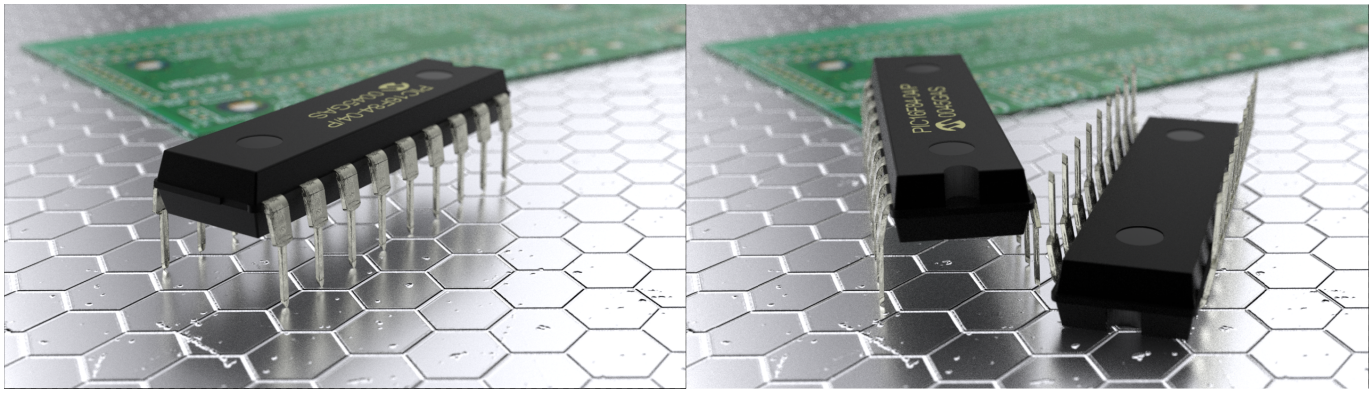
Rendering Assignment

Vijin Ravindran*
MSc CAVE, Bournemouth University



1 Abstract

This is an assignment to model and render a real-world object in a minimal scene, complete with an appropriate BRDF, camera artifacts and natural variations. The object I selected for this project is a microchip. I used the Renderman Python interface in Visual Studio for coding, and Renderman It for image output. The version of Renderman used is 21.2.

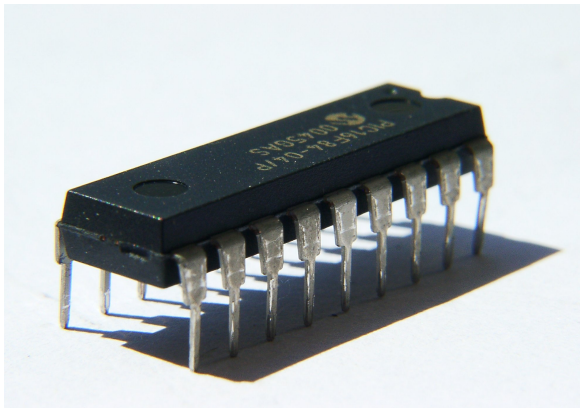


Figure 1: *Reference image*

2 Object Analysis

Microchips vary in size and shape, so I selected one high resolution image as reference, but the rest of the images were helpful in analysing the shape of the object from angles other than that of the reference image. I also have a single-board computer with an exposed circuit board for reference. However, modern microchips are thinner and smaller. The older models were a better candidate for modelling because the features were clearly visible and well-defined.

I set up different viewing angles apart from the main final angle, to observe the details closely. I was able to get a good amount of accuracy with this approach. It helped to switch faster between views to observe the accuracy of the primitives. The main body was modelled using simple primitives such as cylinders and spheres for rounded edges, so having a close look at the corners was essential for the accuracy of the model. (Stephenson, I.)

