RenderMan Assignment Report - Massage Ball

Jack Purkiss
s5603002@bournemouth.ac.uk
NCCA
Bournemouth, UK

ABSTRACT

A report to demonstrate the use of RenderMan and OSL to render two images replicating a real life massage ball.

KEYWORDS

RenderMan, OSL, RIB

ACM Reference Format:
Jack Purkiss. 2024. RenderMan Assignment Report - Massage Ball. In *Proceedings of RenderMan Assignment Report - Massage Ball (Conference acronym 'XX)*. ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

This report outlines the progress of my rendering project, to create a realistic render replicating the massage ball object (figure 1). Beginning with alternative modelling approaches, I explore the use of a point based approach vs a displacement based approach and adding the details of the spike positioning on the ball. The surface shader is then explored, looking at the BRDF model used and how the specularity and roughness is adjusted across the render to produce a realistic looking result.

2 MODELLING

2.1 Python Point-based approach

Upon initial observation, it is clear that this object has two main modelling aspects: the sphere and the spikes. The object is a ball with 12 rows of spikes, which are mostly evenly distributed. Therefore, a sphere primitive was used as the base, with the aim of adding spikes. My first approach involved using Python to procedurally determine the points on the sphere for placing the spikes, and then positioning a paraboloid quadric at each of these points. This appeared to be a successful approach as I calculated the angles required for each row, and then found the azimuth angle to find the points on the sphere's surface to place the paraboloids. However once I reached this point it became clear that continuing with this approach would cause more challenges than needed, as it would require more adjustments to the shaders and further complications with the spike's rotations.

Conference acronym 'XX, May, 2024, Bournemouth, UK

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

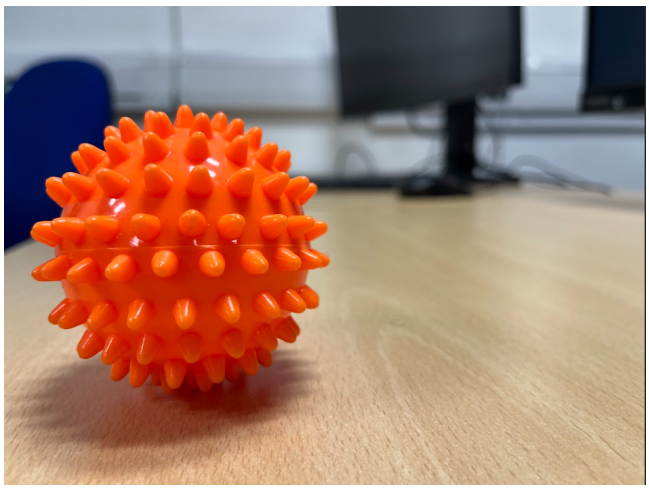

Figure 1: Massage ball

Figure 2: Massage ball from above

2.2 Displacement based approach

In the end, I opted for a displacement based approach, where I used an array with a length of 12 containing the number of spikes within

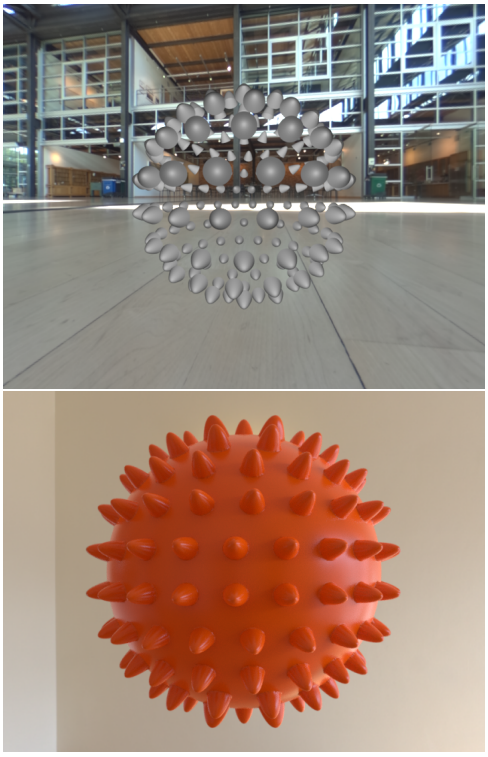

Figure 3: Python points and initial displacement approaches

each row, apart from the poles. Then I calculated the repeat counts for U and V using this to position the tiles within the sphere to displace from. I was able to find the quadratic falloff which was suitable for the paraboloid shape of the spikes by using the function:

$$\text{dispShape} = \max\left(0, 1 - \left(\frac{\text{dist}}{\text{radius}}\right)^2\right)$$

with dist being the distance from the centre point to the tile location, and the radius of the spike.

