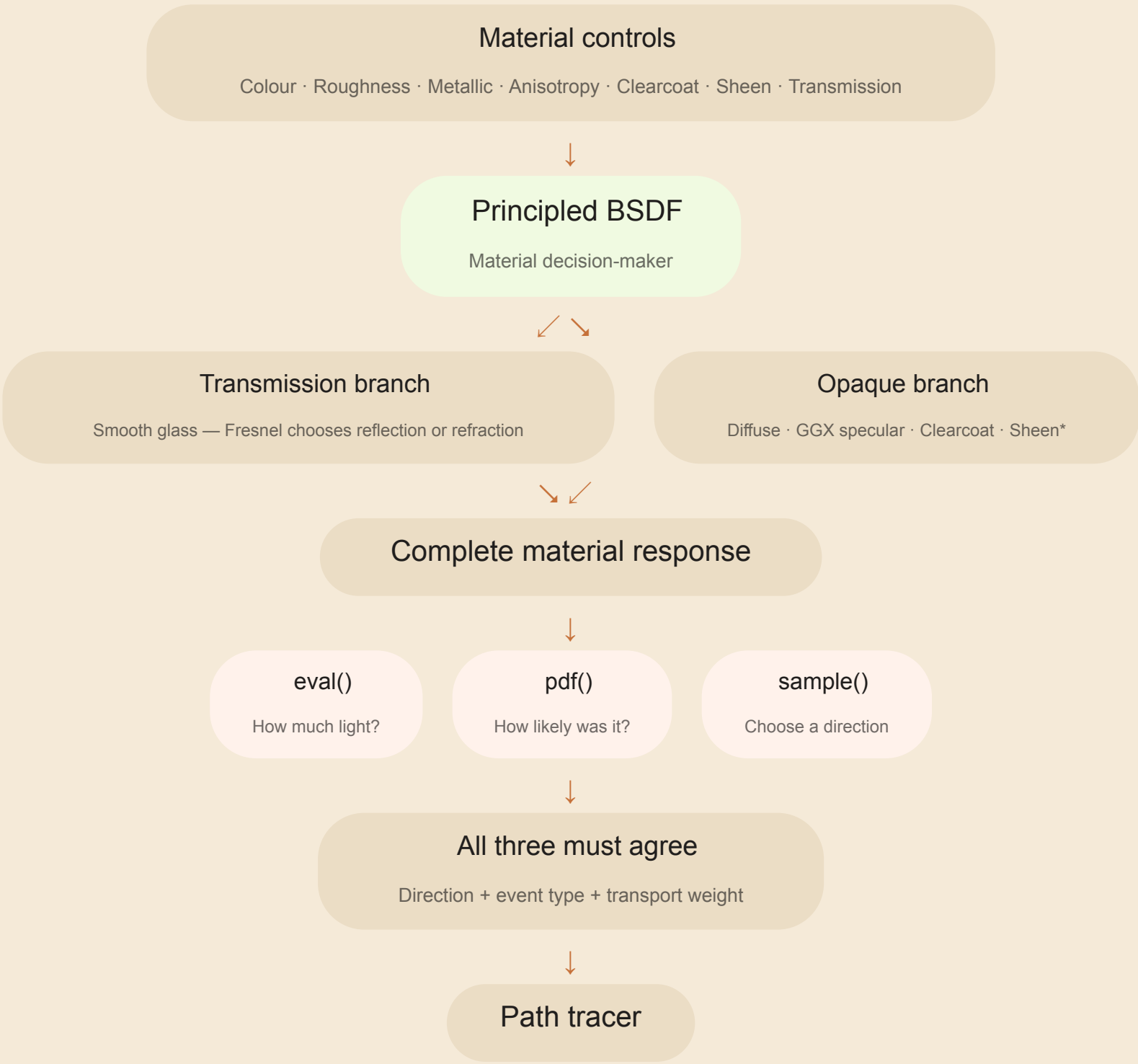
Reports the complete probability of selecting that direction.

sample()

Selects a direction from one of the active behaviours.

Sample weight

Complete BSDF value divided by the complete mixture PDF.



*Sheen contributes to the result but is not sampled separately.

Material behaviours and what is supported

Diffuse — spreads light broadly, producing a matte appearance

GGX reflection — models rough surfaces as collections of microscopic mirrors

Metallic — colours the reflected light and removes conventional diffuse reflection

Anisotropy — stretches the GGX highlight along a surface direction

Clearcoat — adds a narrow secondary highlight above the main response

Sheen — adds a soft grazing-angle response, mainly around silhouettes

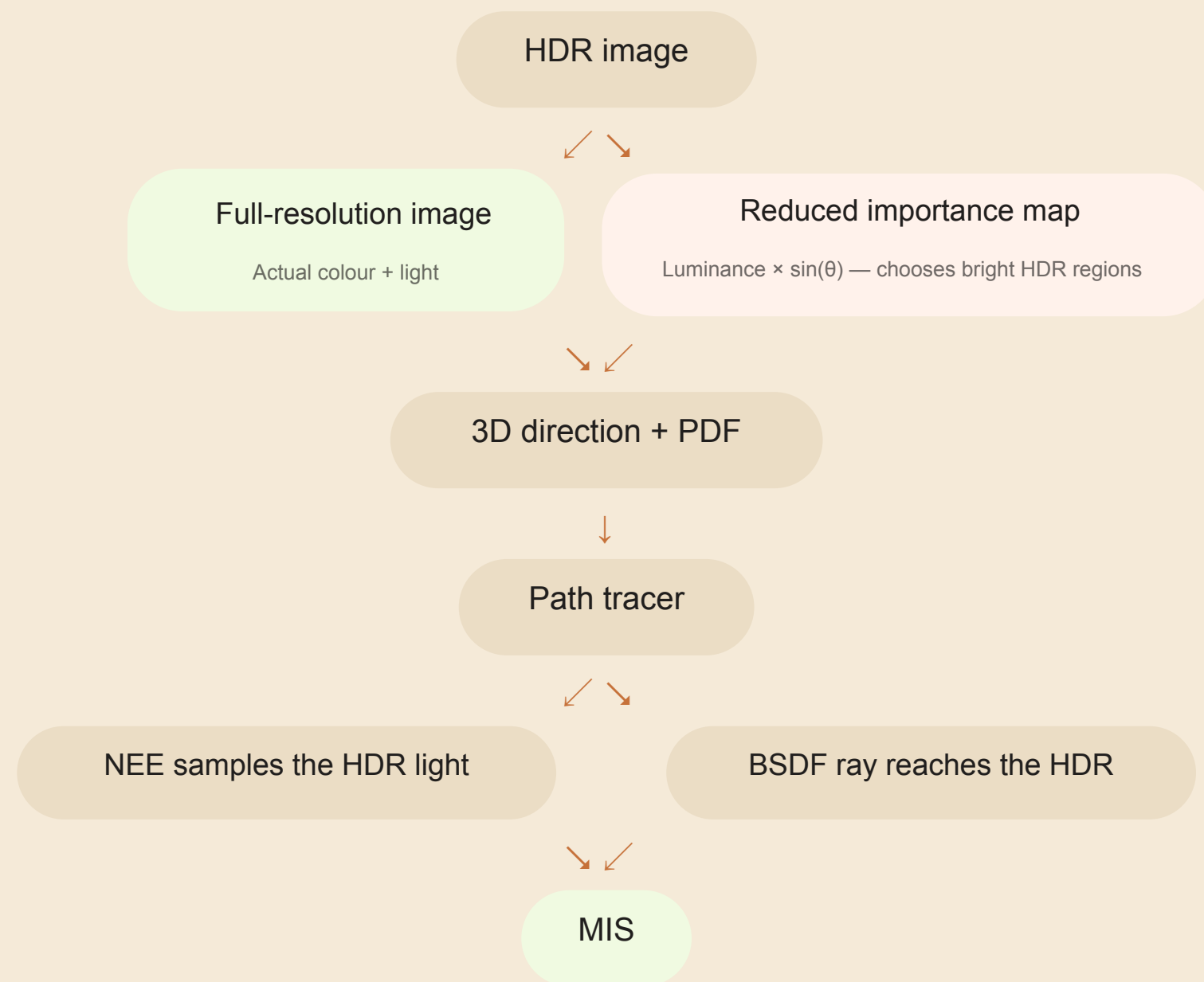
Transmission — uses Fresnel to choose smooth reflection or refraction

