RenderMan Assignment Report - Bowl

Siddhant Sawkar
NCCA
Bournemouth University, UK
s5629020@bournemouth.ac.uk

Abstract

This report documents my journey to replicate a ceramic bowl using Pixar's RenderMan and custom OSL shaders, producing a realistic render of the Bowl on a wooden table. The project involved modeling the bowl, writing shaders for its two-tone surface with dotted patterns, setting up lighting with an HDRI dome and point light, and configuring high-quality rendering parameters.

Keywords

RenderMan, OSL, ceramic bowl, rendering, texturing

ACM Reference Format:
Siddhant Sawkar. 2025. RenderMan Assignment Report - Bowl. In *Proceedings of SIGGRAPH*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/XXXXXXX.XXXXXXX

1 Introduction

This report outlines my process of rendering a ceramic bowl. I aimed to recreate its unique geometry, textured surface, and lighting interactions. The journey involved modeling the bowl's components, writing custom OSL shaders, setting up realistic lighting, and fine-tuning rendering settings. Along the way, I faced challenges in shader design and geometry alignment, which shaped my approach and deepened my understanding of RenderMan's capabilities [1].

2 Modeling the Ceramic Bowl

2.1 Analysis of the Bowl

Observing a ceramic bowl, I noted its design: a smooth spherical outer surface, a slightly smaller inner cavity, a sturdy cylindrical base, and a thin rim. The bowl's exterior features a brown strap near the base, transitioning to a lighter upper region with small, scattered dark dots, while the inner surface mirrors this light color with similar dots. The base and rim are uniformly brown, suggesting a glazed finish, and the bowl rests on a wooden table with a natural oak grain texture.

Figure 1: Real life Bowl Reference 1

Figure 2: Real life Bowl Reference 2

2.2 Implementation in RenderMan

Initially, I considered using a single sphere with displacement for the bowl but realized the inner cavity, base, and rim needed distinct primitives for accuracy. I used a sphere (radius 1, scaled by 0.6) for the outer bowl, a slightly smaller sphere (scaled by 0.585, offset by 0.006) for the inner cavity, a cylinder (radius 0.4, height -0.98 to -0.92) for the base, and a torus (major radius 0.5925, minor radius 0.0075) for the rim. Positioning these primitives was tricky; I struggled with

SIGGRAPH, Los Angeles, CA, USA

ACM ISBN 978-1-4503-XXXX-X/25/08
https://doi.org/10.1145/XXXXXXX.XXXXXXX

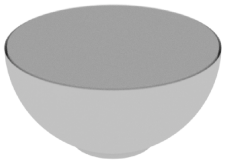

Figure 3: initial model without rim

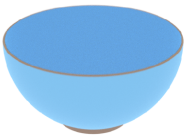

Figure 4: model with rim

aligning the inner and outer spheres to avoid visible seams. After several adjustments, I applied translations and rotations to align them correctly, ensuring the bowl sat naturally on the table, defined as a bilinear patch.

3 Texturing and Shading

3.1 Analysis of the Real Object

The ceramic bowl's surface is visually striking, with a two-tone color scheme: a dark brown strap near the base and a light gray upper region adorned with small dark dots. The inner surface is uniformly light gray with similar dots, while the base and rim maintain the brown hue, suggesting a glossy ceramic material. Capturing these details required shaders that could handle procedural patterns.

3.2 Implementation in RenderMan

To achieve the bowl's textured appearance, I developed two OSL shaders: `OuterBowlShader` for the exterior and `TwoToneBowlShader` for the interior drawing inspiration from procedural pattern techniques like those in the Redshift OSL Dots shader [3]. The `OuterBowlShader` uses a height-based condition (z <= -0.35) to apply the brown strap (RGB [0.22, 0.06, 0.04]) below this threshold and a light gray (RGB [0.45, 0.45, 0.45]) above, creating a sharp transition mimicking the real bowl's glazed band. For the dotted pattern, I implemented a procedural loop iterating over 400 points, using pseudo-random noise (based on sine functions with seeds 12.9898 and 78.233) to distribute dots in object space (s, t coordinates). Each dot's radius varies (base 0.0035, variation 0.80) to avoid uniformity, with distance checks (sqrt(dx*dx + dy*dy))

Figure 5: Working Shader without dot size variation

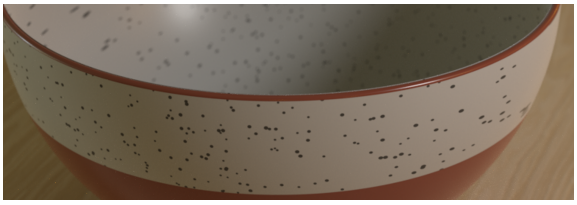

Figure 6: Added Dot size variations

determining if a point lies within a dot's radius, setting the color to dark gray (RGB [0.05, 0.05, 0.05]). The `TwoToneBowlShader` applies a uniform light gray with a similar dot pattern (base radius 0.0028, variation 0.65) but uses a split height (0.0) to maintain consistency across the inner surface. Early tests showed dots clustering near the poles, so I adjusted the noise frequency and added boundary checks (dx, dy adjustments for periodicity) to ensure even distribution. Both shaders connected to PxrDisney BxDFs [1], with parameters (metallic 0.0, specular 0.3–0.4, roughness 0.2–0.3, clearcoat 0.1–0.3) fine-tuned through iterative renders to balance glossiness and realism. The table uses `PxrTexture` and `PxrNormalMap` for oak veneer albedo, normal, and roughness maps, scaled by 5 to enhance grain detail, connected to a PxrDisney BxDF with anisotropic properties (0.3) for wood-like reflections.