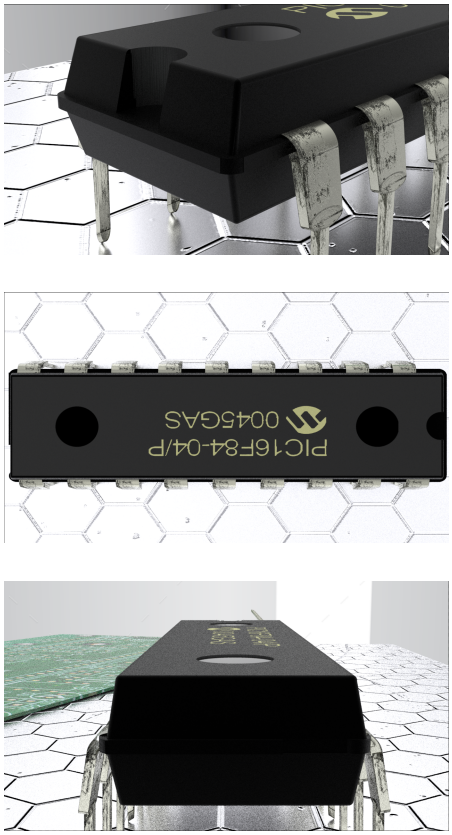


Figure 2: *Camera Angles*

3 Modelling

The first step was to build the main body of the chip, using simple patches. I only built two adjacent sides and the top. The other

*e-mail:s4900744@bournemouth.ac.uk

two sides were copies of the same sides rotated 180 degrees on the y-axis. The same method was used for the underside. This was the basic shape of the chip. The pins were irregularly shaped and had many different individual primitives in them. However, I used cylinders and patches (both bilinear and bicubic) to model the basic shape. The bottom thinner portion (lead) was basically a scaled down copy of the top wider portion.

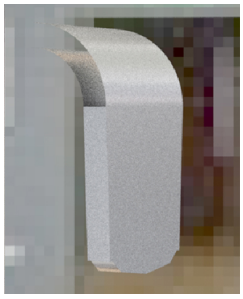


Figure 3: *Pin shape*

I added the protruding edge in the middle with slightly rounded edges, to make the model as similar to the real object as possible. This included the 8 edges and corners of the top part, and the four corners of the middle protrusion. I was not getting the look I wanted with displacement shaders, so I opted to use cylinders and spheres to simulate the effect of a smooth bend. Similarly, the corners of the protrusion was made out of discs and cylinders.

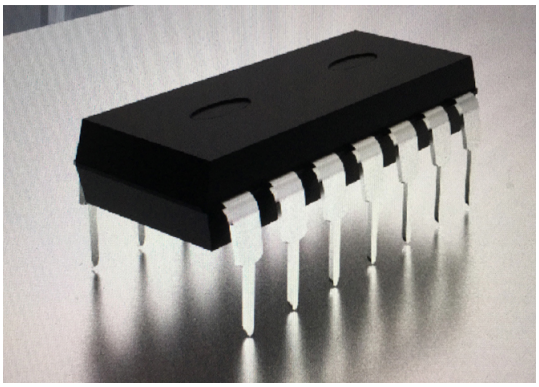


Figure 4: *Initial Model*

There were no primitives that could be used for the sides of the pins top portion, which was basically a flat 2D ring with a 90 degree sweep. For this I used a bicubic patch modelled based on the two cylinders that formed the pins curve.

The 16 points positions could be derived using trigonometric functions at 30 and 60 degrees for equal spacing as shown in figure 6. The other parts where made using patches and replicated nine times for each side using a for-loop.

The primitive was placed at precisely the right position to cover the cavity of the pin's top section.

4 BRDF Model

The main part of the body is black, with a slightly metallic, rough texture. The supporting images shows white reflective metallic textures for the pins but for applying a natural variation, it was useful to have a tarnished look with scratches on it. The light source in the reference image casts a sharp shadow on the ground plane.