Supported

Lambert and Burley diffuse
Dielectric and metallic GGX reflection
Anisotropic highlights
GTR1 clearcoat
Tinted sheen
Smooth solid-glass transmission
Thin-sheet transmission
Texture-driven colour and roughness

The HDR environment is used in two ways

The full HDR image answers "**how much light is here?**", while the smaller importance map answers "**where should samples be taken?**" The selected image position is converted into a spherical direction. **NEE** then checks that direction from the current surface, while the **BSDF** route may reach the same environment independently. **MIS** combines the two routes.



Key points

Radiance image — provides the actual HDR colour and intensity

Importance map — decides where samples should be concentrated

Direction mapping — converts a flat HDR pixel into a direction on the surrounding sphere

Probability — $\sin(\theta)$ corrects map distortion so the emitter PDF can be compared with the BSDF PDF

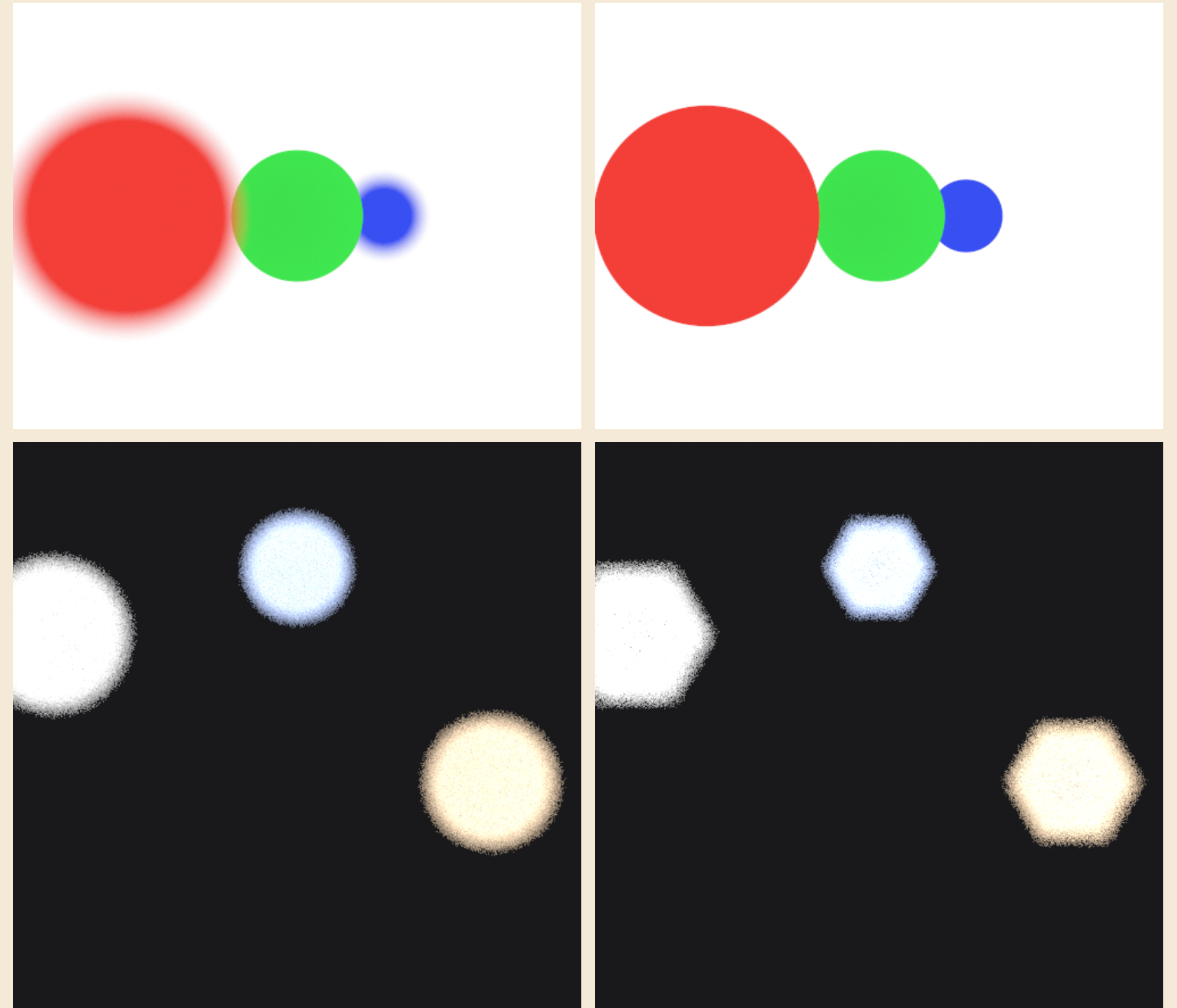
The path tracer performs NEE and MIS. The environment emitter only chooses an HDR direction and returns its light and PDF.

Physical thin-lens camera

The implemented camera models focus and aperture geometry, but aperture size changes blur rather than exposure.

- Lens rays meet at the **focal plane**
- Focal plane stays **sharp**
- Near and far objects **become blurred**
- Larger aperture = shallower depth of field
- Aperture **shape** controls **bokeh**
- Aperture **rotation** controls bokeh **orientation**
- Zero aperture = pinhole camera

Testing:



- At zero aperture, the projection closely **matched** Mitsuba's camera.
- A finite aperture kept the chosen focal distance sharp and **blurred nearer and farther objects**.
- Circular and **six-blade apertures** produced matching circular and **hexagonal bokeh**.

