Computer Vision Assignment 1

Wolf Vos 10197923
Kasper Bouwens 6137547

February 2016

1 Photometric stereo

In this section the surface albedo, surface normals and the reconstructed shape are shown. This is also what the matlab script start.m outputs. The function analyses five different pictures of a round shaped object which is illuminated from different angles. This allowed us to recreate the shape of the object as shown in figure 3. In order to accomplish this a couple of steps had to be taken:

1. Calculate the surface albedo (figure 1)
2. Calculate the surface normals (figure 2)
3. Calculate the height map from the surface normals
4. Combine step 2 and 3 to reconstruct the object shape (figure 3)

Some problems arose when there was a $\frac{0}{0}$ in the code, this resulted in NaN's on a lot of places. Matlab turns Nan + 1 into Nan so therefor we changed all the NaN entries to 0.

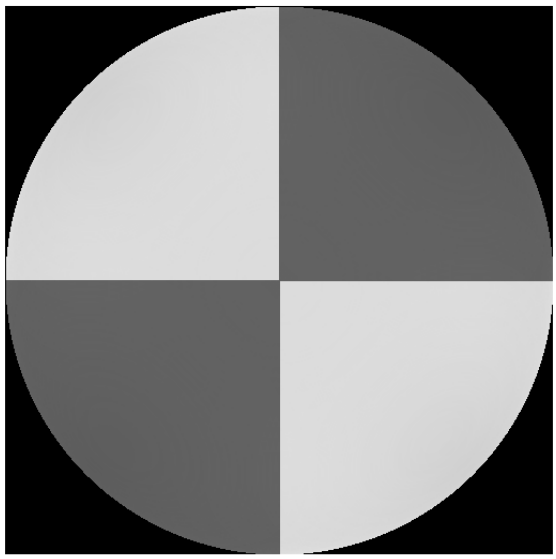

Figure 1: Surface albedo

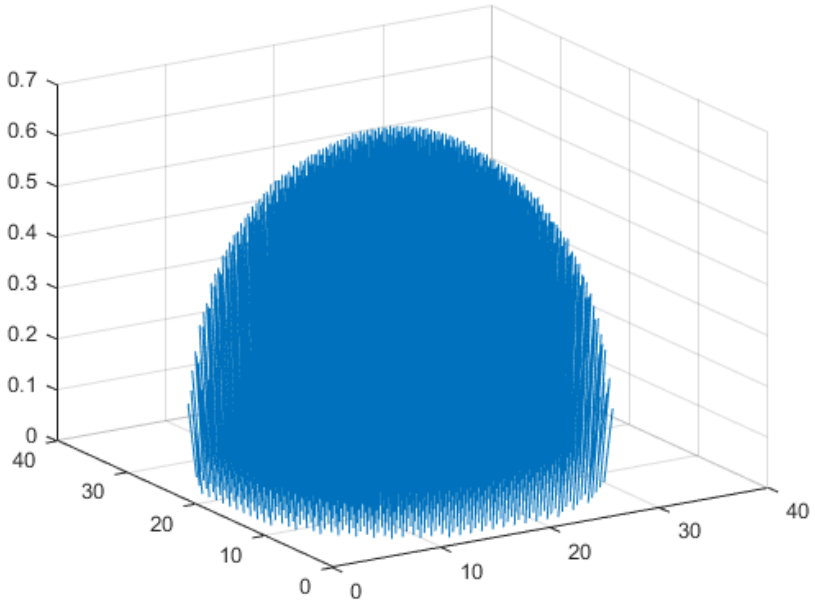

Figure 2: Surface normals

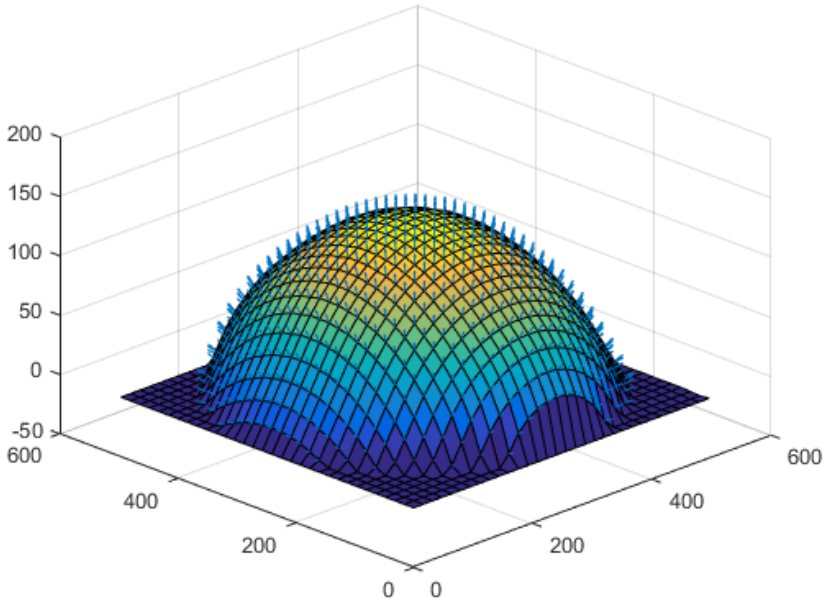

Figure 3: Reconstructed shape

2 Color Spaces

In this assignment we had to plot multiple color spaces. The first color space was the regular RGB color space. When loading an image in MATLAB the image is automaticly converted to a 3 dimensional array. The first two dimensions are the X and Y axes of the image. The third represents the three different channels. To display each channel separately the other two channels had to be set to zero. This involved creating a matrix with the same width and height of the 2D channel and replace the channels we did not want to show with this array. Combining the three arrays again would retrieve the original image. The resulting images are shown in figure 4.

The opponent colors were calculated and plotted as seperate images containing grey values The results are shown in figure 5

The seperate normalized channels were plotted by taking the normalized channels and setting the values in the other channels to zero. Due to the fact that image in matlab expects normal rgb values, the values had to be multiplied by 255 to make the normalized values visible. This resulted in some information loss as shown in figure 6.

We used the built in function rgb2hsv to convert the rgb image to hsv. We then plotted the Hue, Saturation and Values in different grey-scale images. The

results are shown in figure 7.

