# Principles of Rendering

Chengyan Zhang*
National Centre for Computer Animation

## Abstract

This document is for the presentation and analysis of my chosen object for the Principles of Rendering assignment. This paper will be covering various issues such as: caustic effects using ”PxrLMGlass”, general Bxdf using ”PxrDisney” and the use of OSL generic shaders to produce displacement patterns on the applied surfaces.

## 1 Introduction

Firstly, my object chosen for this assignment consists of mainly quadric primitives provided in the Renderman framework.

With a lens frame being a cylinder, the lens being a transformed sphere, the handle too, a cylinder and the rear of the handle being a cone. The section in between the lens frame and the handle can either be a defined patch or a sliced and transformed torus.

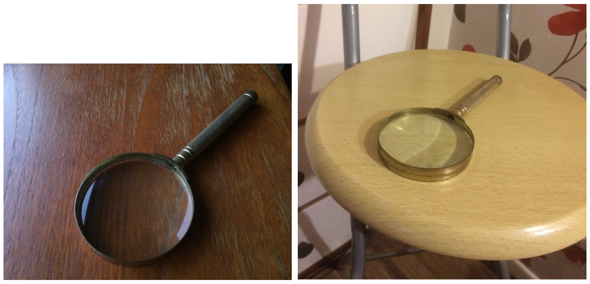


**Figure 1:** *A photo of the object in an interior environment afternoon.*

The magnifying glass in Figure 1 exhibits:

1. Caustic properties on the lens.
2. Metallic surface overall.
3. Dirt and oxidation on top of the metal, making the specular dull at places.
4. Sharp specular highlights on certain areas.
5. Lattice pattern with several indented ring at specific areas of the handle.
6. Wooden plane sits under the object.
7. Has a distinctly strong light in one direction (A window).

The stated effects can be achieved through mainly using Renderman’s Bxdf as a foundation and have specific surface shaders to produce color patterns or noise to produce effects such as rust and dirt. The lattice pattern on the handle can be achieved by either using a series of formulas in a displacement shader or by using a cross-patterned texture to repetitively map the displacement patterns onto the handle.

After looking through the material section of the Renderman Documentation, I’ve found examples of using PxrGlass or PxrLMGlass to produce effective caustic effects, which is very useful for creating the lens of my object.

Use of references are cited here:

1. [Renderman Support 2017]
2. [Sony Pictures Imageworks Inc - Larry Gritz 2017]

## 2 Method Overview

For the lens, I used PxrLMGlass with ”thin” variable turned on to avoid dark shadows affecting the reflections according to the documentation. Most surface properties are handled by PxrDisney using a noise shader to introduce rust and adjusting global specular.

For the lattice effects are produced by a series of repeating displacement patterns in a shader which is read by PxrDisplace. All displacement patterns on the handle are compounded into a single float value for each point on the cylinder to proceed with displacement scalar.

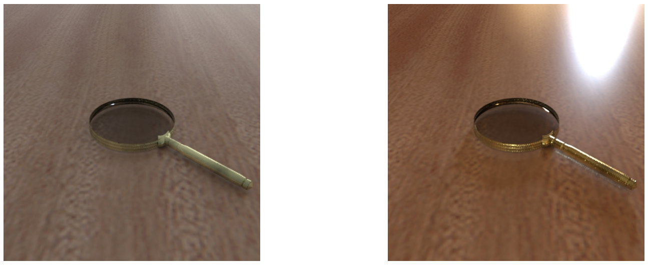
An example of a displacement pattern logic:

```
procedure REPEATINDENT($objBound - P, repeat - r, depth - d, fuzz - f, width - w$)    ▷ Repeat indent on a cylinder.
    while $P \neq 0$ do    ▷ We have the solution if routine exceeds object boundary.
        $V \leftarrow r * v \bmod 1$
        $disp \leftarrow |V - offset|$
        $result \leftarrow 1 - interpolate(width - fuzz, width + fuzz, disp)$
        $final \leftarrow -result * d$
        $Increment(P)$
    return $final$▷ The displacement of the entire point group.
```

In addition, the lens frame’s patterns can be done with a similar approach; using a repeating indentation with a repeat count of 2 for the two rear edges of the cylinder and a bigger ring of positive displacement at the center of the frame. The same method can be used for the small indentation patterns on the rear edge of the frame by repeating in the u-axis of the object.

[*] e-mail:s4903744@bournemouth.ac.uk

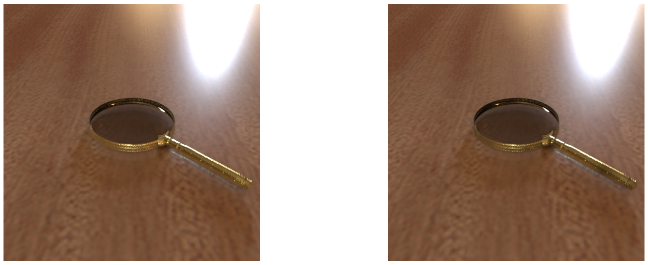
## 3 Results



**(a)** *.tx made from .jpg*

**(b)** *.tx made from .hdr*

**Figure 2:** *A side by side comparison of the use of an environment light with the source being a jpeg [Frederick - The Belgian VFX Guy 2013] and the other being a hdr [Adaptive Samples 2015]. Although the two maps have different maps but the with a lower dynamic range, the jpeg map causes some reflection artifacts and dull lighting.*



**(a)** *Linear Base Colour*

**(b)** *With Perlin Noise*

**Figure 3:** *A side by side comparison of a linear base colour in a shader compared to a Perlin noise shader. Dull and rusty stains are apparent on the handle and the lens frame. Furthermore, I have tested that increasing the internal frequency parameter separates the stains into small partitions of noises which isn't desirable for a metallic material in most cases. However, increasing turbulence is not tried and tested.*

## 4 Evaluation

Firstly, the lens on the original object from Figure 1, has more internal reflections and highlights at certain areas whereas my renders shows only a simple glass surface without any additional details, see Figure 3. Using a PxrVCM to carry out a bidirectional raytracing would not help as I have not implemented a complex shader for my lens and only used the default PxrLMGlass due to time constraints.

Secondly, the pattern on the lens frame might be somewhat inaccurate in terms of their proportions where the rim patterns are much smaller than my renders. However, the displacement pattern I've defined on the frame seems to produce artifacts and tearing if I set the displacement scalars too high or too low. I have not found out the reason behind the tearing due to time constraints.

Additionally, the model of the part connecting the lens frame and the handle can possibly be more accurately depicted by using polygon patches with specified points. Using a sliced torus is simpler but makes the component bulge out.

Finally, I think that the global base colour might require more noise and turbulence instead of a single layer of Perlin noise. The current render may still look a-little too platonic, changing the frequency made the noise less pronounced in a single area so I decided to not increase the frequency and stay at the current value.

## References

Adaptive Samples, 2015. [hdri] st lucia interior. `http://adaptivesamples.com/tag/free-hdr/`.

Frederick - The Belgian VFX Guy, 2013. Ibl and environment maps for cgi. `http://www.thebelgianvfxguy.com/2013/02/ibl-and-environment-maps-for-cgi.html`.

Renderman Support, 2017. Ris developers' guide. `https://renderman.pixar.com/resources/RenderMan_20/geometricPrimitives.html`.

Sony Pictures Imageworks Inc - Larry Gritz, 2017. Open shading language 1.9 language specification. `https://github.com/imageworks/OpenShadingLanguage/blob/master/src/doc/osl-languagespec.pdf`.