Assignment 1: Photometric Stereo & Colour

12055980, 12153605, 12202770

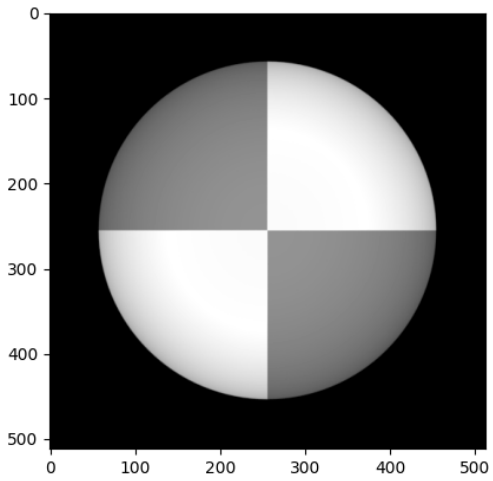
September 2021

Introduction

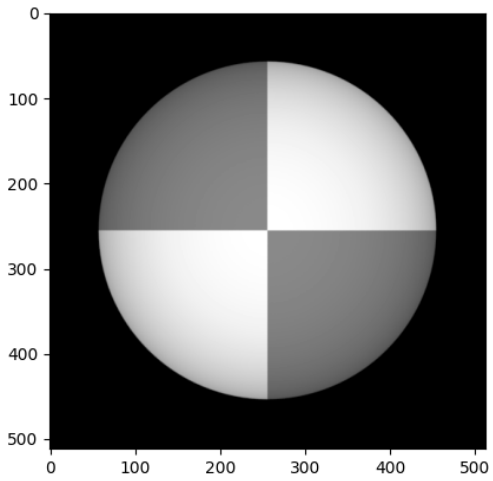
This report will look at photometric stereo, colour spaces, intrinsic image decomposition and colour constancy. In the photometric stereo, the albedo and surface normal will be estimated. The test of integrability will also be done. We will also look at the shape of the integration and experiment with different objects. In the colour spaces section, different colour spaces will be explained and conversion methods will be programmed. We will show the process of intrinsic image decomposition with examples obtained by programming. Finally, we will look at colour constancy, by implementing the grey-world algorithm.

1 Photometric Stereo

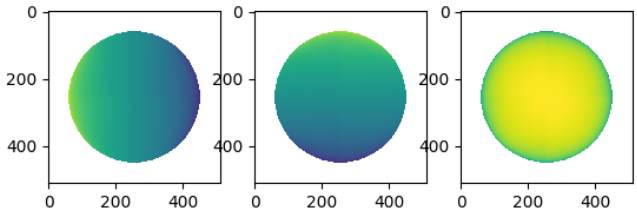
1.1 Estimating Albedo and Surface Normal



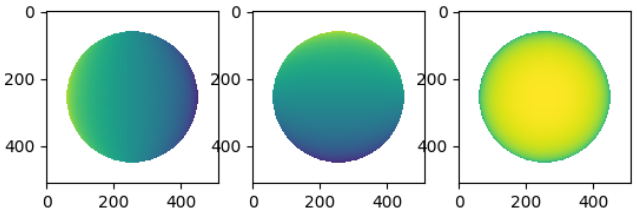
(a) Albedo map when using 5 images (with shadow trick)



(b) Albedo map when using 25 images (with shadow trick)



(c) Normal components estimated with 5 images (x, y, z) (with shadow trick)

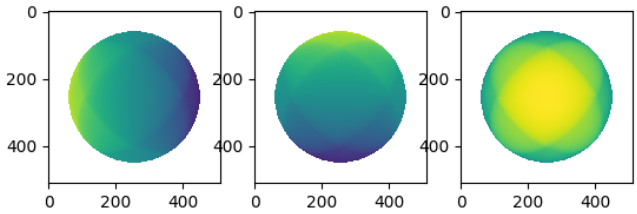


(d) Normal components estimated with 25 images (x, y, z) (with shadow trick)

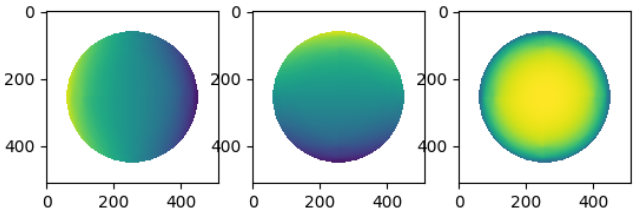
1. Figures 1a and 1c show the albedo and normal maps approximated using 5 images and using the shadow trick. Given the fact that the shape of the object in the image is spherical, one would expect the albedo intensity to decrease towards the edges of the shape. This is however not clearly visible in the approximated map.