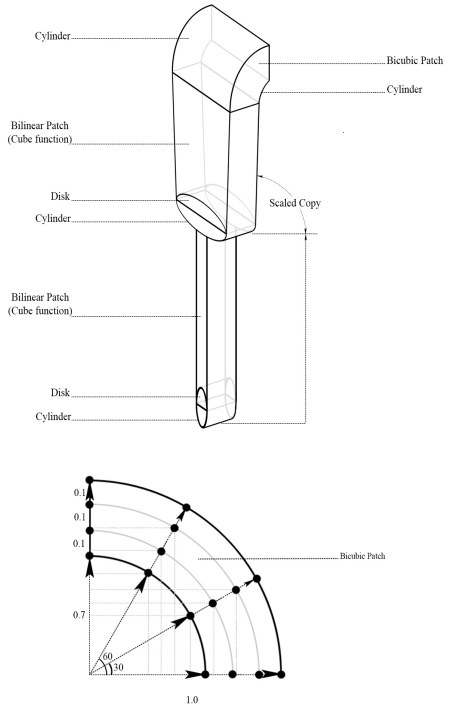


Figure 5: *Pin primitives*

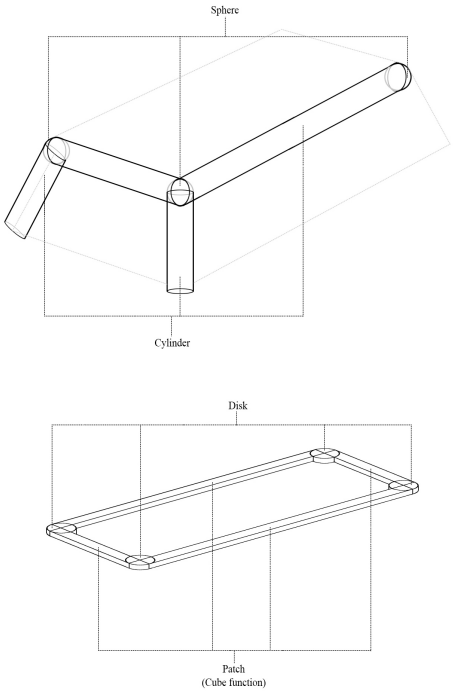


Figure 6: *Rounded corners*

The main part of the body was given an RGB value of (0,0,0) with the metallic parameter set to 1 and a roughness value of 0.5. This gave a matte finish to the surface. The pins were metallic as well,

but a lower roughness value of 0.2. As per the reference image, the pins were given a green/yellow tinge. This added to the tarnished look of the pins.

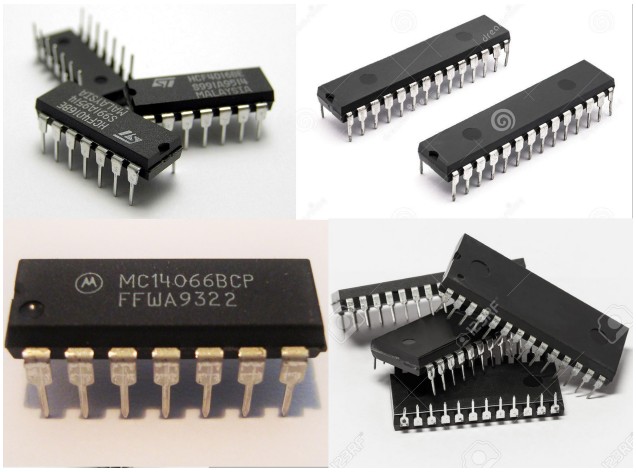


Figure 7: *Supporting images*

5 Textures and Displacement

The top face of the micro-chip's main body has two embossed discs with a glossy finish, and a semi-circular cutout at one of the the smaller edges that went all the way down to the middle section. It also has the model number and logo printed in yellow on the top.

In the shader topDisp.osl (Gritz, L.), I divided the patch that forms the top face into two parts - where the texture coordinate v is less than 0.1 and where v is greater than 0.1. At the points where v is less than 0.1 (farther edge), I applied the semi-circular cutout and the for the rest of the geometry, the equidistant discs were displaced using the same shader. For the glossy finish, I places two disks of the same material as the rest of the body, but with roughness value 0 on these discs, and used a texture map for the printing the text on top. The same was done for the bottom patch (bottomDisp.osl), but without the cutout in the edge, as some supporting images showed similar glossy discs on the bottom as well.

6 Natural Variation

The reference image showed the pins as having signs of wear and tear, so the points where the shape was deformed were slightly brighter than the rest of the surface.

I applied a bump map on the pins of the microchip. Adding only a color noise seemed too uniform through a range of frequencies and it was not looking naturally degraded, so I added a bump map inside the for-loop that generates the pins and set the scale to a random number between 0 and 1, and also the invertT parameter to a modulo of the loop iteration and the integer 2. This way I was able to get good randomness on the pattern. Also, the reference image had no textures on the ground plane. I added a white metallic texture with a bump map of hexagonal blocks, and added some scratches to the bump map.

7 Environment map

I used the PxrDomeLight with an HDRI image as the light source to light the scene. This image gave good overall lighting. and subtle shadows for the bulk of the body. Since neither the object nor the floor are fully reflective, the purpose of this image was mainly to provide lighting, and not reflections off shiny objects in the scene.

