COMP 70001 Spring 2021 Assignment 1

George Castle Zhiyuan Zhang
Email:{george.castle20, zhiyuan.zhang17}(@imperial.ac.uk)

1. Part1: HDR Processing

Zhiyuan Zhang does the first part of this coursework. It mainly discusses HDR formation from images taken with exponentially growing stops and tone mapping to produce a better HDR image result. The two sections would be discussed separately below.

1.1. Section1: HDR Formation

This section followed the instructions from the specification. The practical implementation can be described in the following steps:

1. Load each image and store them in list Z.
2. Using template to create empty image F with size 576*864*RGB. The name used in the template is wrong, which the initialisation function should swap the order of width and height.
3. Concatenate the images from Z into F. The Formula is given in the specification.
4. A center-weighting function w(z) is introduced. To meet the requirement, w(z) is defined as $w(z) = 0.5 * \sin(2\pi z - 0.5\pi) + 0.5$. Shown in Figure 1. The exposure is given in example.

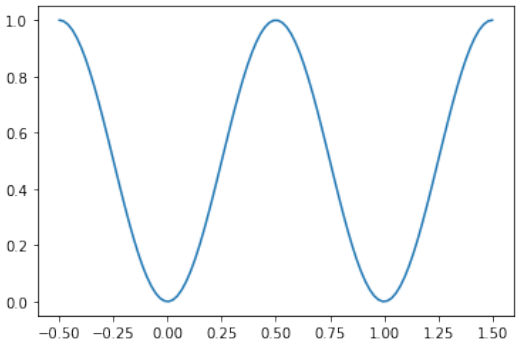


Figure 1: w function

5. A preprocessing function is applied to each pixel before adding to F. It approximates value smaller than 0.005 to 0.005 and value larger than 0.92 to 0.92.
6. Using provided saving function to save F in $F\_HDR.pfm$.

1.2. Section2: Tone Mapping

Using linear tone mapping, convert the pixels into the range [0,1]. Dr Ghosh suggested using brightness to convert all three channels. Applying brightness may miss some information(some pixels have colour larger than 1). I also tried using max value among all RGB channels. The result would be darker than using brightness, but more information is preserved. However, by experimenting, the difference can be neglected.

Brightest pixel returns brightness of 0.0347. Dimmest pixel returns brightness of 1.131e-5. Dynamic range results in 3065.

Result using linear tone mapping is shown in Figure 2.

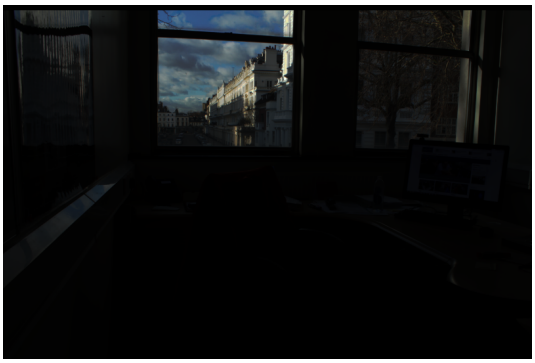


Figure 2: Linear Compressed image

The second task is to increase the exposure of the image. Due to approximation, some colour information is still lost and caused in the wrong colour. Black values cause this since we set it to non-zero, it would increment when exposure getting larger. The stops are selected from 1 to 7. The

best images correspond outdoor view, and an indoor scene is shown below:

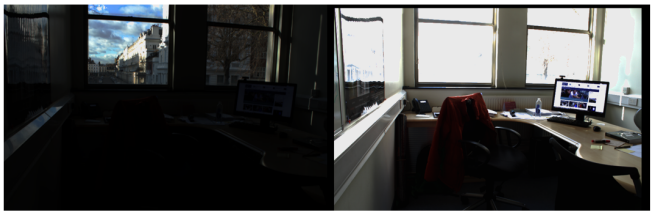


Figure 3: Best images(Left: 1 stop, Right: 5 stop)

The last task is to apply different gamma correction (1.5, 1.8, 2.2, 2.5, 3.0) to each image. The best image is selected in 4.

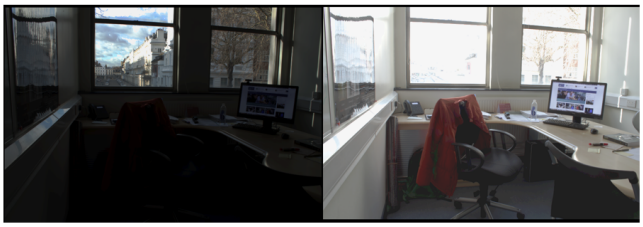


Figure 4: Best images(Left: 1 stop 1.5 gamma, Right: 4 stop 1.8 gamma)

The code and all the images generated during this coursework can be found in 70001-CW1/Part1/.