2. Since three unknown values have to be approximated using least-squares, at least three images are needed for the estimation. The algorithm was executed using 25 images as to use as many different light directions as possible. Figures 1b and 1d show the albedo and normal maps after doing approximation with 25 images.
   Although not clearly visible, the albedo values do decrease towards the edges of the shape, unlike those when estimating with 5 images. When using the shadow trick, not much difference can be seen between the components of the normals. A difference can be seen when not using the shadow trick, which is shown in figures 2a and 2b. The normal components estimated with 25 images demonstrate a smoother underlying shape and clearly better resemble a sphere.

3. If a part of an object is shadowed in a multiple the images used for estimation, it will cause a distortion in the reconstruction. (Chandraker, Agarwal, & Kriegman, 2007). The trick used in the book will cause the pixel values to significantly decrease and effectively become zero, which is the result of squaring already small values. Figures 2a and 2b show the normal components estimated with 5 and 25 images respectively. As can be seen, the shadow trick is needed when using 5 images, as the effects of the shadows are visible in the maps of the components. Because in 4 out of the 5 images a part near the end of the sphere is darkened by a shadow, it looks as if there are two diagonal ellipses. The problem is not occuring when using 25 images, thus it can be said that the trick is needed when using 5, but not when using 25 images to estimate the normals.



(a) Normal components estimated with 5 images (x, y, z), (without shadow trick)



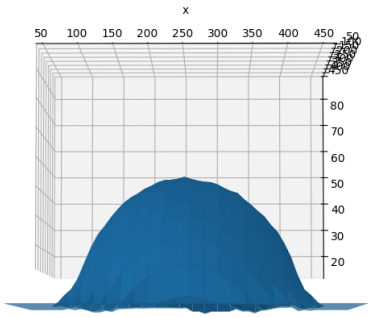
(b) Normal components estimated with 25 images (x, y, z) (without shadow trick)

1.2 Test of Integrability

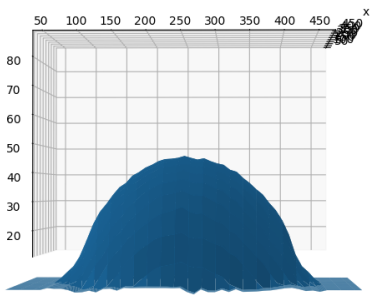
1. The derivatives of f w.r.t. x and y were calculated with $\frac{\partial f}{\partial x} = \frac{\mathbf{N_1}}{\mathbf{N_3}}$ and $\frac{\partial f}{\partial y} = \frac{\mathbf{N_2}}{\mathbf{N_3}}$, where $\mathbf{N}$ denotes the surface normal at (x, y). The second derivatives were approximated with neighbour difference. Out of bound values were chosen to be 0. When using 5 images, the squared errors differed much depending on the usage of the shadow trick. When using the shadow trick, the largest error is approximately 9.95 and the mean error is 0.005. Without using the shadow trick, those values become about 1.38 and 0.001 respectively. When using 25 images without the shadow trick, as this is not necessary, the largest error is about 1.21 and the mean error is also about 0.001. A reasonable threshold was determined to be 0.001, leading to 2011 and 1528 outliers when using 5 and 25 images respectively. The largest errors occur around the edges of the sphere. A possible reason could be due to the fact that a spherical shape represented on a discrete grid, meaning at the edge there are points where the second x derivative was taken using two points in the object and the y derivative between a point in the object and the background and vice versa. Since the background colour is totally black, this will cause one of the absolute values of the second derivatives to be substantially larger than the other, causing a high error.

1.3 Shape by Integration

1. When reconstructing the shape in column-major order, the height of the reconstruction can be seen to increase step-wise when looking along the y-direction, as visible in figure 3a. When reconstructing the shape in a row-major fashion, this occurs when looking along the x-direction, as visible in figure 3b.

