Computer Vision - Practical 1

Liang Huang - 12108812 Weitao Luo - 12064831 Yijie Zhang - 12255807
Rostov Maxim - 11808470

1 Introduction

The following work aims to provide an overview of the results from the experiments conducted with gray-scale and coloured image data. More specifically, we discuss a technique for surface map reconstruction (3D) in the first section. Later, we experiment with the different colour space and identify their advantageous and disadvantageous. The following section provides a brief introduction to image decomposition and explains simple experiments related to the intrinsic images and recolouring. The final section focuses on the applications of colour constancy algorithms which help to distinguish the true colour of the object regardless of the source light's colour (wavelength).

2 Photometric Stereo

Surface reconstruction is an important part of processing visual information. In this section, we describe an approach of acquiring a surface map of an object by reconstructing the information obtained from a number of images of this object taken under different lighting. By fixing the position of an object in space and varying the illumination around it, we can try to create a realistic map of this object in 3D space. This approach is based on estimating the light reflectance at different parts of the object and, consequently, shades created by that object. Having a set of images, we can deduct albedo and normal maps by making several assumptions. We assume that light reflects equally at every side (Lambertian surface) and that the object surface has a constant albedo everywhere.

Considering that we can estimate the height as a function of x and y: $[x, y, z] => [x, y, f(x, y)]$ one can estimate albedo and normal of each pixel ($P = (x, y)$) using the 'radiance equation' below:

$$I(x, y) = k * \rho(x, y) * N(x, y) * S \tag{1}$$

Here, I is intensity value for a pixel, S is the light source vector and N is the normal vector to this pixel. Normal vectors are possible to estimate under the assumption that we have a static camera and object as well as constant albedo (and no specularities) for each pixel.

After manipulating the equation 1. We can calculate the normals and albedo matrices by solving systems of equation for each pixel by least squares estimation.

Estimating Albedo and Surface Normal

Surface albedo is defined as the ration between radiosity reflected from the object and its irradiance (amount of energy 'absorbed' by the object). In our experiments albedo is derived for each pixel as euclidean norm of the normal vector to that pixel. Considering, we have images under different light source, we expect that the albedo matrix (image) is a representation of the reflected light intensity at each particular pixel. Different colours reflect light in a different way at various angles. Thus, albedo image should indicate the object's boundaries and its interior as well as the different colours the object consists of. After running the experiments, we find that albedo image clearly separates the object and its shape by giving different colours to pixels on the on the object: pixels with the same colour have also the same colour on the albedo image, see Figure 2.

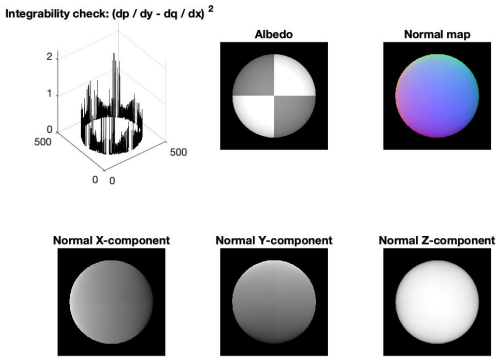

Figure 1: Shadow Trick On.

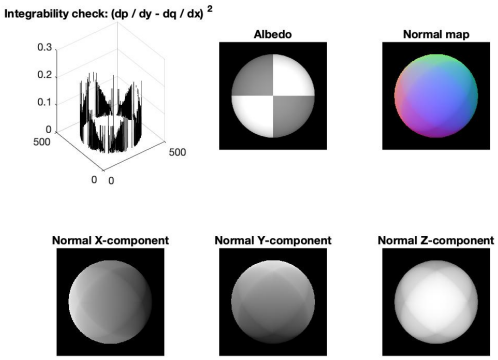

Figure 2: Shadow Trick Off.

For the estimation of the surface normals, we need to consider several images. One would require at least three images of an object taken at a fixed angle with the light source's direction changed each time. This would give a crude estimation of an object shape at each perspective (according to each direction). This comes from the fact that we need at least three equations to estimate x, y, and z.

We first estimate the normal map for 5 gray scale images of a sphere. We see that already 5 images produce an adequate normal map with or without a shadow trick, see 2. Normal map should ideally differentiate various depth, height, and width values on the (3D) image with different colours (when all three components x, y, and z are combined together). Possibly, low complexity of the object on the images plays a role. Incrementally increasing the number of images taken for the estimation does not reduce the SE-outliers (discussed further), and does not significantly change the normal maps.