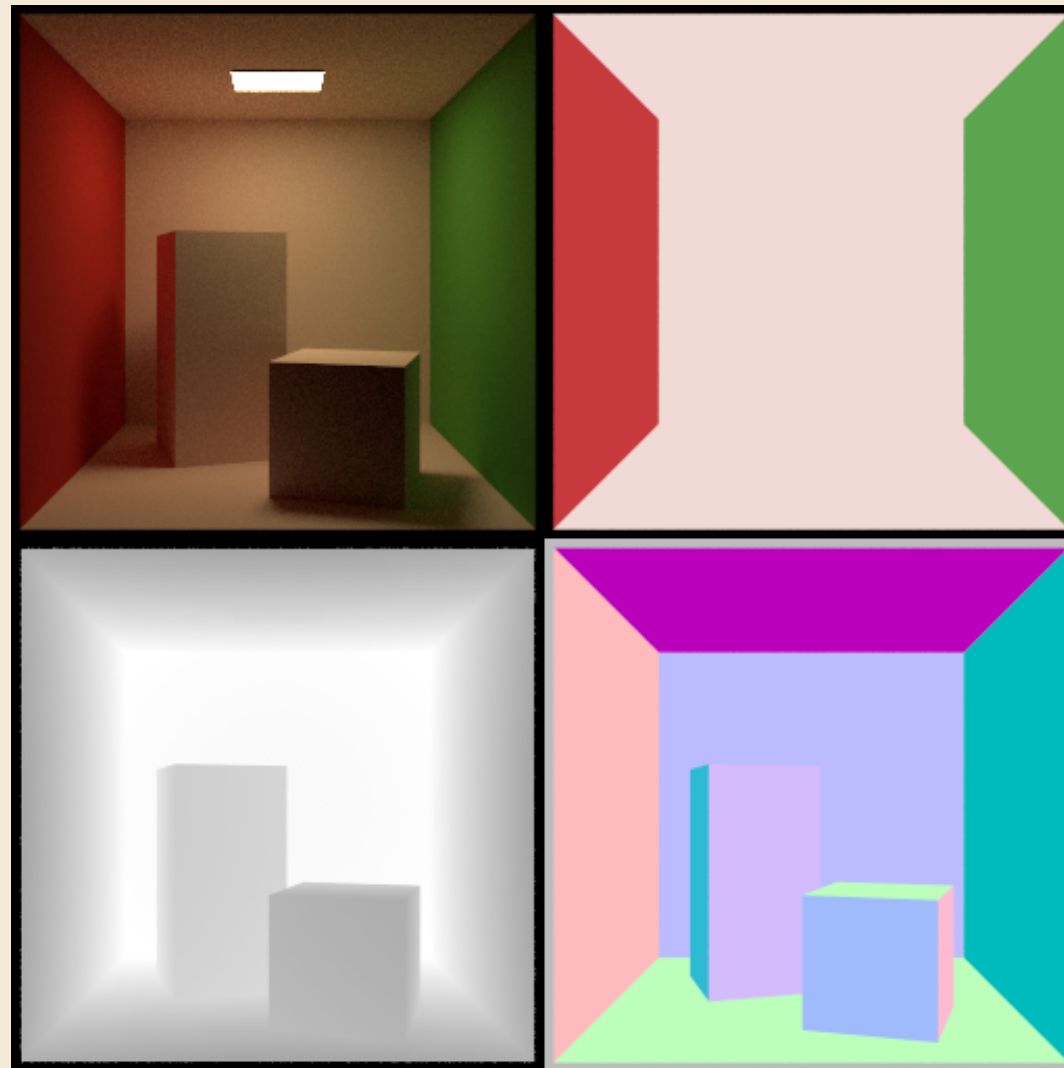
From camera focus to denoised output

First-hit AOVs

Albedo — basic surface colour without lighting

Normal — direction the visible surface faces

Depth — distance from the camera



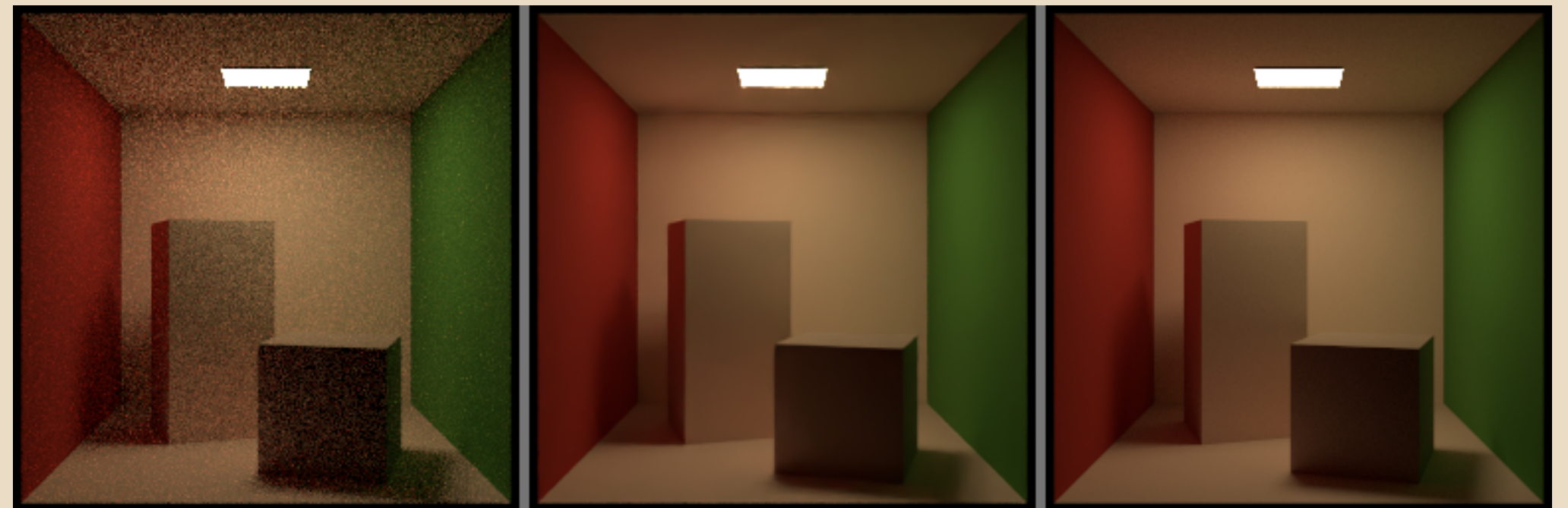
OIDN denoising

OIDN processes the completed beauty image using **albedo** and **normal** guidance.

Depth remains **diagnostic** and is not passed to OIDN.

Denoising **improves presentation** but does not prove transport correctness.

Can the completed image be reconstructed more cleanly?



At 16 spp, OIDN reduced error by 64.2%.

Its benefit decreased as the original render became cleaner.

Denoising improves presentation but does not prove correctness.

Important boundary — the camera and AOVs participate in rendering. OIDN operates afterward as a separate reconstruction step.