4 Lighting Setup

4.1 Analysis of the Real Object

In a real-world setting, a ceramic bowl on a table is lit by a mix of ambient and direct light. Ambient light, often from a room's windows, creates soft, even illumination that highlights the bowl's colors and dotted patterns. A direct light, like a lamp, adds specular highlights and shadows, emphasizing the ceramic's glossy finish and the table's wood grain. I envisioned

the bowl in a cozy lounge setting, where an HDRI dome could simulate ambient room light, and a point light could enhance the bowl's curvature and material properties, creating a balanced, realistic scene.

4.2 Implementation in RenderMan

I set up a `PxrDomeLight` with an 8K HDRI map to provide ambient illumination, rotating it (90° X, 180° Y, 230° Z) to align with the scene [2]. A `PxrSphereLight` at (4, 2, -1) with intensity 40 adds directional highlights, particularly on the rim. Both lights use white (RGB [1, 1, 1]) to preserve color accuracy. Initially, the point light was too harsh, washing out the bowl's textures, so I adjusted its intensity and position through test renders. The HDRI map's lounge setting proved challenging to balance with the point light, but tweaking the exposure achieved the desired soft yet defined lighting, closely matching a real-world scenario.

5 Rendering Configuration

5.1 Analysis of the Real Object

Photographing a ceramic bowl requires capturing fine details like surface dots and gloss while maintaining focus. A real camera would use a shallow depth of field to emphasize the bowl against a slightly blurred background, with high resolution (e.g., 3840 x 2160) for clarity. Ray tracing is crucial for realistic light interactions, especially for the ceramic's specular highlights and the table's wood texture. The goal was a photorealistic render that highlights the bowl's material and context, with a perspective that feels natural and engaging.

5.2 Implementation in RenderMan

I configured the scene with the `PxrUnified` integrator, using a raytrace hider, 1024 max samples, and 0.001 pixel variance for high quality. The output is a 3840 x 2160 RGBA image saved as `Bowl.tiff`. A perspective projection with a 30-degree field of view provides a natural look, and depth of field (f-stop 100, focus distance 3.2, lens distance 2.8) keeps the bowl sharp. Camera placement (translated to [0, 0.12, 3.2], rotated -20° X, 22° Y) was adjusted iteratively to frame the bowl effectively. Early renders showed aliasing on the rim, so I increased samples to smooth edges, though this extended render times. I found a indoor HDRI with good lighting setup that matched my reference for some good renders.

5.3 Depth of Field Calculation

The depth of field was calculated using the thin lens model, where the circle of confusion determines the blur for out of focus regions. The focus distance (3.2 units) was derived as the distance from the camera's position ([0, 0.12, 3.2]) to the bowl's center, approximated at the origin ([0, 0, 0]), using Euclidean distance: $\sqrt{(0-0)^2 + (0.12-0)^2 + (3.2-0)^2} \approx 3.2$. The f-stop (100) was chosen to create a shallow DoF, balancing blur with computational efficiency, as lower f-stops (e.g., 2.8) caused excessive blur in test renders. The lens distance (2.8 units) was set slightly less than the focus distance to fine-tune the focal plane, ensuring the bowl's rim and upper surface remained sharp. The CoC radius, computed as $\text{CoC} = \frac{|\text{object distance} - \text{focus distance}| \cdot \text{focal length}}{\text{f-stop} \cdot (\text{object distance} - \text{focal length})}$, guided the blur for the table (approximately 0.3 units behind the bowl), producing a subtle background softening that emphasized the bowl's details without distracting from the scene's context.

Figure 7: Final Render 1 (Fixed Dot size Variation)

Figure 8: Final Render 2 (Two tone Color)

6 Future Improvements

Looking back, I'd refine the alignment of the inner and outer bowl spheres, for more thickness. I spent considerable time tweaking transformations, but a more precise mathematical approach to scaling might have helped. The dot patterns in the shaders could be improved; they sometimes appeared too regular near the bowl's poles. Adding a shader for the rim for chipping effect would have helped me achieving realism. Better reference photos would have helped

document the bowl's subtle textures, as I struggled to capture its glossiness in my initial observations.

7 Conclusion

This project was a rewarding journey in recreating a ceramic bowl with RenderMan. From modeling the bowl's components to crafting custom OSL shaders and balancing lighting, each step presented challenges that deepened my understanding of rendering techniques. The final render captures the bowl's essence—its two-tone design, dotted patterns, and glossy finish—while the wooden table adds context. Despite some hurdles, the result is a visually compelling scene that demonstrates RenderMan's power in translating real-world objects into digital art.

References

[1] Pixar Animation Studios. 2025. RenderMan 26 Documentation. https://rmanwiki-26.pixar.com/space/REN26/19660802/RenderMan+26+Documentation Accessed: 2025-05-12.

[2] Poly Haven. 2025. Lythwood Lounge HDRI. https://polyhaven.com/a/lythwood_lounge Accessed: 2025-05-23.

[3] Zap Andersson. 2019. Redshift OSL Shaders: Dots.osl. https://github.com/Maxon-Computer/Redshift-OSL-Shaders/blob/main/Dots.osl Accessed: 2025-05-16.