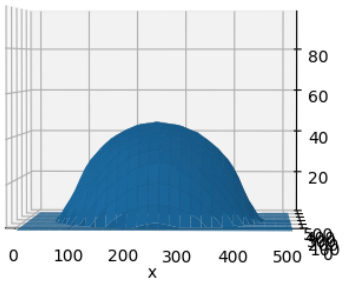


(a) Reconstruction with column-major integration. (5 images)

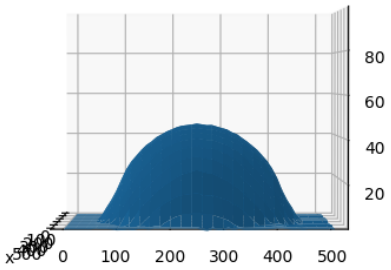


(b) Recontruction with row-major integration. (5 images)

2. Using the average of the two reconstruction methods as the final reconstruction leads to a smooth shape. The step-wise increases along the axes are no longer visible, as can be seen in figures 4a and 4b. Using the average of the two methods thus improve reconstruction quality. There is no significant difference in shape between reconstructions made with 5 and 25 images.

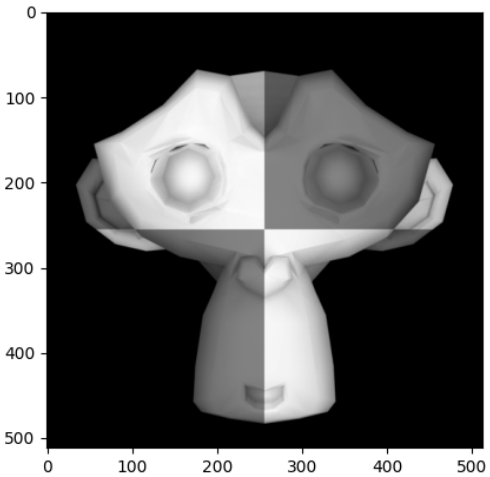


(a) Average reconstruction looking along the y-axis. (5 images)

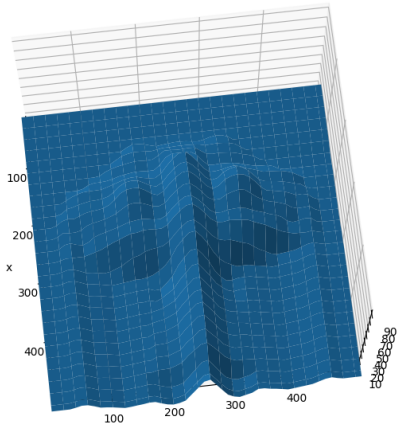


(b) Average reconstruction looking along the x-axis. (5 images)

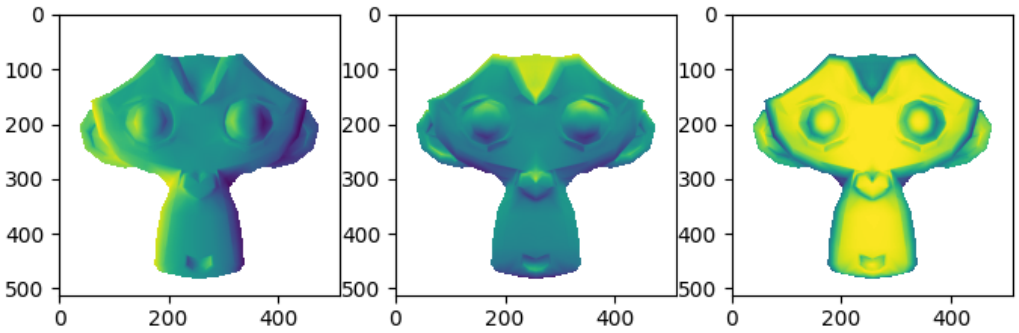
1.4 Experiments with different objects



(a) Albedo map of the gray monkey image.



(b) Reconstruction of the gray monkey image.