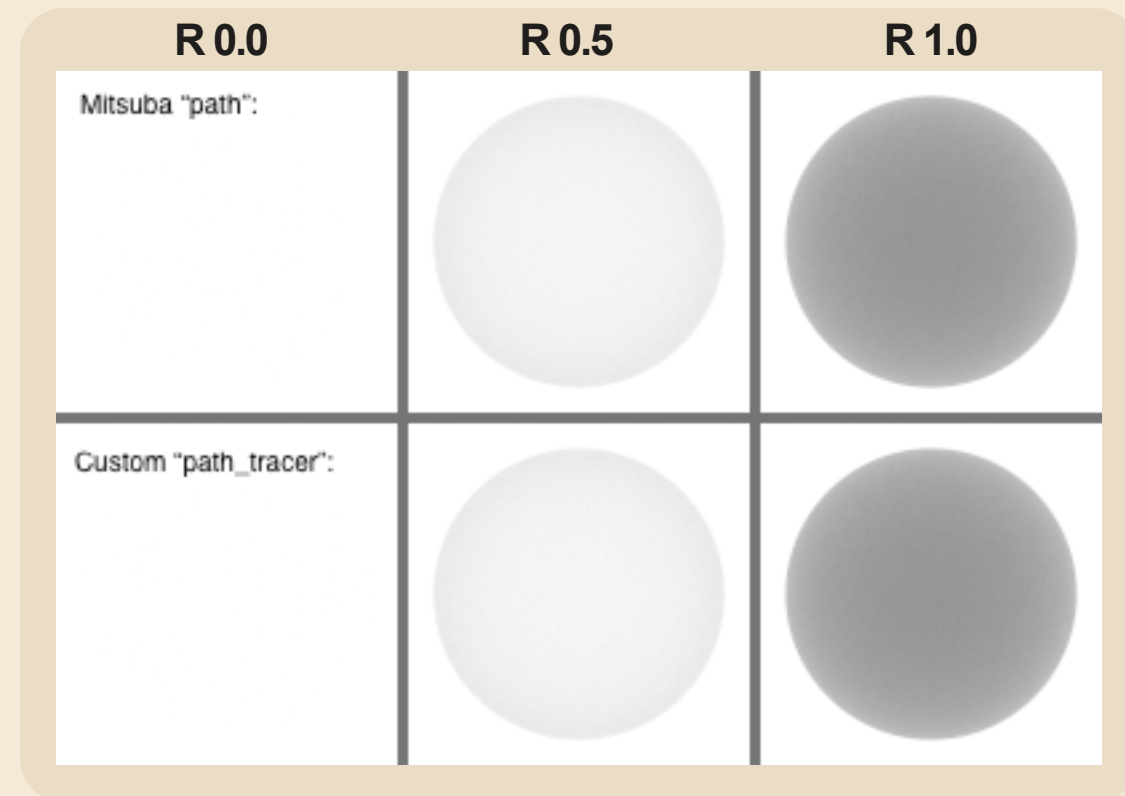
Validation added one difficulty at a time

Each stage answered a different question before adding more complexity.

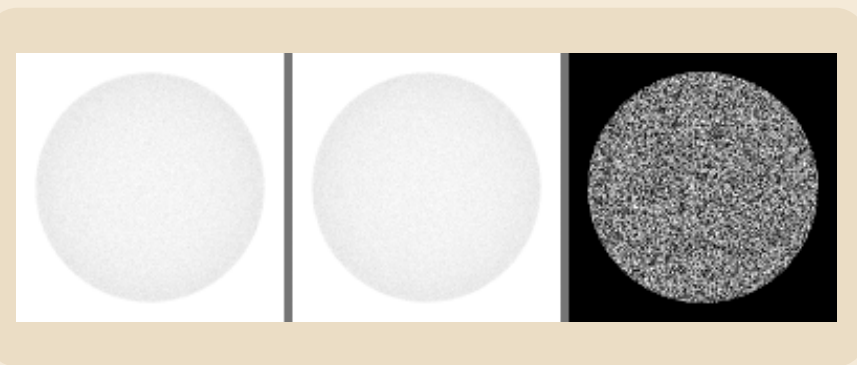


The path tracer matched known and reference behaviour

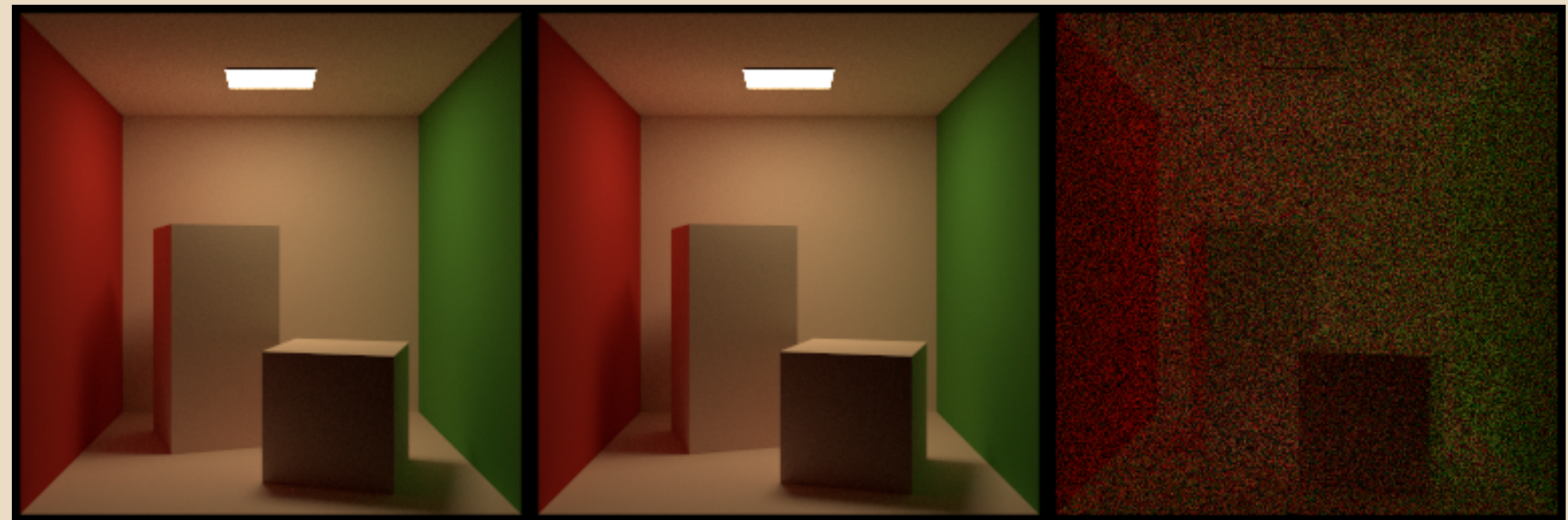
The scene, materials, camera and settings remained fixed; only the integrator changed.



- The custom and Mitsuba integrators **produced the same result**.
- Increasing samples from **256 to 1024** approximately **halved the noise**.
- **Rough GGX also matched Mitsuba**, showing that its energy loss came from the shared material model.



