# CV1 Assignment 1

Tycho Koster
10667687

Jeroen Terstall
10766030

February 21, 2018

## 1 Introduction

In this report photemetric stereo and color algorithms will be implemented and discussed. The goal of this report is to understand how the components of a photometric stereo work and to understand the intrinsic components of an image e.g. color.

## 2 Photometric Stereo

### 2.1 Estimating Albedo and Surface Normal

To implement the algorithm to obtain the estimation of the albedo and the surface normals, a vector I is created which stores the values of an image point of all the images in the stack. A diagonal matrix called script I is then created, which is a diagonal matrix of vector I. A shadow trick can then be applied, which zeroes out any equations from points that are in shadow. The following linear equations are solved with the use of the **mldivide** function in Matlab to retrieve the value for g. With the shadow trick applied:

$$scriptI * scriptV * g = scriptI * I$$

otherwise:

$$I = ScriptV * g$$

The albedo for that image point is equal to $|g|$, while the normal is equal to $\frac{g}{|g|}$. This is done for each image point in the images. An albedo image is the reflectance of the original image, without shadow and highlights. So the original colors of an image. So the image of the sphere gray without shadows is expected. The albedo of the gray sphere with the use of 5 images is shown in figure 1. The image shows that it is the gray sphere without any shadows, which was expected. The albedo of the gray sphere with the use of 25 images is shown in 2. There is no significant difference between the albedo images when a different amount of images is used. A reason could be that the images that are used are sufficient enough. In principle, to correctly estimate albedo and surface normal the minimum number of images depends on the quality of the image. If one image has a lot of shade and half of the image is obscured, more are needed. But if an image is perfectly clear, one image should suffice. There is also no noticeable difference when using the shadow trick, this could be because there is ambient illumination present at all points.

(a) No shadow trick applied

(b) Shadow trick applied

Figure 1: Albedo images of the Gray Sphere with the use of 5 images

(a) No shadow trick applied

(b) Shadow trick applied

Figure 2: Albedo images of the Gray Sphere with the use of 25 images

## 2.2 Test of Integrability

To test the integrability of the computation performed in the previous section, the partial derivatives p and q have to be calculated. These are obtained with the following equations given by chapter 5 of the CV: A modern approach book [6]:

$$p = \frac{a(x,y)}{c(x,y)}$$

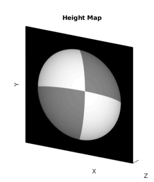
$$q = \frac{b(x,y)}{c(x,y)}$$

Where a,b and c are the measured values of the surface normal.
The second derivatives of p and q are in turn computed to perform the test of

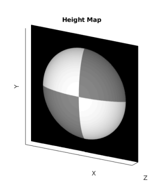
integrability. This is computed with the approximation of the second derivative by neighbor difference, which is a Matlab function called **diff**. The squared mean error is in turn computed between the second derivatives to detect outliers that are below a chosen threshold. The squared error should be zero, but because we estimate the surface normal, errors occur that both second derivatives are not equal. When we add more images to estimate the surface normal should become less, because we can more accurately estimate based on more data. This is also what is observed when running the algorithm where 25 images of the gray sphere result in 2940 outliers and 5 images of the gray sphere result in 3807.

## 2.3 Shape by Integration

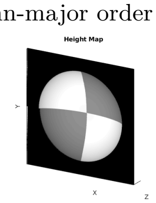
To implement the reconstruction of the surface height map, two different methods are applied, column-major and row-major order. The difference in height map between the two is shown in figure 3. With the column-major it looks like the sphere is dented, while the other looks bulged.
The average of the two combined is shown in figure 4. The height map is now back to 2D instead of 3D. The only difference that is noticeable is that the height map with more images is more smooth. As can be seen from figure 3d there is a slight indentation on the left and in figure 3c there is an indentation on the bottom. So the height map is more precise when more images are used to construct the albedo and normals.