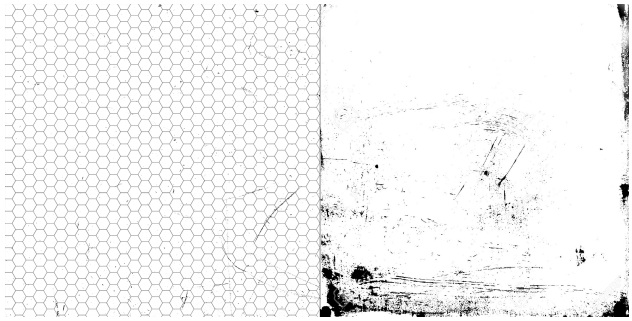


Figure 8: *Bump maps - floor(L), pins(R)*

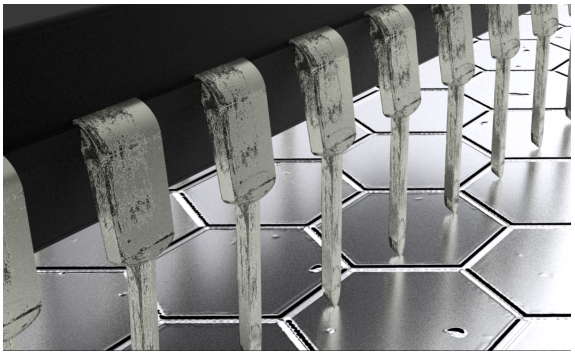


Figure 9: *Scratches on pins*

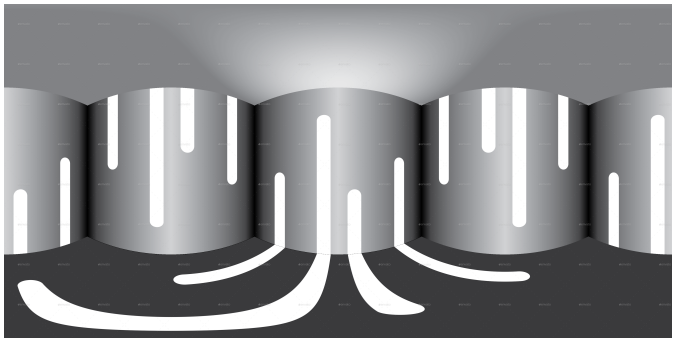


Figure 10: *Light Source*

8 Camera Artifacts

The main reference image has a focal point at a point in the middle of the microchip, with the ending of the printed text shown with clearly defined edges. The chip's farther edge is blurred, as was the near bottom edge and a few pins close to the camera.

I used a depth of field setting with fstop at 5.6, focal length at 0.2 and focal distance at 2.2 to get this effect. The field of view is set at 20. I also placed a cube with a circuit board map in the background to emphasise the blurring effect.

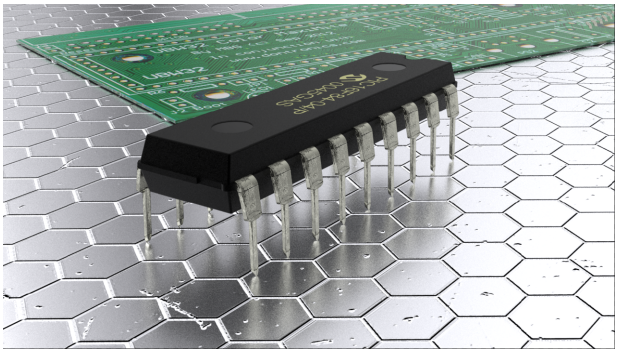


Figure 11: *Image with no depth of field*

9 Conclusion

The rendered image has 1280 x 720 resolution, has been gamma corrected and should be viewed in sRGB mode. Two different angles have been shot to illustrate both sides of the chip. The two angles can be switched in the python file by changing the name of the output file as ”Chip-underscore-01 and Chip-underscore-02.

10 References

GRITZ, L., 2017. Open shading language 1.8 language specification.

STEPHENSEN, I. 2002.Essential Renderman. Springer.

Figure 1: Available from: https://de.wikipedia.org/wiki/ IntegriertErSchaltkreis/media/File:IntegratedCircuit.jpg [Accessed 14 April 2017]

Figure 9: Available from: https://3docean.net/item/hdri-studio-lightbox-01/16389276 [Accessed 15 April 2017]