2.3 Spike alignment

With a closer look at the ball, it becomes evident that the points are not all perfectly aligned. To address this, I introduced some noise to the centers of the spikes, which slightly altered their locations and shapes. This was because of the change in distance between the uTile and vTile variables adjusting the quadratic falloff function. Additionally, the bottom hemisphere of the ball does not completely line up with the top. To add this to my render, I adjusted the value used to find the uTile variable based on if v was in the top or bottom hemisphere, causing a rotation in the position of all the spikes in the lower hemisphere.

2.4 Middle band

On the ball, there is also a seam where the two hemispheres meet. To make this, when v=0.5 I added a value called inLine to find the area within that line. I made a band with a bit of noise and a narrow bandwidth finding the difference between the top and bottom of

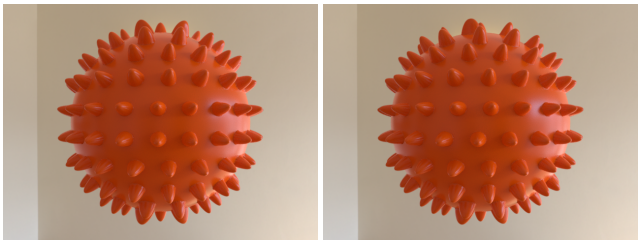

Figure 4: Adjusted point and rotated lower hemisphere

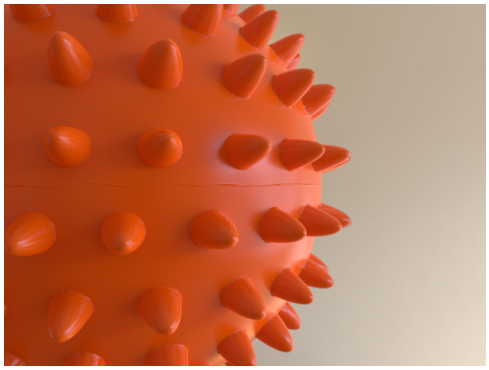

Figure 5: Middle Band Displacement

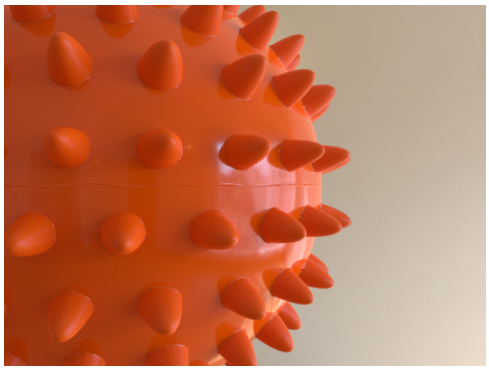

Figure 6: Adjusted specularity and roughness

the bandwidth with smoothsteps to determine whether a point is in the line. Then if it is in the line, a small amount of displacement is added to produce this seam.

3 BRDF

I chose to use the PxrDisney BRDF as the main areas I was focused on were the contrasting specularity and roughness between the main ball surface and the spikes. The main surface appears quite reflective and smooth, while the spikes produce more diffuse reflections and have a rougher surface overall. With this in mind, I began by using the values from creating the spikes to determine whether to be more specular or more rough by hard-coding these values directly.