Shadows in the image represent the pixels that were not exposed to the light. In this case, we use the point source with distance, which means that substantial regions of the surface may be in shadow for one or the other light direction. To calculate normal and albedo from many views, we need to build a vector for each image pixel, each vector contains all the image brightness observed at that point for different sources. Therefore we have: $i(x, y) = Vg(x, y)$. In theory, we need at least three views to solve the linear system for albedo g(x,y). However, if an image pixel is always in the shadow, then we are not able to accurately estimate the albedo at that point. Because the image brightness observed at that point is zero, which is not helpful for solving the linear system.

To deal with shadows, a simple trick is to form a matrix from the image vector and multiply at both sides of the $i(x, y) = Vg(x, y)$ equation, which will zeroes out the contributions from points that are in shadow, because the relevant elements of the matrix are zero at points that are in shadow.

As we can see in the Figure 2, without shadow-trick, some shadow boundaries were shown on the object's surface. It is because that the shadow pixels were taking into account when we calculate the surface normals, so the gradients of the shadow area were decreased.

Test of Integrability

After computing the normals, we can estimate how the height dimension vary with the values of x and y. As by definition, the out-ward pointing normal vector is represented as follows: we can calculate df/dx and df/dy from the values of the normal vectors. These values will be consequently used to calculate the height at each pixel by integrating over x and y until the values x_p and y_p of the pixel are reached.

To see whether our surface normals are estimated correctly with the available images, we run the Integration

$$\boldsymbol{N}(x, y) = \frac{1}{\sqrt{1 + \frac{\partial f}{\partial x}^2 + \frac{\partial f}{\partial y}^2}} \left\{ -\frac{\partial f}{\partial x}, -\frac{\partial f}{\partial y}, 1 \right\}^T$$

Table 1: SE-Outliers at 0.005 threshold

Dataset	Sphere5	Sphere25	SphereColour
SE Outliers Shadoff	1689	1614	1623
SE Outliers Shadon	2369	1831	1784

Test. The order in which the derivatives w.r.t x and y (when performing the second order differentiation of f) should not matter as long as our computed numerically df/dx and df/dy were estimated correctly.

The Integration Test shows that taking more images into account helps to reduce the standard error when utilizing the shadow trick. When we choose to use shadow trick we reduce the amount of data that we take for the estimation of the normals. In order to compensate for that data loss, we would want to add more images as to increase the number of equations used in the estimation of the normals, and, consequently, the derivatives.

In the case when we choose to not use shadow trick, the additional images do not contribute that much to the reduction of SE-outliers. This is logical because if we already had 5 images where the direction of light did not significantly overlap between images (different V vectors), we could estimate x, y and z without adding any (possible) noise from the other images.

Shape by Integration

After conducting the checks, we can perform the numerical integration of our derivatives to arrive to the heights (z) for our images.

There are different ways one can perform numerical integration. We need to give an initial c ($c = 0$) value to start integration with. Then, we can choose to start integration along x or y axis, filling in the first row or column. Based on that choice, we can then sum the derivatives along y or x axis. Additionally, we can choose to take the average values of integrals over row and column paths. Depending on the choice of constructing the surface map, we can get different results for the final 3D image of the object. Looking at the 2-D perspective (X and Z axis) of the reconstructed sphere, we see that 'row' mode produces visible shift of height values along x-direction, while the shift is not visible for 'column' mode and slightly less visible for 'average' mode (see Figure 2).

Experiments with different objects

Further experiments show that monkey image reconstruction is expected to comprise more albedo errors. Monkey shape is more complicated than a sphere shape having more and sharper edges. This leads to higher difficulty when approximating the normals and derivatives, because the surfaces are not so smooth which causes problems such as self-occlusion [12].

Correct albedo error estimation would require acquiring ground-truth alebedo map (see for example [11]). This is not available, therefore, we experiment with SE-errors which are influenced by the same factors.

As seen in section 2, SE errors decreased when we stopped using the shadow trick. We experimented with reducing the number of images for monkey object by deleting the images that had bigger shadows areas (darker images) and then turning the shadow trick on and off (see Table 2). It could be seen that decreasing the number of monkey images does not help and, opposite to reducing the noise, we increased the number of errors.