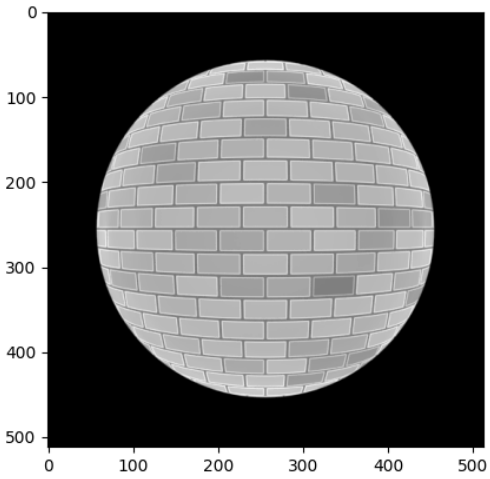


(c) Normal components of the gray monkey image.

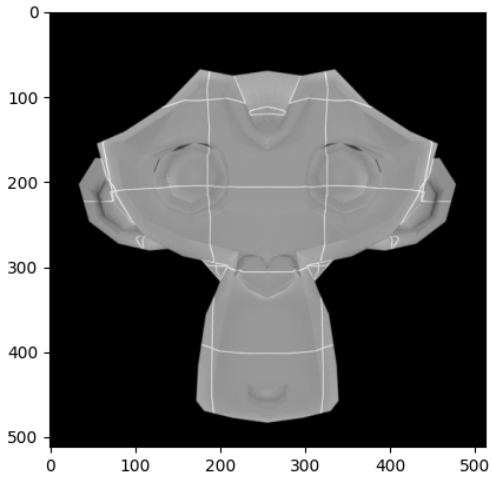
1. The errors in the monkey image occur again mostly around the edges of the shape. However, because there is some relief in the shape itself, these kinds of errors also occur locally, for instance around the eyes and ears. A reason could be the fact that in many images, parts of the shape are in the shadow. Using less images or using images containing multiple light sources (Chandraker et al., 2007) could possibly limit the amount of errors.

2. Colour images are loaded per channel and stacked in a 4 dimensional tensor. To estimate the albedo maps of the colour channels, the surface normals are estimated using the colour images converted to grayscale. Afterwards, the albedo maps for each channel are estimated based on the normals.[1] The albedo values are estimated per channel such that the squared error between the estimates and the true pixel value $(\mathcal{I}_{xy} \cdot -\rho\mathcal{V}\mathbf{n}_{xy})^2$ is minimized, where x and y denote the pixel coordinates. This is done by setting the derivative to 0 and solve for ρ, resulting in $\rho_{xy} = \mathcal{I} \cdot \mathcal{V}\mathbf{n}_{xy} / \mathcal{V}\mathbf{n}_{xy} \cdot \mathcal{V}\mathbf{n}_{xy}$. A final albedo map was formed by averaging the albedo maps of the channels.
   Figure 6 shows the results for the coloured images. Since the normals were approximated using grayscaled versions of the images, there are no noticeable errors. The albedo of the monkey image does show errors at the borders between blue and red, since the albedo values of both channels contribute to the final value at those coordinates.

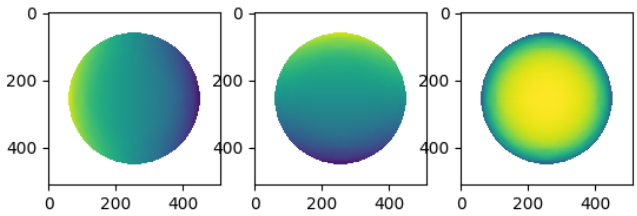
[1] https://www.cs.cornell.edu/courses/cs4670/2018sp/lec26-photometric-stereo.pdf



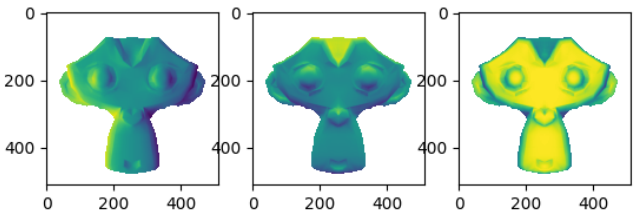
(a) Albedo map for coloured sphere.