Cornell box — multi-bounce reference



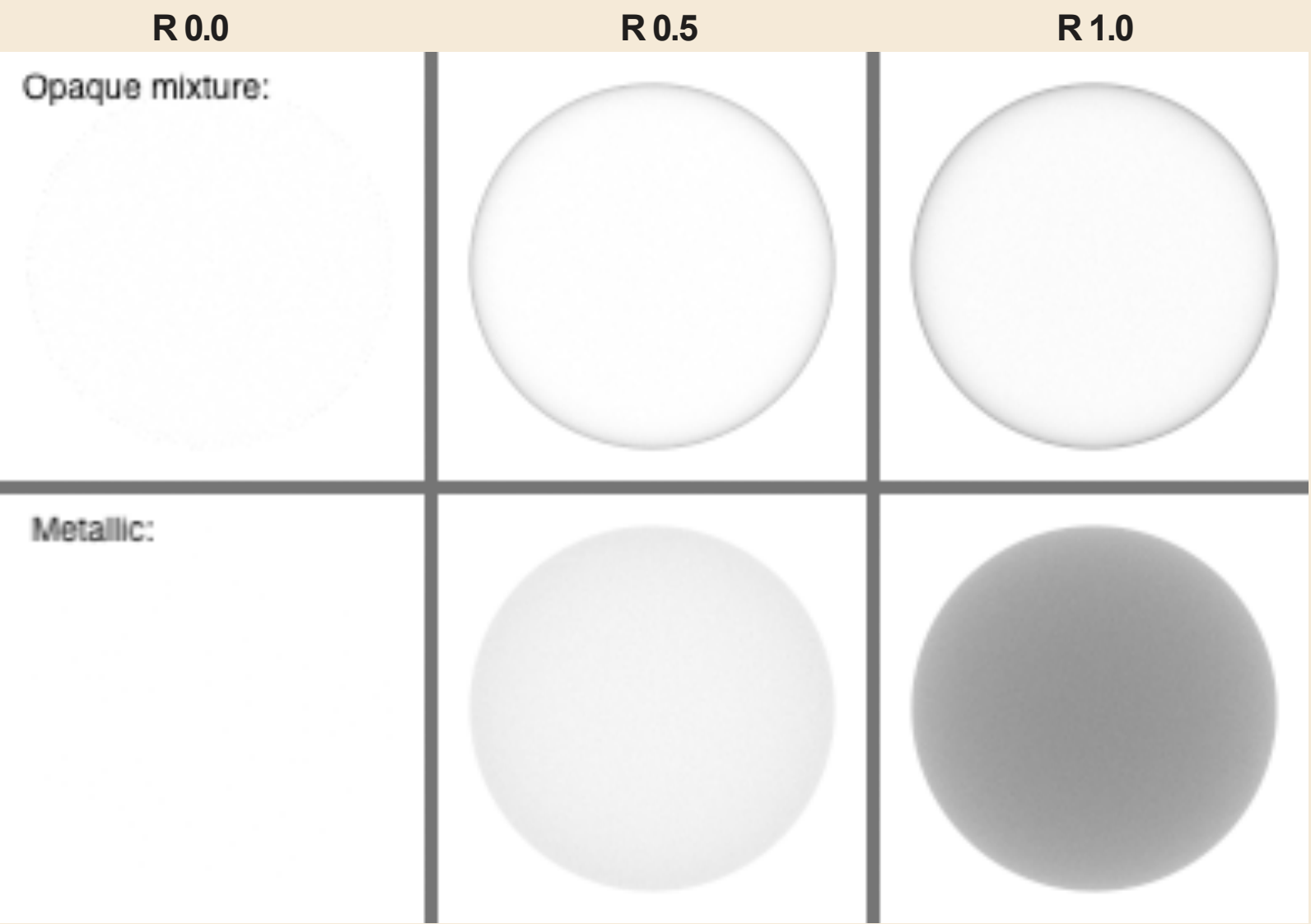
Added shadows, occlusion, colour bleeding and indirect light.
Mean radiance differed by only -0.0035% .
Pixel MAE: 0.000856.

The furnace verified basic weighting and convergence; the Cornell box verified complete multi-bounce transport.

Principled BSDF, tests check

1 — White furnace

Does the material create or lose light?



Lambert returned the expected value of one.
Rough GGX darkening **matched** Mitsuba, identifying a shared model limitation rather than an implementation error.

2 — Chi-squared test

Does *sample()* do what *pdf()* claims?

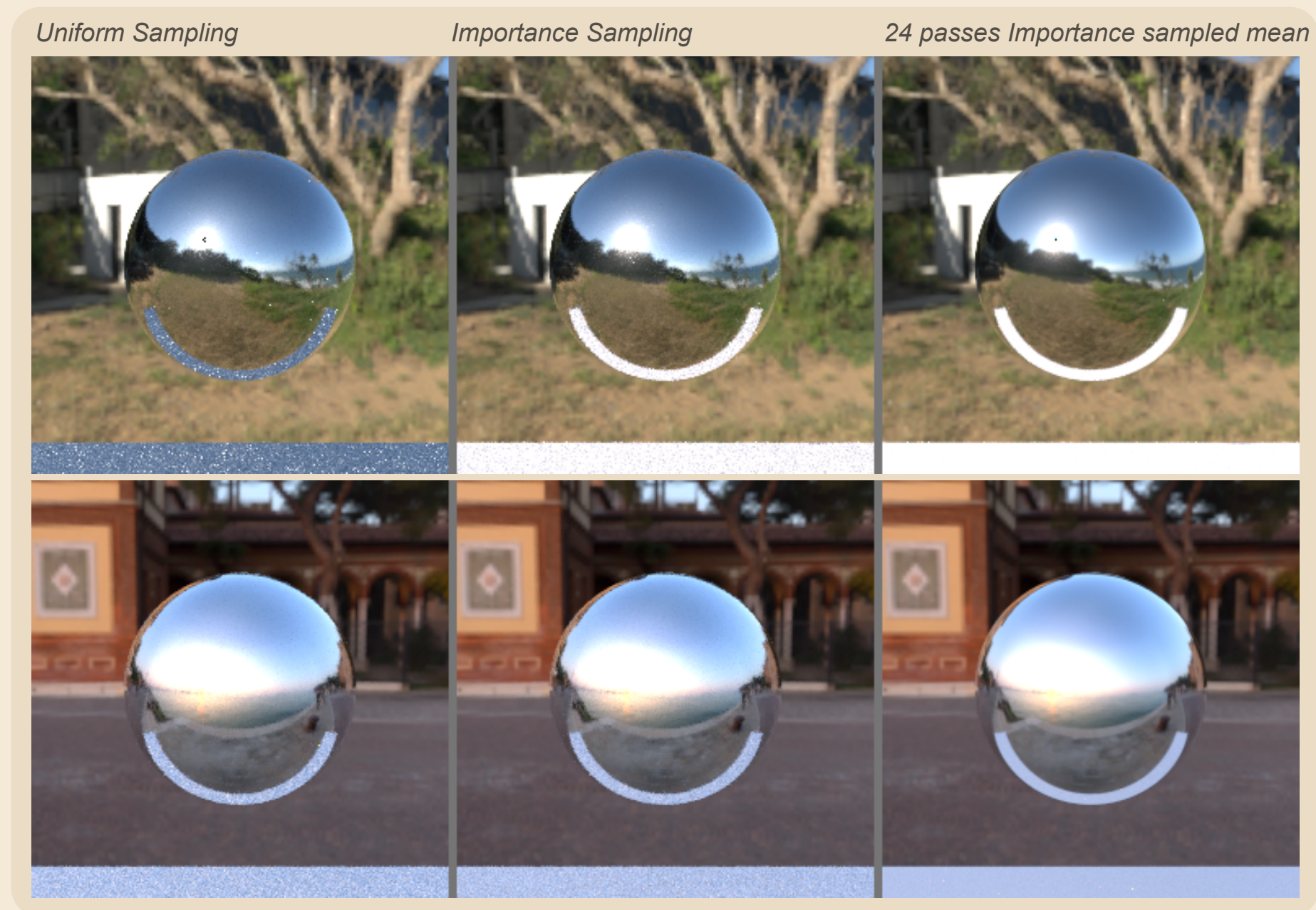
Sampled rays **were counted** across direction regions.
Those counts were **compared** with the probabilities predicted by *pdf()*.
Representative material mixtures **passed**.