Figure 3: ‘Row‘ surface map construction

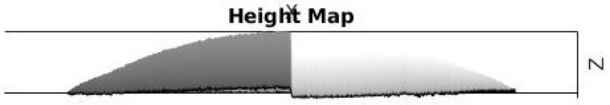

Figure 4: ‘Column‘ surface map construction

Figure 5: ‘Average‘ surface map construction

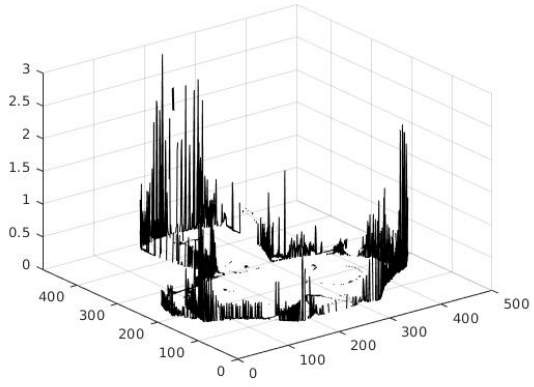

Figure 6: MonkeyGray SE errors @ 0.005 threshold

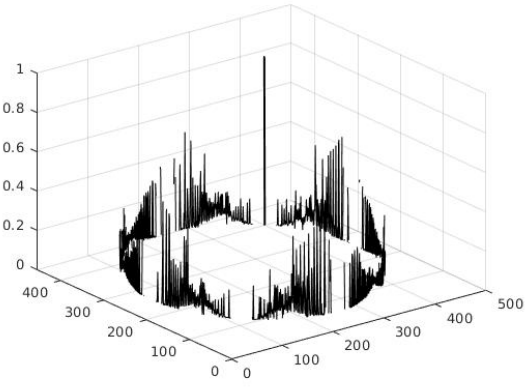

Figure 7: SphereGray25 SE errors @ 0.005 threshold

Table 2: SE-outliers @ 0.005 threshold. Starting with 121 images, then, deleting darker images.

Dataset	MonkeyGray121	MonkeyGray90	MonkeyGray82
SE Outliers shadoff	3626	3734	3798
SE Outliers Shadon	4546	5003	5203

Experiments with coloured images

The given data also includes the colour images that consist of three channels (RGB). In order to experiment with these images the initial implementation of the code had to be slightly adjusted. One has several options as to how to treat 3-channel images when reconstructing the surface maps. For example, we could treat each channel as a separate image when estimating the albedo and surface normal maps, or combine (e.g. sum, average, weighted average) the channels into a single one for each image (this might lead to loosing some useful image information). We employ the former approach and see each channel as a separate image of an object.

Constructed height ('average' mode) map is presented on the Figure below. We can also more vividly see the surface distortions for the height map (see Section 2).

Figure 8: Shadow Trick is on G

It is also noticeable how the black pixels and the shadow trick affect the images reconstruction. Ignoring the black pixels (picture on the left) by using the shadow trick produces the lighter colour schema (also when putting the colour bakc).

Face Reconstruction

Photometric stereo can be used to reconstruct human faces. Experiments were conducted to produce a 3D map from photos of a human face. Shapes were first constructed with the use of the shadow trick. It was noticed that then the reconstruction maps have several 'outliers' pixels which disturb the image (see appendix). They stem from fact that shadow trick skips the information from the black pixels (assumes they are shadows) and, hence, dark eyebrows are considered partly as shadows. Additionally, specularities in the eyes violate the assumptions of Lambertian surface. These factors disturb the picture when we start to add the derivatives together when making the height maps.

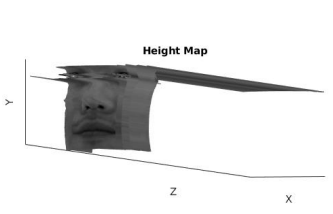

Figure 9:

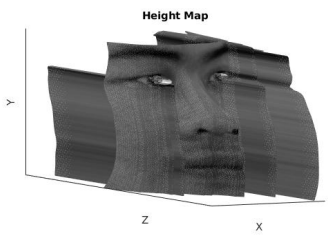

Figure 10:

Thus, in order to make the height map we disregard the shadow trick and produce the images of height maps as shown in Figure 22.