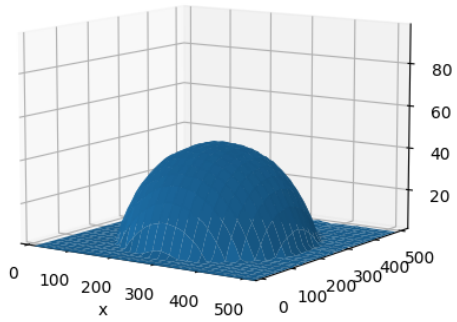
(b) Albedo for coloured monkey.



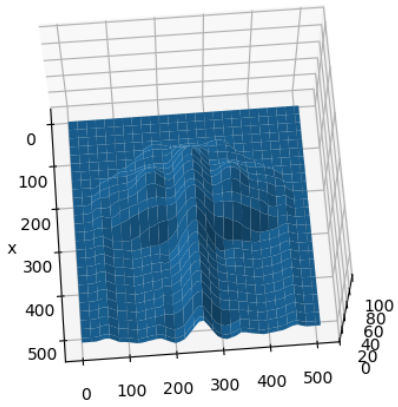
(c) Normal map for coloured sphere.



(d) Normal map for coloured monkey.



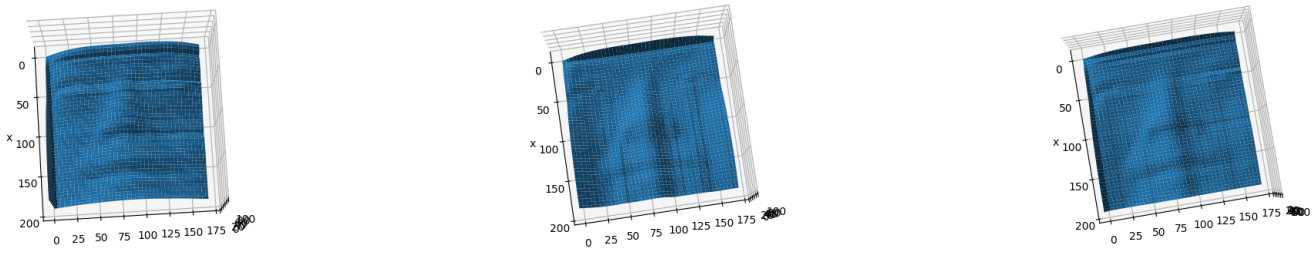
(e) Recontruction of coloured sphere.



(f) Reconstruction of coloured monkey.

Figure 6: Results for the coloured sphere and coloured monkey images.

3. Figure 7 shows the reconstructions of a face using different integration paths. It can be seen that the height of the reconstruction suddenly decreases in the middle of the face when performing row-major integration and that there is a sudden increase in height in the area beneath the right eye. Integrating in a column-major fashion will result in a more natural curve of the face but has a considerable height increase along the x-axis in the direction towards the top of the face. Taking the average of both methods does lead to a better reconstruction, although it can be said that it is not a perfect method, as height errors are still visible.



(a) Reconstruction with column-major integration.

(b) Reconstruction with row-major integration.

(c) Reconstruction with the average of row- and column-major integration.

Figure 7: Recontructions of a face with different integration methods.

Shape-from-shading methods assume the surface in question has Lambertarian reflection. Different areas of the human face can however have different reflective properties. The eyes can for instance exhibit specular reflection, as could the skin. Because of the fact that a face is not a totally Lambertarian surface, the images violate the assumptions of the shape-from-shading methods. An example of the specular reflections can be seen on the nose and in the eyes in figure 8.

Figure 8: A sample from the Yale face database

2 Colour Spaces

2.1 RGB Colour Model (3-pts)

The RGB colour model can be used to create all colours within the colour triangle. It is based on the human perception of colours. The RGB colours maximize the difference in the responses of the cone cells, they have very different wavelengths. This creates a very large colour triangle. It is also very convenient for digital colour, because every pixel has to only store three values to recreate the colour of that pixel.

2.2 Colour Space Conversion (10-pts)

The colour spaces are converted from RGB to other colour spaces using formulas and the build in function of OpenCV. To visualize all of these colour spaces, the visualize function is used. It uses the image and the colour space as the input to give the correct visualization.

2.2.1 Opponent Colour Space

To convert to the opponent colour space, the luminance, red-green channel and blue-yellow channel are calculated using the following formulas:
$O_1 = (R - G) * \frac{1}{\sqrt{2}}$
$O_2 = (R + G - 2 * B) * \frac{1}{\sqrt{6}}$
$O_3 = (R + G + B) * \frac{1}{\sqrt{3}}$
These O's together give the opponent colour space.