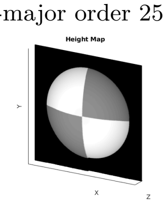

(a) Column-major order 25 images




(b) Row-major order 25 images




(c) Column-major order 5 images




(d) Row-major order 5 images

Figure 3: Height maps of the gray sphere image

## 2.4 Experiments with different objects

The albedo of the gray monkey image is shown in figure 5. This albedo contains almost double the errors as the gray sphere image. This could be due to the fact that the monkey is a significantly more complex image than a simple sphere,


(a) 5 images

(b) 25 images

Figure 4: Average height maps of the gray sphere

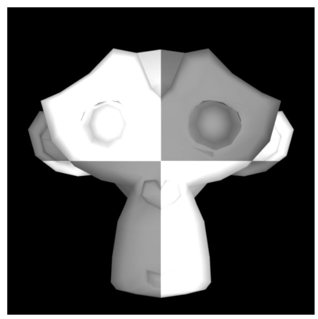


Figure 5: Albedo of the grey monkey

where the monkey has more detail. In this case, more images should also reduce the amount of errors because with more data, more accurate estimations can be made.

To make the algorithm work for RGB images, the same operations are used as with gray images. However, this was done for each individual channel separately. At the end, the average of the three channels is taken for the normal, the height map, and squared errors to return them to their original shape. The albedo is not averaged and works as is. Because we take the average of the three channels, zero pixels shouldn't be a problem and no problem is observed. The result of the colored monkey image is shown in figure 6. The results of the colored sphere image are shown in figure 7.

There were two images in the dataset which were different from the rest. In these images the light does not seem to originate from a single point. These are considered noise images. The different height maps of the images with and without the 2 noise images are shown in figures 9 and 8 respectively. There is no significant difference between the two figures, meaning the algorithm is robust to noisy images. However, there is a difference in height maps considering the integration paths. The row-major height map has a fold in the vertical direction by the nose that is dented, while the column looks like a normal face. The average of the two has a slightly dented nose, which is to be expected when averaging between the two.

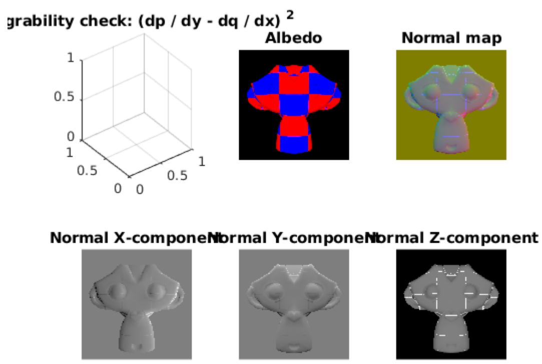



Figure 6: Results of the color monkey

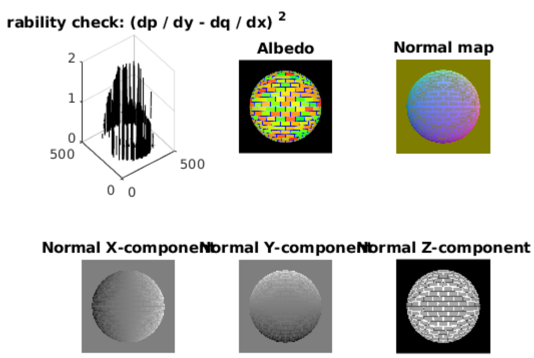
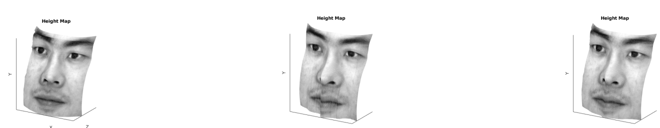



Figure 7: Results of the color sphere



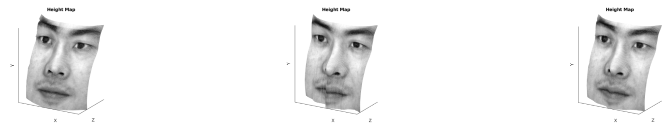
(a) Column-major (b) Row-major (c) Average

Figure 8: Height maps of the faces without the noise images

# 3 Color Spaces

## 3.1 RGB Color Model

RBG color models are used as the basis for digital cameras and photography because digital cameras use an additive color model which means that colors



(a) Column-major (b) Row-major (c) Average

Figure 9: Height maps of the faces with the noise images

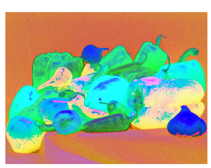
are created by combining other colors. For an additive model red, blue, and green are primary colors and can be used to create all other colors [4]. Because of this RGB is chosen over other color models for digital photography.