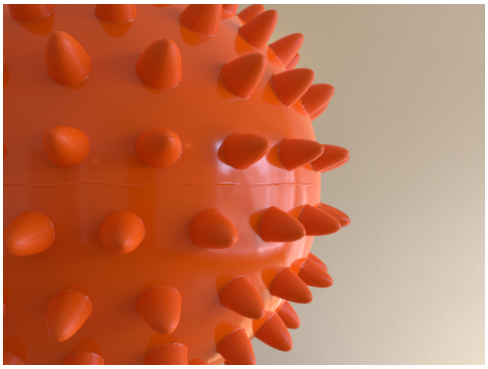

Figure 7: Spike roughness displacement

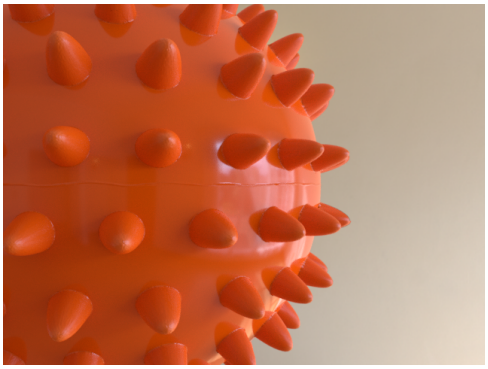

Figure 8: Spike tip wear

3.1 Spike roughness displacement

The spikes on the existing object have more texture than what is seen on the main surface. There appears to be small rings of displacement around the circumference. as the spike gets higher. In an aim to replicate this, while adding the displacement to form the spike shape, I used a mod function for every 0.1 value to put a small amount of additional displacement to the spike creating a physically rougher appearance.

3.2 Spike tip wear

As the existing ball is used, the tips of the spikes are what makes contact with anything, resulting the tops of these areas being a bit paler in colour and shinier as well. With this, I added it so that when the spike value is above 0.8 I slightly adjust the colour and the specularity mix for the colour with a smoothstep for the factor to change by.

3.3 Surface wear

On the main specular surface, there are a few blemishes that I attempted to add as well. As the surface is not perfectly shiny, I first added some layers of noise to slightly increase the roughness and reduce the specularity. This added a general layer of variation to the surface as a base for more. On top of this, the surface has quite a few blotches from small stains from water, and a closer roughness with lots of smaller scratches. For this, I adapted code [1] to distribute

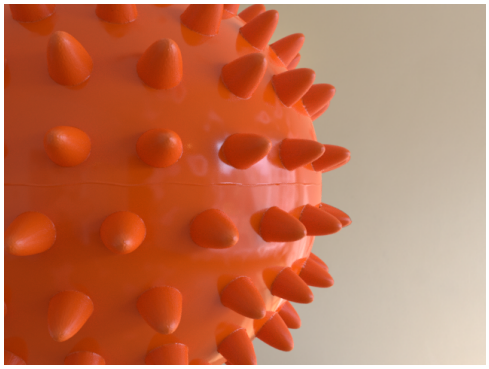

Figure 9: Initial surface noise

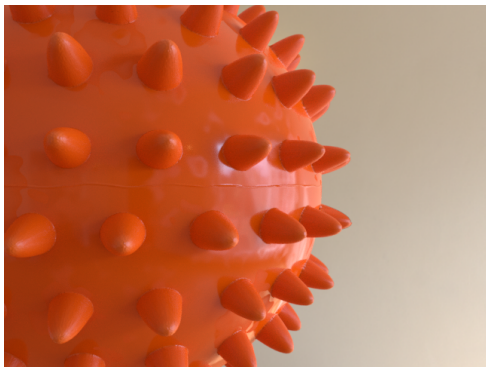

Figure 10: Added blotches

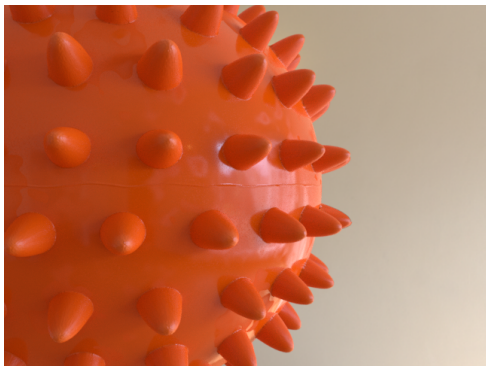

Figure 11: Added small dots

spots across the surface. This involved setting a scale and a radius for each spot and iterating over the x, y, z, to create a rndpint with some noise, to randomly distribute them across the surface. By setting a larger scale and a noisy radius with a higher frequency, I was able to achieve the larger blotches by increasing the roughness in these areas. For the small dots, I significantly reduced the scale, and applied the same technique, this time also making them a slightly darker colour than the rest of the surface.