2.2.2 Normalized RGB Colour Space

To convert to the normalized rgb colour space, the RGB values are each normalized by deviding them by all the RGB values:
$R = \frac{R}{R+G+B}$
$G = \frac{G}{R+G+B}$
$B = \frac{B}{R+G+B}$
This gives the normilized RGB colour space.

2.2.3 HSV and YCbCr Colour Space

To convert to these colour spaces, the build in function of OpenCV is used:

```
hsv_image = cv2.cvtcolour(input_image, cv2.colour_RGB2HSV)
ycbcr_image = cv2.cvtcolour(input_image, cv2.colour_RGB2YCR_CB)
```

2.2.4 Grayscale

The grayscale conversion is done in four different ways:
Lightness: $\frac{max(R,G,B)-min(R,G,B)}{2}$
Average: $\frac{(R,G,B)}{3}$
Luminosity: $0.21 * R + 0.72 * G + 0.07 * B$
OpenCV:

```
gray_image = cv2.cvtcolour(input_image, cv2.colour_BGR2GRAY)
```

These all convert the RGB image to a grayscale image.

2.3 Colour Space Properties (5-pts)

1. Opponent Colour Space The opponent colour space is based on the human perception. It is more efficient for the human visual system. It has three components: Luminance component, a red-green channel and a blue-yellow channel.

2. Normalized RGB Colour Space Colour normalization is used for computer vision. This colour space reduces the intensity of the colours. It helps with the artificial colour vision and the object recognition. These algorithms benefit from removing the intensity values of an image, while keeping the colours.

3. HSV Colour Space, HSV stands for hue, saturation, value. The HSV approach looks more like the way people observe colour compared to RGB. The angle of a colour on the RGB colour circle is represented by hue. The amount of colour used is represented by saturation. The brightness of a colour is represented by the value parameter. HSV is closer to how people see colour compared to other colour spaces.

4. YCbCr Colour Space, YCbCr represents colours as a combination of luminance (Y) and two colour difference signals (Cb and Cr). Cb is the chroma blue and Cr is the chroma red. Storing YCbCr images is more memory efficient compared to storing RGB images.

5. Grayscale. For each pixel in a grayscale image an intensity value is stored (how bright is that given point). For a grayscale image only 1 value is stored for each pixel, while an RGB image needs to store 3 values per pixels. This makes the use of grayscale images more memory efficient.

2.4 More on Colour Spaces (2-pts)

CMYK (cyan, magenta, yellow and key (black)) is a colour space based on the primary colours. It is used for colour printing. Black is included in this colour space, because a lot of printing is done in black, and it is cheaper to already have black ink, instead of mixing CMY to get black.

3 Intrinsic Image Decomposition

3.1 Other Intrinsic Component (4-pts)

A method for decomposing an image could be by separating the image into a structural (J) and a textural (T) part, where $I = J + T$ The presence of, for example, wrinkles in fabric in the photo may be confused with reflectance, since a common assumption is that changes in reflectance are correlated with changes in chromaticity and shading is smooth. However, this change may also occur when fabric is wrinkled. By decomposing the image in a structural and textural part, the image can be smoothed by removing the texture in the image. By smoothing the image it is this way, it is easier to find parts in an image with similar reflection values. [2]

3.2 Synthetic Images (2-pts)

By using synthetic images the input and output is known. This gives you a ground truth, which makes validating algorithms that use the dataset easier. Prior knowledge about the reflectance and illumination is needed for image decomposition, when using synthetic images this information is available.