## 3.2 Color Space Conversion

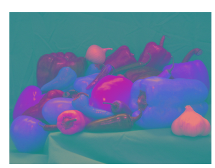
Images coded in the RGB color space can also be converted to another color space by applying calculations to each pixel. Four different algorithms have been implemented in MATLAB and are applied to the peppers image and the results are shown in figures 10a, 10b, 10c, and 10d. The image has also been converted to grayscale using four different algorithms [1]. The result is shown in figure 11.

(a) Image converted to opponent color space

(b) Image converted to normalized rgb color space



(c) Image converted to hsv color space



(d) Image converted to ycbcr color space

Figure 10: Pepper image converted to 4 different color spaces

## 3.3 Color Space Properties

Opponent color space is useful for computational applications. Opponent color space retain the hue and saturation of the original image but change the bright-

[1] https://www.johndcook.com/blog/2009/08/24/algorithms-convert-color-grayscale/

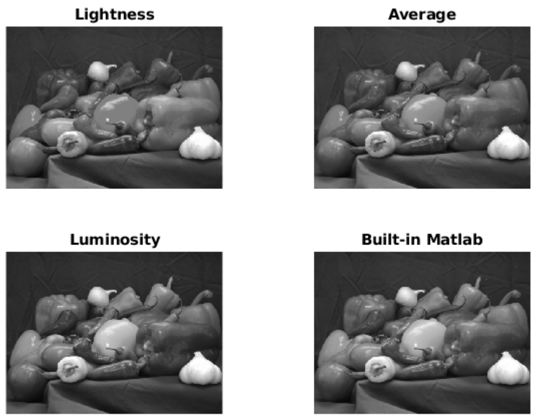



Figure 11: Image converted to grayscale

ness [5]. An example of when opponent color space is used over RGB is to categorize an object by color temperature (cool/warm) [5].

Normalized RGB space does not model the intensity/brightness of a color such as regular RGB does. Normalized RGB measures the proportion of red, blue, or green in a particular image. This might be more useful over RGB when interested in what color an object is regardless of brightness. An example would be when classifying images in its most basic form e.g. red, blue and not dark red, light red.

HSV maps rgb color space into a space which consists of hue, saturation, and value [8]. This is represented in a three dimensional cylinder where the axes represent hue, saturation, and value. This color space might be useful for color selection because it makes the process of selecting a color more intuitive than mixing colors.

YCbCr color space is often used in digital video, and image processing [3]. Luminance is represented in component Y, color information in Cb and Cr. Where Cb and Cr are blue and red difference values. The advantage of YCbCr is that it is luminance independent, because it is represented in a separate component. This might make it useful for image processing applications where this component needs to be separate.

## 3.4 More on Color Spaces

Another color space that is being used is xvYCC [7]. The paper by Matsumoto et al. describes that xvYCC can express 100% of surface color in contrast to 55% of sRGB for wide-gamut video displays. Every color visible to humans can be expressed. The use case will be wide-gamut displays for its increased accuracy in reproduction of colors.

# 4 Intrinsic Image Decomposition

## 4.1 Other Intrinsic Components

Two other intrinsic components have been mentioned besides reflectance and illumination in the paper by Barrow and Tenenbaum [2]. These components are distance and orientation. Distance is a component that holds information about how far an object is from the observer. Orientation is a vector containing the direction of the surface normal at each point in the images. Other components that might be useful are color, shape, size, etc. Everything that can be used to describe an image or objects in that image could be a component.

## 4.2 Synthetic Images

Most datasets containing these intrinsic image decomposition components consist of synthetic images. The most likely reason for this is because synthetic images are easier to train and test your algorithm on. Synthetic images can be generated to test a specific part of your algorithm. They can be made as simple and as complex as wanted to determine what part of your algorithm works and what not. Another reason could be that it is easier to generate images than to collect real-world images which fit your goal.

## 4.3 Image Formation

This section describes the results in obtaining the original image from the intrinsic components to understand how they work. Shading and reflectance will be combined to retrieve the original objects using the formula for how an image ($I\vec{x}$) is composed of reflectance ($R(\vec{x})$) and shading ($S(\vec{x})$):
$I(\vec{x}) = R(\vec{x})xS(\vec{x})$
The result of applying this in matlab is shown in figure 12.