CONFIGURATION	P-VALUE	RESULT
Diffuse	0.934	Pass
Plastic	0.893	Pass
Metallic	0.171	Pass
Mixed dielectric/metallic	0.828	Pass
Broad clearcoat	0.848	Pass
Clearcoat with all three lobes	0.909	Pass
Clearcoat over metallic base	0.572	Pass

$p \geq 0.01 \rightarrow$ pass: no strong evidence of disagreement. $p < 0.01 \rightarrow$ fail: investigate a possible mismatch.

Noise was reduced in different ways, with different costs

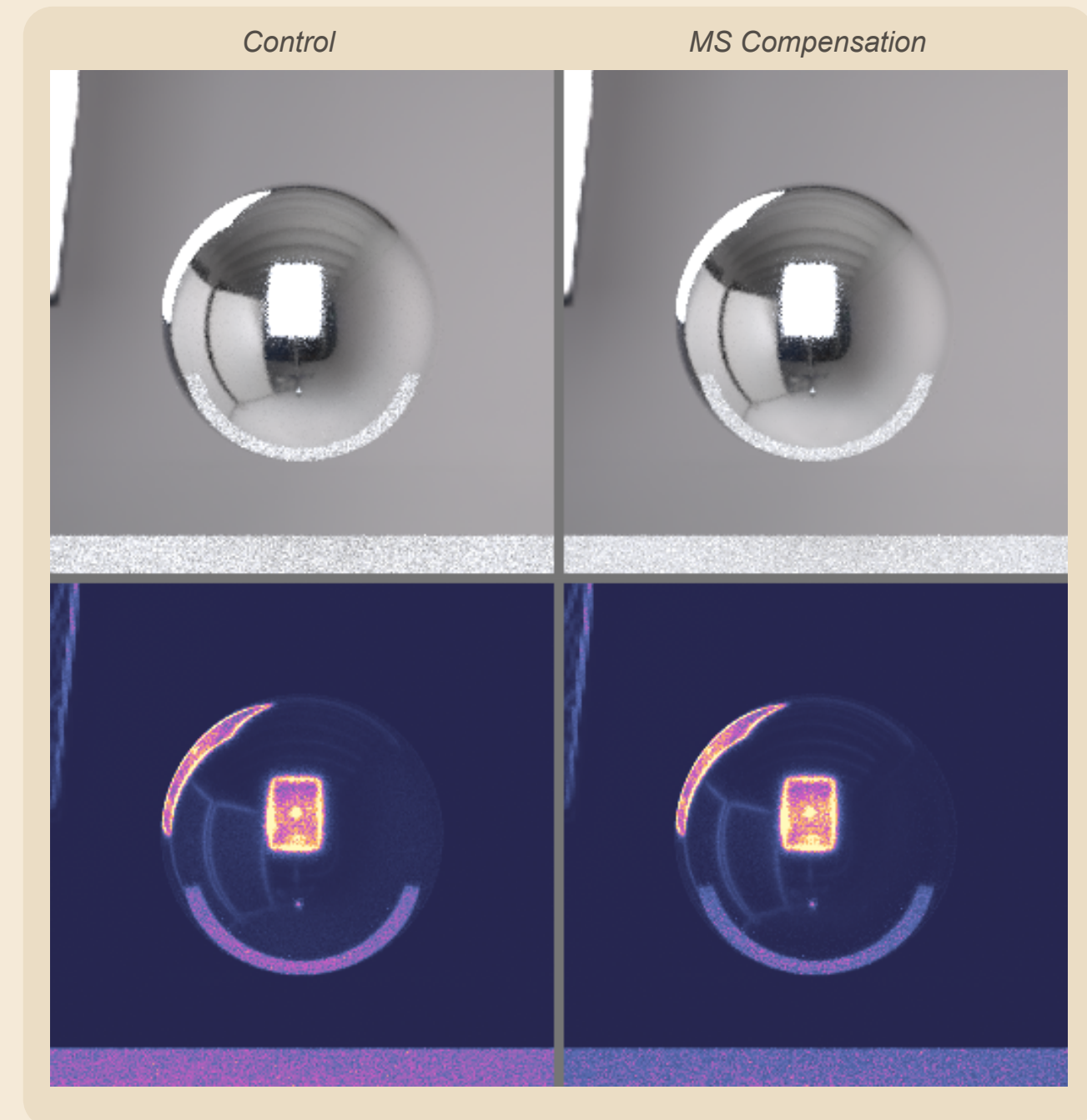
1 — Environment importance sampling



This test compares uniform environment sampling with importance sampling.

Uniform sampling treats every direction equally, while importance sampling sends more samples towards bright areas of the HDRI. At the same sample count, importance sampling reduced variance by 24% in Venice HDRI and 77% in Sundowner HDRI.

2 — Sampling configuration



Mean-subtraction compensation helped some HDRIs but worsened another, making it **scene-dependent**.

Stratification reduced noise by 14.2%.

A higher-resolution, max-pooled **importance map** reduced noise by 11.6%.

Noise was reduced in different ways, with different costs

3 — Firefly clamping

Can extreme samples be removed safely?



Clamping **removed** bright outliers effectively.

Clamp 3 **reduced mean brightness** by 40% overall and 61% on geometry.

It is **biased** and remains disabled by default.

Unbiased sampling improvements should come first; clamping and denoising are separate presentation tools.