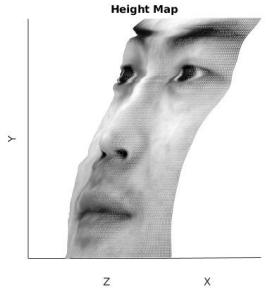

Figure 11: .

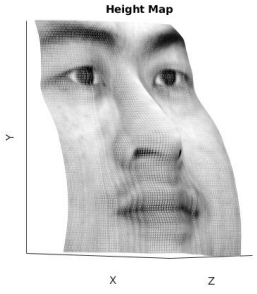

Figure 12: .

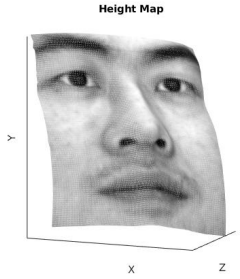

Figure 13: .

In case of the face data-sets, several assumptions of photometric stereo are violated by the images. First, you can notice that the person sometimes moves his eyes and, hence, we do not have a static images. Next, one can also notice that there are specularities on the pictures which disturb the assumptions of Lambertian reflection model. More generally speaking, the images of the faces are not constructed under the simulated environment where lighting (shading), object position and surface properties are not directly controlled. This makes it difficult to use photometric stereo techniques.

3 Color Spaces

RGB colour model

RGB color model has solid theory based on the human perception of colors Trichromacy), which is the possessing of three independent channels for conveying color information, derived from the three different types of cone cells in the eye. Similarly, modern cameras have three sensors (phosphoric elements) that each capture some range of visible spectre waves (e.g. a sensor captures intensity of the red light) coming through a camera lens. Therefore, digital cameras use RGB color model to produce pictures that human can perceive well.

To capture the full RGB color image, there is a method called Single-shot capture systems. It use three separate image sensors (one each for the primary additive colors red, green, and blue) which are exposed to the same image via a beam spliter. [1]

Color Space Conversion

Alternative to RGB representation of colour, we can employ Oponent colours. Opponent Color Space (OSP) has three components: luminance component, red-green channel and blue-yellow channel. OSP suggests that people don't perceive redish-greens, or bluish-yellows, because human vision system process the color signals in an antagonistic manner. Thus, the opponent process theory accounts for mechanisms that receive and process information from cones [2], see Figure 6.

Next, we could use Normalized RGB Color space. nRGB divides each channel's value by the sum of the pixel's value over all channels. By doing that, we can remove the distortion caused by lights and shadows in the image effectively [3], see Figure 7.

Yet another model is HSV Color Space. It stands for Hue, Saturation and Value, which is a projection of the RGB color cube onto a non-linear chroma angle, a radial saturation percentage, and a luminance-inspired value. Such a decomposition makes it good application in color selection tools [4], see Figure 8.

RGB signals are not efficient for storage and transmission, since it has many redundancy [5]. YCbCr is a approximation to color processing, where the primary color are processed into perceptually meaningful information.

By doing this, subsequent image/video processing, transmission and storage can do operations and introduce errors in perceptually meaningful ways [6].

The other common model is Grayscale space. It can represent an image using one color channel, the image will lose some color information. However, compared to RGB, Grayscale can save 66 percent storage space for storing an image [7]. In Figure 9, we demonstrate four methods to transform an image from RGB to Grayscale.

CIELAB The intention of CIELAB (or L*a*b* or Lab) is to produce a color space that is more perceptually linear than other color spaces. Perceptually linear means that a change of the same amount in a color value should produce a change of about the same visual importance. This space is commonly used for surface colors, but not for mixtures of (transmitted) light [8].

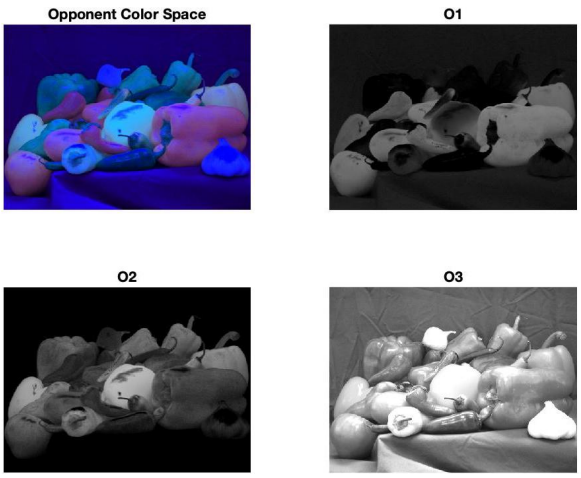

Figure 14: Visualize Opponent color space and O1, O2 and O3 channels separately.