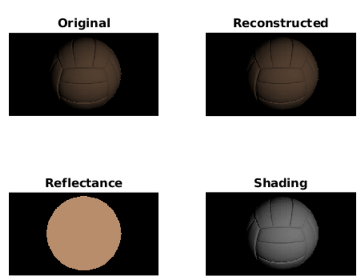



Figure 12: Reconstructed image using intrinsic components shading and reflectance

### 4.4 Recoloring

Shading and reflectance can also be used to recolor the original image in its reconstructed version. This is done by changing the true material color of the ball in RGB space. The true material color is what is reflected by the object, the reflectance of the image. The reflectance is the image without having shading affect the color making it darker or lighter.

To demonstrate this, a script in matlab is made which recreates the ball used in the image formation section in the colors magenta and green. To create the green ball, the reflectance is replaced by a matrix where the R and B values are zero and the G values are set to 255. To create the magenta ball, the reflectance is replaced by a matrix where the G values are set to zero and R and B values are set to 255. These reflectance matrices are combined with the original shading and reconstructed as described in the previous section. The result is shown in figure 13.


Figure 13: Recolored ball by changing the reflectance matrix

Although we have changed the color of the ball completely, there still is not a uniform distribution of that color over the ball. This is because of the shading of the image. Places where the shading is darker also affect the color of the ball by making it darker. Similarily, places where less shading is present, the color reflects the original color in the reflectance matrix more.

## 5 Color Constancy

### 5.1 Grey-World

To implement the grey-world algorithm the following approach was taken [2]. First, a channel from the image (red, blue, or green) is taken and summed up for all pixels in the image. This sum is divided by the total amount of pixels to get the mean of that channel. Then, the assumed average of 128, is divided by that mean and multiplied with the actual channel values to get the corrected image. This is done for the red, blue, and green channels. This resulted in the following corrected image as shown in figure 14. Here it is shown that the reddish color on the image is removed to create a more natural looking image.

[2] https://www.codeproject.com/Articles/653355/Color-Constancy-Gray-World-Algorithm


Figure 14: Corrected image and original image, using the grey-world algorithm

However, not every image will be correctly processed by this method. Here, the assumption is made that the image is taken under a white light source. This means that images with a different light will be wrongly corrected. In this case, a different assumption of average color should be selected based on the image.

Two other color constancy algorithms will be explained here briefly. An overview of these algorithms was done by Agarwal et al. [1]. The first algorithm is the Machine Learning approach using neural networks which is a data-driven approach. In this approach a mutli-layer perceptron neural network is trained to do the transformations needed to color correct an image based on the images it has seen. Images and the corrected version are given to the neural net to train on. After training, the idea is that the neural net uses the learned method to correct unseen images.

A different approach are diagonal transformation based approaches which is not a data-driven approach but a pre-calibrated approach. In this approach, the dot product of the original image and a diagonal matrix is taken to get the corrected version. This approach basically scales the original image by a factor. In this approach, the problem lies in the need to pre-calibrate the diagonal matrix to work for unknown images.

# 6 Conclusion

A deeper understanding of how the components of a photometric stereo work and the intrinsic components of an image e.g. color has been achieved.

## References


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

[2] Barrow, H. and Tenenbaum, J. (1978). Recovering intrinsic scene characteristics. *Comput. Vis. Syst*, 2.

[3] Basilio, J. A. M., Torres, G. A., Pérez, G. S., Medina, L. K. T., and Meana, H. M. P. (2011). Explicit image detection using ycbcr space color model as skin detection. *Applications of Mathematics and Computer Engineering*, pages 123–128.

[4] Boughen, N. (2004). *Lightwave 3D 8 Lighting*. Wordware Publishing, Inc.

[5] Bratkova, M., Boulos, S., and Shirley, P. (2009). orgb: a practical opponent color space for computer graphics. *IEEE Computer Graphics and Applications*, 29(1).

[6] Forsyth, D. A. and Ponce, J. (2002). *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference.

[7] Matsumoto, T., Shimpuku, Y., Nakatsue, T., Haga, S., Eto, H., Akiyama, Y., and Katoh, N. (2006). 19.2: xvycc: A new standard for video systems using extended-gamut ycc color space. In *SID Symposium Digest of Technical Papers*, volume 37, pages 1130–1133. Wiley Online Library.

[8] Sural, S., Qian, G., and Pramanik, S. (2002). Segmentation and histogram generation using the hsv color space for image retrieval. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 2, pages II–II. IEEE.