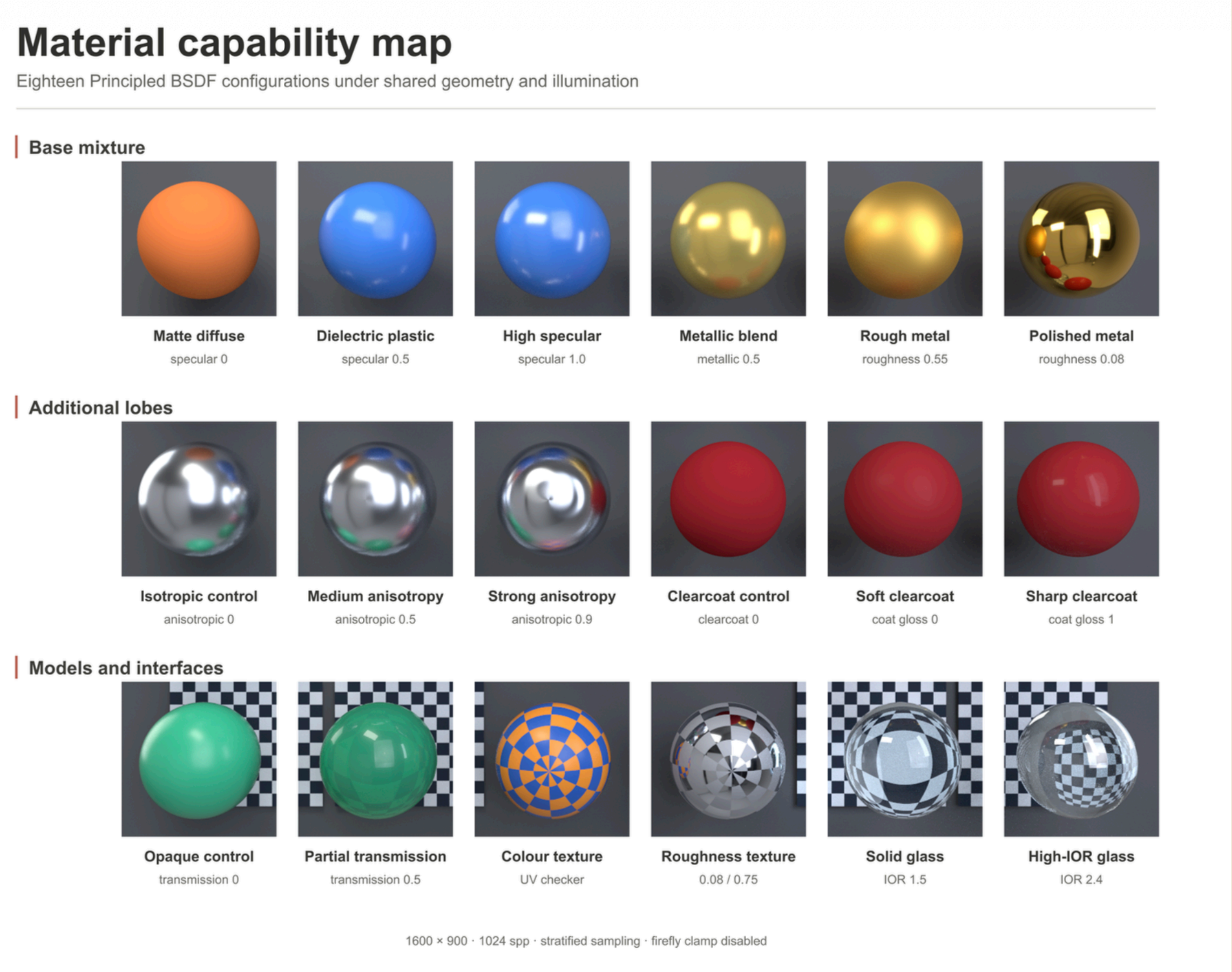
Features behaved as expected in isolation

All tests kept the scene and lighting fixed while changing one feature at a time.

Materials

- 18 Principled material configurations were tested.
- Roughness changed highlight sharpness.
- Metallic coloured the reflection and removed diffuse response.
- Anisotropy stretched the highlight in one direction.
- Clearcoat added a second glossy layer.
- Sheen added colour around grazing edges.
- Textures correctly controlled colour and roughness.

Each control produced the intended visible change without changing unrelated features.



LEGO tested whether an external scene survived the pipeline

Geometry and camera silhouettes **matched exactly at the tested resolution**. Overall framing, colours, lighting direction and brightness remained **close to Cycles**.

Cycles



Custom Renderer



Before attempting the Kitchen, the LEGO scene tested **scene transfer under manageable complexity**.

The Kitchen combined the complete renderer

Approximately **85 source materials** converted
301 shapes, **six physical-camera views**
Environment lighting, **practical** lights, **glazing** and complex **occlusion**
Raw, AOV and denoised outputs retained



The Country Kitchen was the first production-scale test of all custom components working together.

The realistic scene found a bug the small tests missed

Production scale exposed a visibility error that component tests could not see.

Unexpected room illumination



Remove practical lights — problem remains



Remove indirect bounces — problem remains



Make the window opaque — room is still illuminated

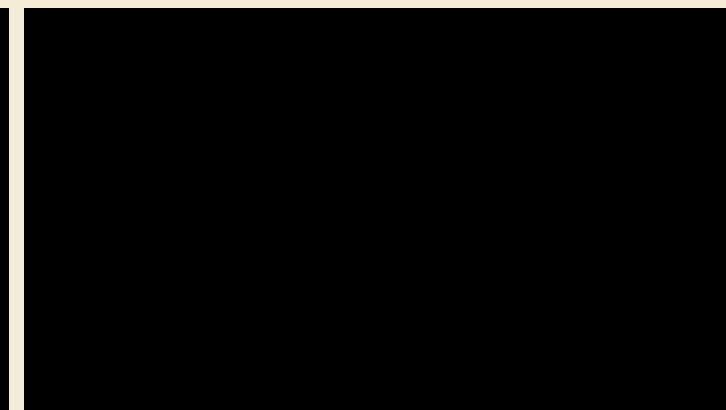


Environment shadow ray is too short



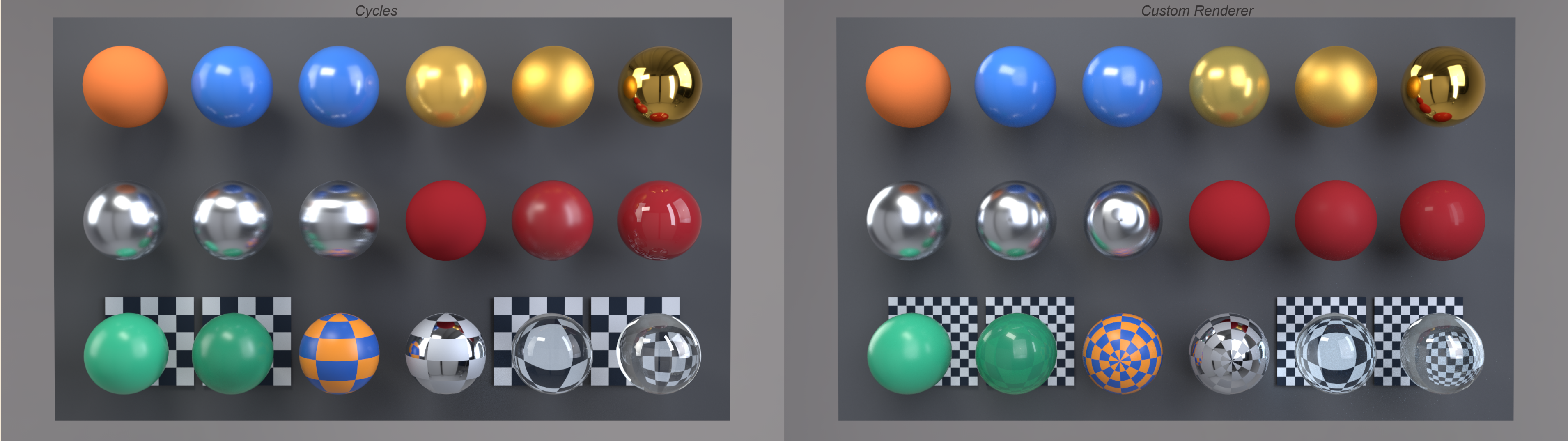
Correct its distance — opaque room becomes black

Restore transmission — light enters through the window correctly



The environment was **infinitely** far away, but the framework still needed a finite shadow-ray distance. The **ray stopped before reaching some distant walls**, so the renderer incorrectly believed the sky was visible. The failure does not weaken the validation, finding and isolating it demonstrates why production-scale tests were necessary.