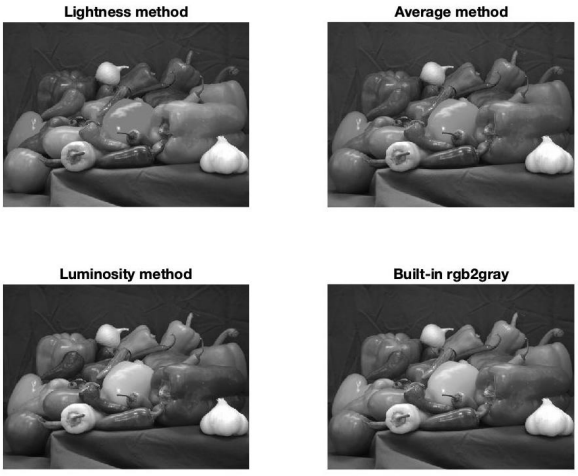

Figure 15: Visualize Normalized RGB color space and r, g and b channels separately.

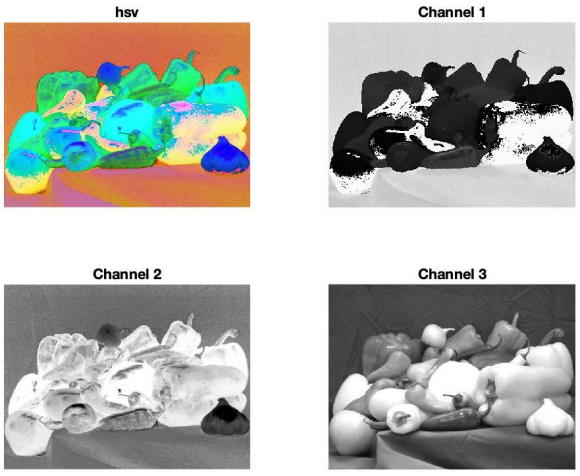

Figure 16: Visualize HSV and Hue, Saturation and Value channels separately.

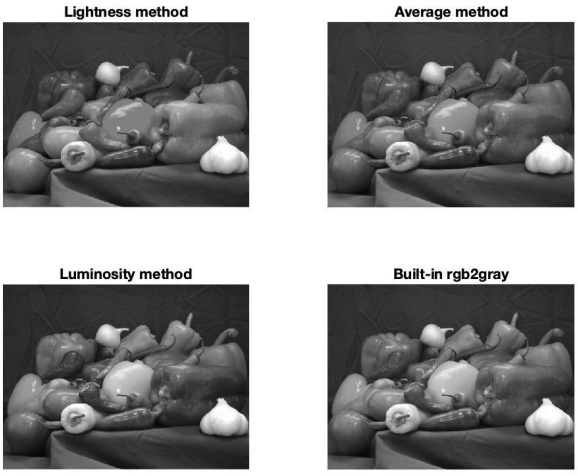

Figure 17: Four methods for transforming image from RGB to Gray.

4 Intrinsic Image Decomposition

Other Intrinsic Components

The observed images can be decomposed into several (intrinsic) components. These components include shading (illumination), depth, reflectance (albedo), and illuminant color [9]. Additionally, we can add a number of more 'fine-grained' components which can relate to either of the 'major' components described above. For example, we can distinguish between specular and Lambertian reflectance, and interreflection [9].

In the experiments conducted by Innamorati et.al [10], we see that images are automatically decomposed into multiple intrinsic layers that then can be handled separately to, for example, adjust the shades/colour of the object to a changed scene on the image (Photoshop). Noticably, authors also include oclusion in their decompostion.

Figure 18: Decomposing to intrinsic components

Synthetic Images

It is very difficult to gather ground-truth intrinsic images since intrinsic image decomposition is an ill-posed and under-constrained problem. For the real data, collecting and generating ground-truth intrinsic images require excessive effort and time. We need to separate intrinsic images step by step from the original image with very delicate settings in a fully controlled lab environment.

At this moment, the only existing data set with real world images and corresponding ground-truth intrinsic images contains as few as 20 object-centered images. As a result, intrinsic image decomposition research is dependent on synthetic data sets.