2. Part2: IBL

George Castle did the second part of this coursework. A simple procedural image-based lighting model is discussed below, along with its implementation.

2.1. Surface Normal Map

Using the provided 'CreateAndSavePFM' I generated a 512*512 square pfm image.

The algorithm first finds the coordinates for the centre (a,b) of the square by dividing the image height and width by 2.

It then iterates through all array values using a circle equation:

$$(x-a)^2 + (y-b)^2 = r^2$$

to check if the x and y values are within the bounds of a circle of diameter 511.

If x and y are outside the circle their x, y, z colour values are set to 0.

Else the x and y values are normalised into [-1,1] making the square's centre point (0,0). The z value is then calculated using the equation of a sphere with centre (0,0,0):

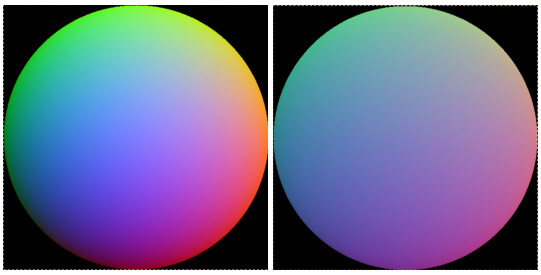
$$z = \sqrt{r^2 - x^2 - y^2}$$

mapping these values to [0,1] and setting xyz to RGB respectively produces the following spherical coordinates map shown in Figure 8a.

The surface normals are then calculated for each x,y,z vector using the half vector equation:

$$n = \frac{v + l}{||v + l||}$$

Setting the RGB colour channels to xyz respectively (after normalising to a 0 to 1 range) and saving the array as a PPM produces the following result in Figure 8b:



(a) Spherical coordinates map in RGB form (b) Surface normal map in RGB form

Figure 5: Maps

2.2. Reflection Map

To calculate the reflection map I used the provided equation of reflection:

$$r = 2(n \cdot v)n - v$$

to calculate the reflection vector r on each pixel within the sphere, using the same loop and conditional to check which x,y array values were within the sphere's bounds.

As I was using an orthographic view vector v = [0,0,1] the reflection equation, in its vector component parts, simplified to:

$$r_x = 2n_z n_x$$

$$r_y = 2n_z n_y$$

$$r_z = 2n_z n_z - 1$$

Setting the RGB colour channels to r_x, r_y, r_z respectively (after normalising to a 0 to 1 range) and saving the array as a PPM produces the following reflection map:

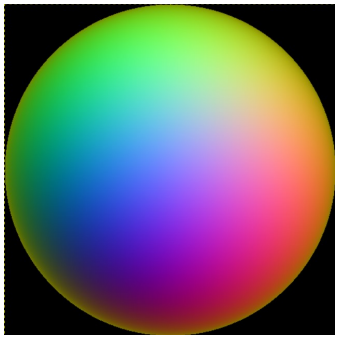


Figure 6: Reflection map in RGB form

2.3. Rasterised LatLong Spherical Map

Using the same for loop and conditional as for the previous parts the algorithm iterates through the entire array checking if the x,y values are within the sphere.

It then uses the reflection vector values r_x, r_y, r_z to calculate the spherical coordinates (θ,ϕ) using the following formulas:

$$\theta = \arccos(-r_y)$$
$$\varphi = \arctan 2(r_z, r_x)$$

(θ,ϕ) are then normalised to the 0 to 1 by dividing by 2π for ϕ as it ranges from 0 to 2π and dividing by π for θ as it ranges from 0 to π.

These values are then scaled to index in to the latlong map by multiplying θ by latitude and ϕ by longitude and taking the result's integer value.

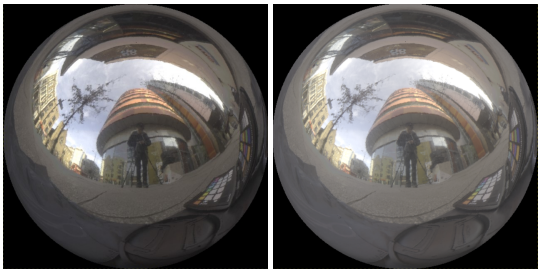
The (x,y) coordinates of the array then have their RGB values set to the RGB values of the latlong map pixel at location (θ,ϕ) producing the following PPM:

Figure 7: Rasterised spherical latlong map

Finally I adjusted the Gamma of the final image m' using the following gamma correction formula with gamma = 1.2:

$$g = pow(m', \frac{1}{gamma})$$

Values greater than 1.2 increased the luminance too much making the image appear washed out, whereas values lower than 1.2 had very dark patches which lacked detail. For my monitor the best image, which displayed bright details of the sky and dark details of the paint pallet was created using a gamma value of 1.2 shown below:



(a) gamma value 1.2 (b) gamma value 1.7

Figure 8: Gamma corrected spherical latlong maps