

# Rendering and Shading Assignment

Rosie Campbell

Jan 2016

## 1 Introduction

This report documents an assignment undertaken as part of a Rendering and Shading course. The assignment involved modelling a simple object using Pixar's Renderman API, writing and applying appropriate surface and displacement shaders and other visual effects.

The object chosen for the assignment was a squash, and the final piece depicts a number of varieties, including a pumpkin.

## 2 Collecting photographs

- a) Collect photographs of the object to support and document each stage of the project.
  - Internet
  - Own

## 3 Modelling in RIB

- b) Model it using simple primitives, in either directly RIB, other RenderMan API (such as the C or Python API's).
  - Squished sphere

## 4 BRDF model

- c) Identify and implement an appropriate BRDF model (or models) for the object.
  - Disney
  - Waxy

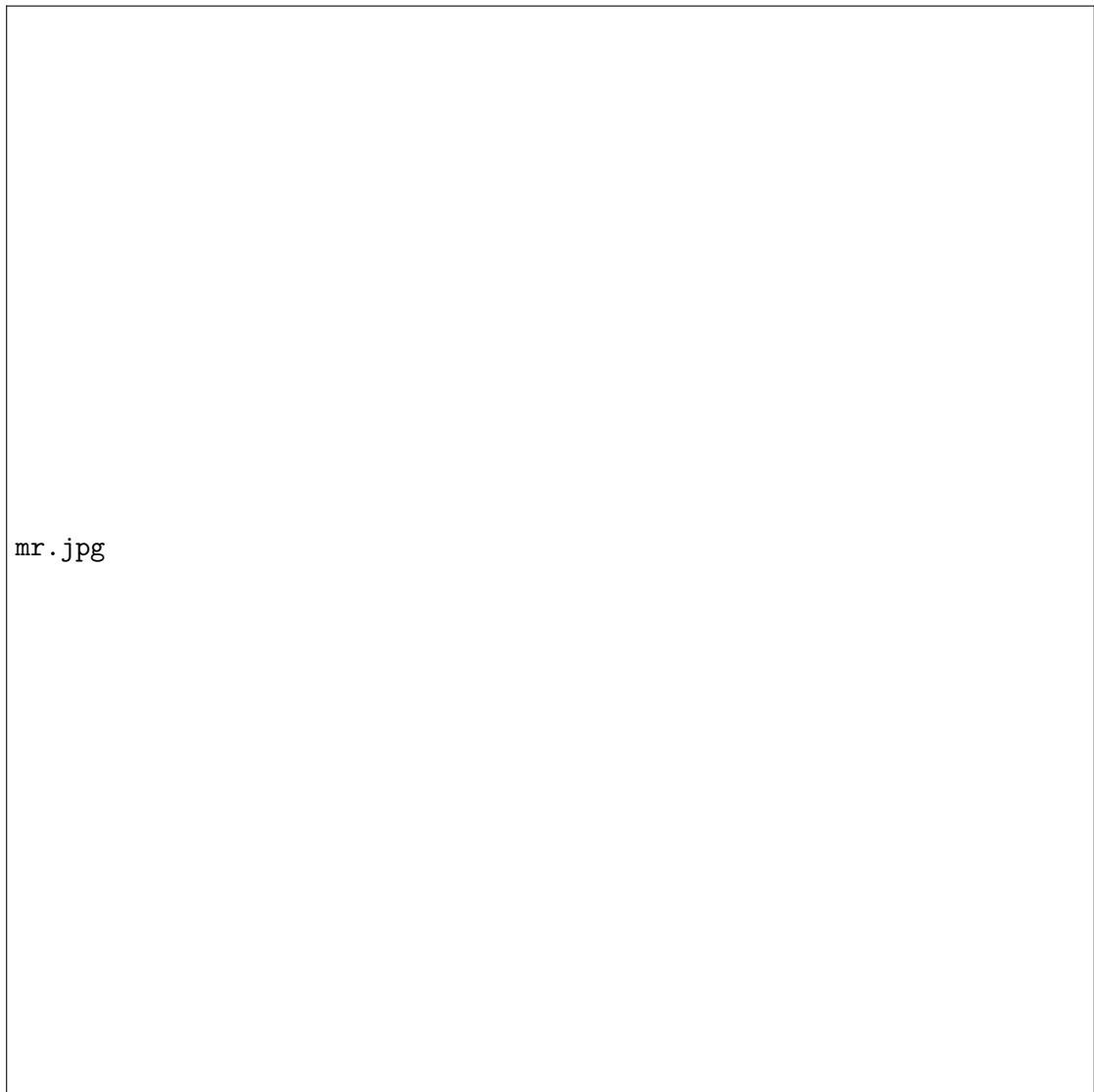


Figure 1: The Mixed Reality scale.

## 5 Texture and displacement

d) Identify and implement any distinct patterns of texture or displacement on the object using procedural and/or painted textures as appropriate. Specs 10 bumps - sine wave, abs Stalk Subsurface scattering

## 6 Adding natural variation and noise

e) Add natural variation and wear to the object using noise and related techniques. - Bumpy - Speckly

## 7 Environment map

Due to the nature of the scene of the squash resting on a table, it seemed p f) Shoot or borrow a suitable environment map (HDR is optional), and add it to your objects lighting model.

## 8 The scene

g) Place your object within a minimal scene (typically a textured ground plane should be adequate). - wood plane - one light (plus env) - 3 squashes

## 9 Depth of field

h) include motion blur, depth of field or other camera artifacts as appropriate. - Tried exposure but was fine as is As the scene is static (the squash are resting on a wooden table), it did not make sense to add motion blur. However, to create a more aesthetically pleasing image, a short depth of field camera effect was applied. This had the effect of keeping the foreground detail sharp, but blurring the furthest edges slightly to give a softer look. Although it is subtle, it gives a more natural look than when the whole image is in focus.

There was some experimentation with exposure and FOV values, however it was decided that the most aesthetically pleasing result came from keeping the exposure at the default values and a FOV of 30.

## 10 Improvements and further work

There are a number of improvements that could be made to the rendering. The first is to spend more time accurately modelling the stalk of the pumpkin, as it currently appears unrealistically sharp and straight. This was due to the main focus of the assignment being to model a round object of simple geometry, so the stalk was not the priority. The stalk would also benefit from more colour and texture variation.

The scene could be made more compelling by the addition of a background, especially one which matched the environment map.

A further exercise could be to try rendering other varieties of squash, particularly those with a different shape, such as a butternut squash.

### **Painterly**

Imitating a hand-painted style, with textured semi-random paintbrush strokes. Examples of work in this area include ?, ?, ?, ? and ?.

### **Sketched**

Imitating a sketchy style, usually monochrome, with semi-random pencil strokes and cross hatching. ? is an example of this.

### **Toon/Cell-shading**

Imitating a cartoon style, with discrete blocks of colours and limited quantisation of shading colours. ? and ? are examples of this.

- Is a mesh-based or IBR approach more appropriate?
- Is there a minimum angle between cameras needed to convincingly simulate novel views?
- Can we extract and use depth information to help synthesise the views?
- Can we achieve consistent lighting across the real and virtual elements?
- Can we extract stylistic qualities from the virtual world and automatically apply them to the video?
- Can we choose a stylistic effect and automatically apply it to both the real and virtual elements?
- Will we need new compression algorithms to deal with this system?
- How would we distribute such a system to multiple players and spectators?
- Can we playback the live free viewpoint video inside a game engine?
- Can we do any/all of this live? Which parts?

## **References**