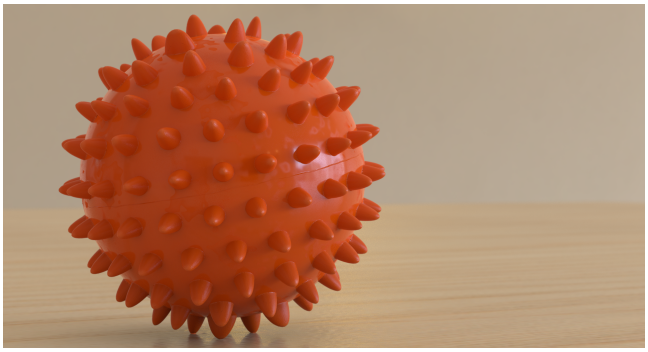

Figure 12: Final Render 1

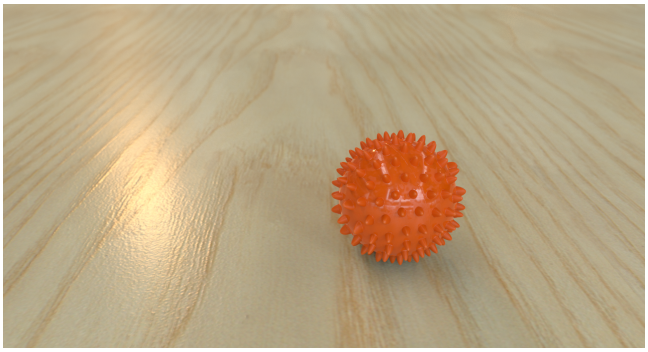

Figure 13: Final Render 2

4 FINAL RENDER SETUP

To set up the first render, I found an environment map that had a nice window as a light source to show some of the specular reflections while still having an overall softer lighting setup [2]. For the second, I used a different environment map based in an airport [3] because I liked the a large overhead lighting to show the specularity from a different angle. I used the Hackberry wood from the pixar library and adjusted it to repeat 4 times across the patch it was on to maintain its quality. I did the same for its bump map for a bit of displacement. In both renders, I set up the ball on the wooden table, and wanted to take one picture closer up to highlight the details of the roughness and one further away to display the object from another angle. In both of these cases, I used depth of field to draw attention to the ball, both using an f-stop of 1.4 with the focal length adjusted for its distance from the camera.

5 FUTURE IMPROVEMENTS

Going forward, the key area I would like to improve is the scaling of the spikes. This was the main issue I had when moving to the displacement based approach. The problem was the tiles scaling differently as they got closer to the poles, resulting in a wider spike than what should appear. I understood that to fix this I had to adjust the points to world space and find their positions on the sphere through that, however after a lot of time spent trying it was unsuccessful. Additionally, I think the general scale and positioning of the spikes is something that is almost there, but is slightly off compared to the real object. I also could have added some subsurface scattering as the ball is hollow and it becomes obvious when a light is shone directly on it. This was just an oversight from me and something I did not think of until it was too late. In terms of documenting the project, I would have liked to have better reference photos to discuss within this report. Many of the details of the ball that I have discussed, I struggled to document with photos as they were quite subtle.

REFERENCES

[1] Inc. Redshift Rendering Technologies. 2021. Dots.osl. https://github.com/redshift3d/RedshiftOSLShaders/blob/main/Dots.osl. Accessed: 2024-05-25.
[2] Greg Zaal. 2020. Lebombo. https://polyhaven.com/a/lebombo. Retrieved May 20, 2024.
[3] Greg Zaal. 2020. Rostock Laage Airport. https://polyhaven.com/a/rostock$_{l}aage_{a}irport$. $Accessed$: 2024 − 05 − 27.