3.3 Image Formation (4-pts)

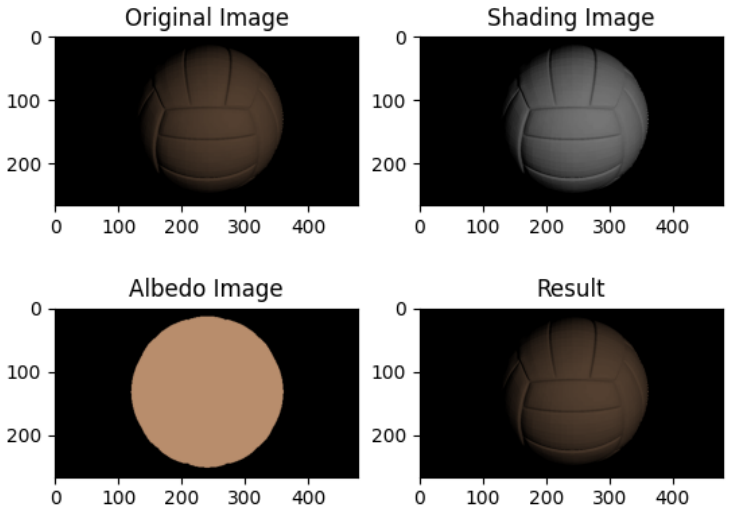


Figure 9: Image formation by multiplying the albedo and shading image ($I(\vec{x}) = R(\vec{x}) \times S(\vec{x})$)

3.4 Recolouring (5-pt)

RGB colour of ball [184, 141, 108]

[2] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5161468/

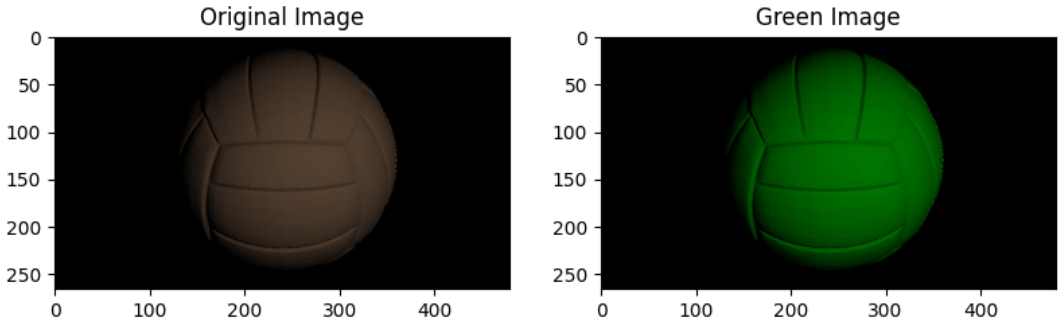


Figure 10: Image recolouring

To make the image green the original albedo image was used, the colour in the albedo image was changed to green and after this $I(\vec{x}) = R(\vec{x}) \times S(\vec{x})$) was applied to the new albedo and original shading image. The pure green image is multiplied by the ball image that has a shade over it, therefore some points on the ball appear darker and some parts appear lighter.

4 Colour Constancy

Figure 11: An image and its Grey-World-normalized counterpart.

1.

2. The Grey-World algorithm might fail when applied to images where one of the three colours (R, G, B) is occurring much less than the others. Low values will be significantly increased leading to an inaccurate and unrealistic correction of the colour channels.

3.
   - **Grey-edge assumption** Besides Grey-World there is a Grey-edge algorithm. The algorithm computes the derivatives of the colour channels, converts them into their opponent colour spaces. The distribution of the colour derivatives are aligned with the third opponent colour space axis when an image was taken with illumination from a white light. The derivative distribution is then aligned with the third opponent colour space to obtain colour constancy (Van De Weijer, Gevers, & Gijsenij, 2007).
   - **Machine learning approach** Another possible approach to maintain colour constancy is to train a multi-layer perceptron to correct the colour spaces of images (2006).

Conclusion

In the assignments we see that photos can be represented in multiple ways. These different representations are useful for performing specific operations. In addition, we see that photos can be decomposed into linear combinations, this is useful for manipulating the photos (for example, adjusting colour). Also, we can use photometric stereo to reconstruct the underlying shape of a photographed object.

References

Agarwal, V., Abidi, B. R., Koschan, A., & Abidi, M. A. (2006). An overview of color constancy algorithms. *Journal of Pattern Recognition Research*, *1*(1), 42–54.

Chandraker, M., Agarwal, S., & Kriegman, D. (2007). Shadowcuts: Photometric stereo with shadows. In *2007 ieee conference on computer vision and pattern recognition* (pp. 1–8).

Van De Weijer, J., Gevers, T., & Gijsenij, A. (2007). Edge-based color constancy. *IEEE Transactions on image processing*, *16*(9), 2207–2214.