Material Comparison



Both renderers reproduce the same diffuse, metallic, roughness, clearcoat, sheen, anisotropy, texture and transmission behaviours. Mean RGB differs by 3.6%; local highlights vary because Cycles uses a layered, multi-scattering material model.

HDR Environment Comparison

Cycles

Custom Renderer

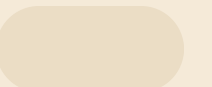


The sun position, horizon, reflections and direction of illumination are **preserved**, confirming correct HDRI **mapping and rotation**. Mean RGB differs by 1.4%; remaining noise differences come mainly from **filtering and importance sampling**.

Physical Camera Comparison



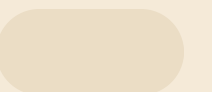
Framing, focal distance and foreground/background blur closely match Cycles. This is **the closest comparison**, with a 0.04% mean RGB difference and MAE of 0.0011.



COUNTRY KITCHEN — RAW COMPARISON



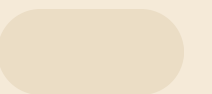
The complete pipeline **preserves** the composition, daylight direction, indirect illumination and main material identities. The custom output contains stronger fireflies around difficult glossy, glass and window-light paths.



COUNTRY KITCHEN — PRODUCTION OUTPUT



Both denoised images preserve the **same composition** and **lighting structure**. The custom result uses **clamping** and **AOV-guided OIDN**, producing a cleaner image while introducing some **brightness loss** from clamping.



STOVE CLOSE-UP



Reflective metals and depth of field are preserved in both renderers. Differences in the pot, knife and kettle highlights come mainly from anisotropy, clearcoat and the custom single-scattering GGX model.

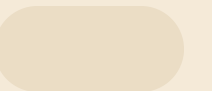


TABLE CLOSE-UP

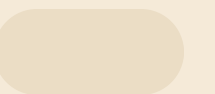


Both images preserve glass transmission, background visibility and contact shadows. The custom renderer uses smooth glass and approximate straight-through transparent shadows, so its glass does not behave exactly like Cycles.

RADIO CLOSE-UP



Texture placement and fine geometric details remain consistent between both images. Small differences in sharpness and surface response come from material conversion, sampling and reconstruction filtering.

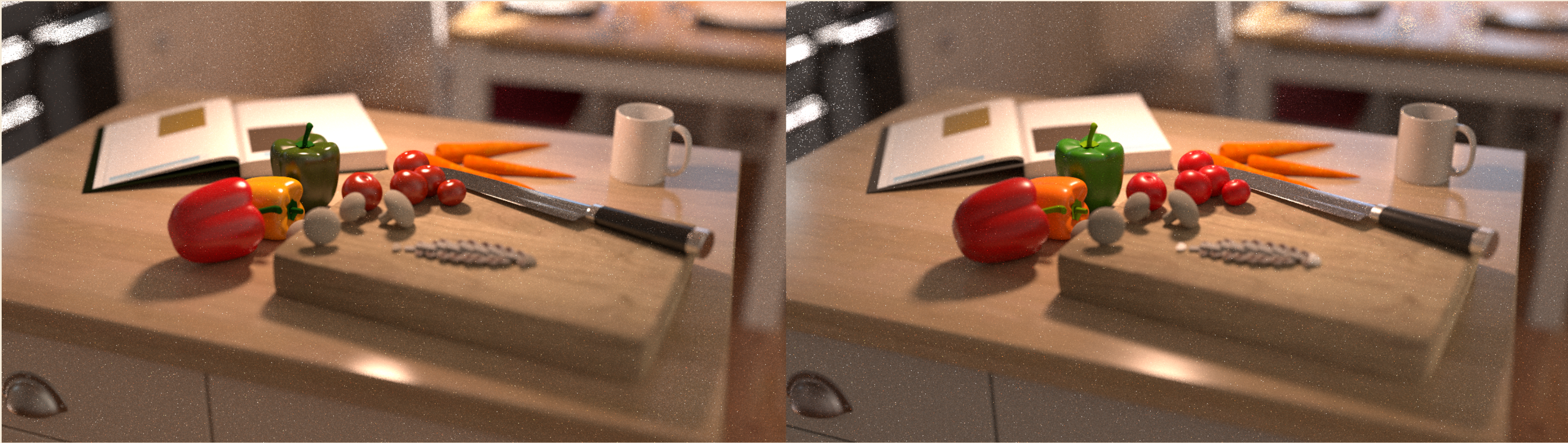


TOWEL CLOSE-UP

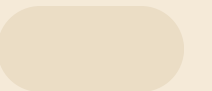


The towel's texture and diffuse shading are successfully preserved. The missing cupboard reflection occurs because the Cycles material contains gloss, while its converted custom material has no specular response.

KITCHEN-ISLAND CLOSE-UP



Both renderers preserve the mixture of materials, close occlusion and contact shadows in a complex area. Mean luminance differs by only 4.4%, showing strong agreement under production-style conditions.



A complete rendering pipeline was built and validated

What was **achieved**

- A **path tracer**, **Principled material**, **HDR environment** light and **physical camera** were implemented as connected Mitsuba plugins.
- **AOVs**, **firefly control** and **OIDN denoising** supported practical image production.
- The system worked from **simple test** scenes to an imported **LEGO** model and the production-scale **Country Kitchen**.

What the tests **showed**

- **Reference and statistical tests** supported the **correctness** of the main components.
- **Importance sampling**, **MIS** and **stratification** **reduced noise** by using samples more effectively.
- Results **closely reproduced** the **main material**, lighting and camera behaviour of **Blender Cycles**.

What was **learned**

- Correct equations are **not enough**: sampling, probabilities and renderer conventions must also agree.
- Small tests find **individual errors**, while **large scenes** reveal problems caused by **scale** and **component interaction**.
- **Clamping** and **denoising** improve presentation, but they do not replace **correct raw rendering**.

Still to improve — more complete rough glass, energy compensation, transparent shadows and Blender-material conversion.

A reliable renderer comes from testing every component separately, and then testing how they work together.

Thank you.