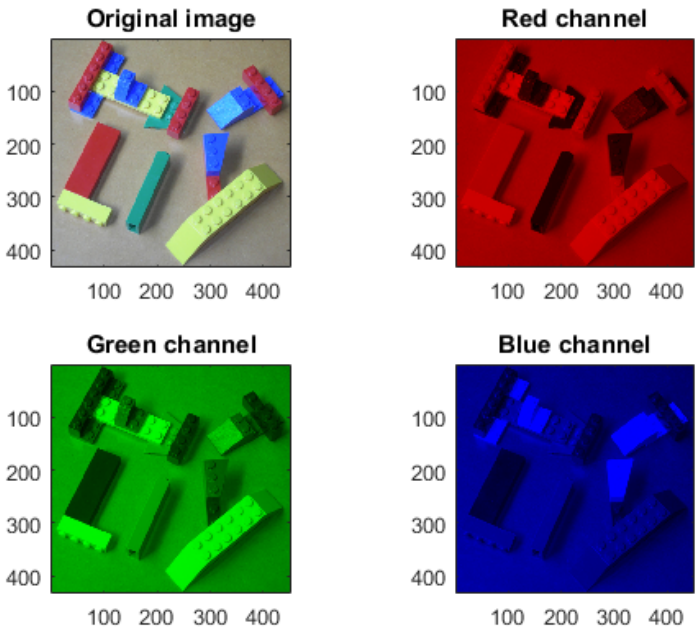

Figure 4: RGB

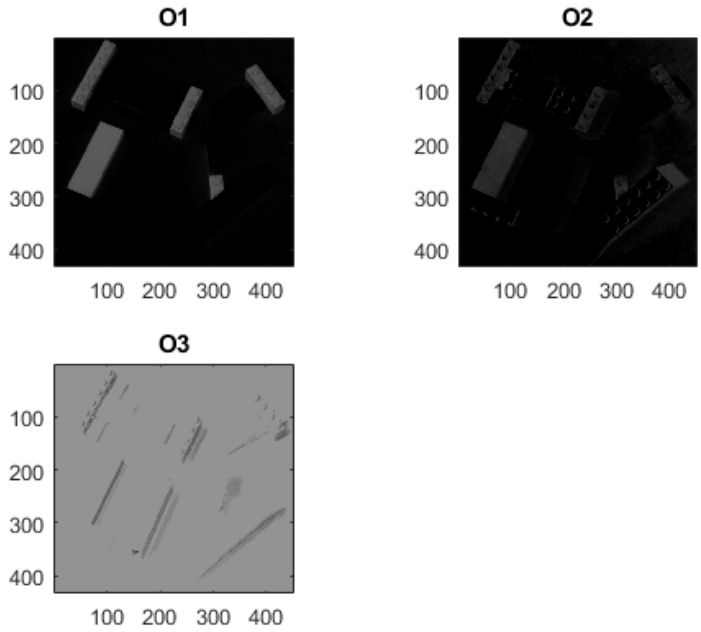

Figure 5: Opponent colors

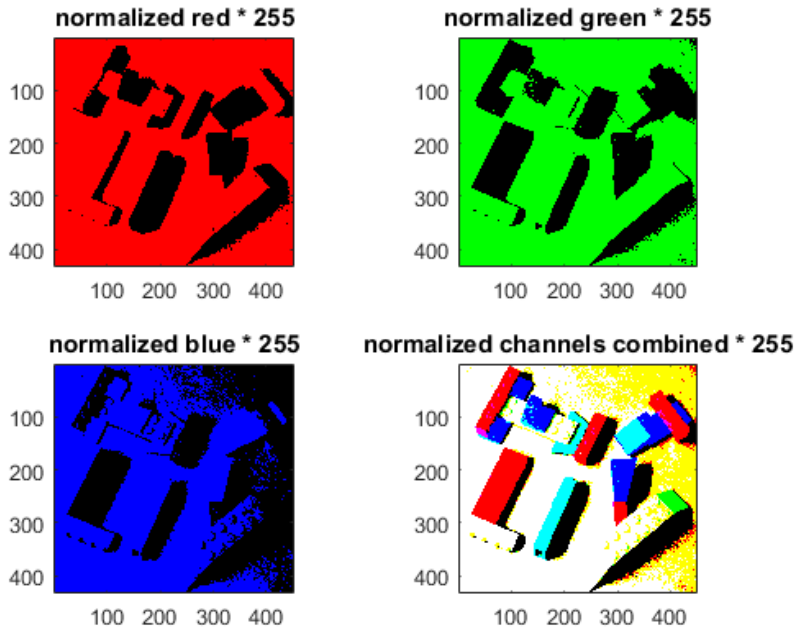

Figure 6: normalizedrgb

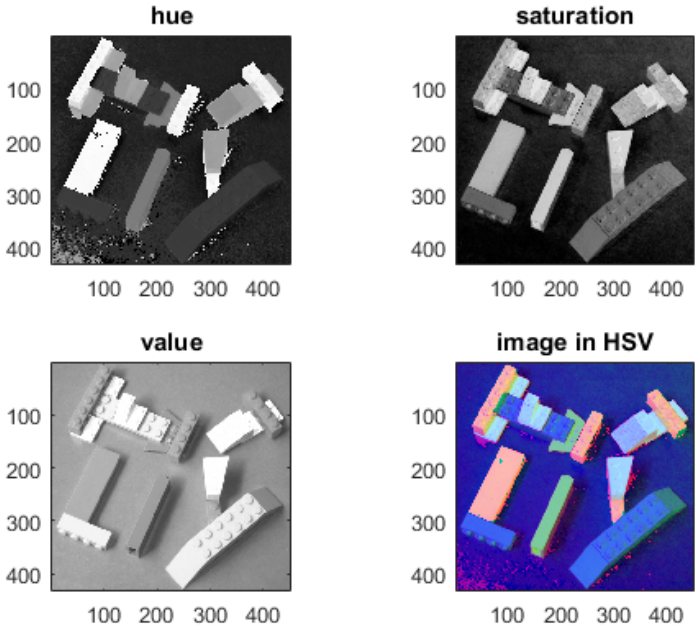

Figure 7: Hue, Saturation and Value