Image Formation

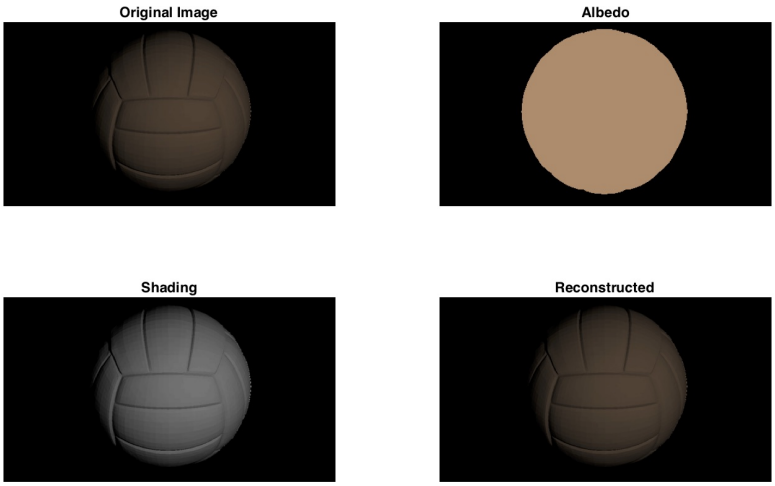

Figure 19: Reconstruct the original ball.png image

As it is shown in the 19, we can reconstruct the original image from its intrinsic images using albedo and shading. The original image $I(\vec{x})$ can be reconstructed by calculating the element-wise production of the albedo $R(\vec{x})$ and shading $S(\vec{x})$ following the equation

$$I(\vec{x}) = R(\vec{x}) \times S(\vec{x})$$

Both of the images are converted to double precision using im2double() before we can calculate the element-wise production of these two matrices directly. The reconstructed image looks exactly the same as the original one.

Recoloring

1. The true material color of the ball in RGB space is (184,141,108) in this case. Since the albedo is the color of the object, this uniform value could be extracted from the ball_albedo.png.

2. After recoloring the ball image with pure green (0,255,0), the original ball image and the recolored version are shown in 20.

First, we found all the pixels in the albedo that has color (184,141,108) and replace their color with pure green (0,255,0). After having the new recolored albedo, we calculate the element-wise production of the recolored albedo and shading to generate the recolored image.

Figure 20: Recoloring

3. The color distributions over the object do not appear uniform. This is because only changing the color of the albedo does not account for diffuse interreflections. This may result in inconsistent shading of the recolored image. In this case, the color distributions are not as we expected to be uniform.

5 Color Constancy

Grey-World

The original image and the color-corrected one are shown in figure 5. We can see that Gray-World actually corrects the reddish image on the left and restores it to a more natural color, see the image on the right.

The main assumption behind the Grey-World Algorithm is that the average of all the colors in an image is a neutral gray and the assumption only holds when there is a sufficient amount of color variations [13]. There is no clear measure of sufficiency for the condition, however, it could somehow be examined visually. For example, when an image is consisted of lots of big blocks in one single color, the Grey-World Algorithm might fail. As we can see from figure 5, after processing, the true color of white (left) becomes a lot bluish (right).

As for the the idea behind machine learning approaches, it is pretty straightforward: learning and predicting. Particularly in the context of color constancy, the algorithms first learn the mapping of the chromaticity space in the orginal dataset to the desired chromaticity space after the correction. Then the algorithms can make the predictions. Machine learning approaches can be applied using various different algorithms and the biggest challenge should be the large dataset and training time needed [13].

Figure 21: Grey-World Algorithm performs well with sufficient color variation in the image.

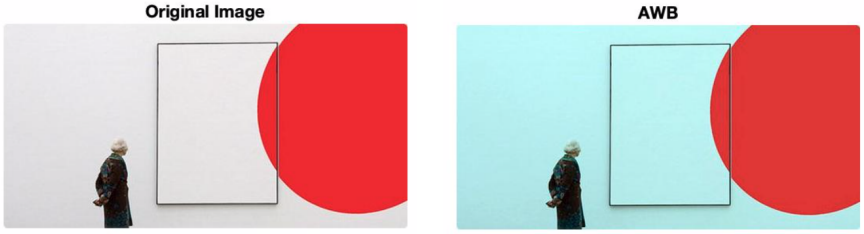

Figure 22: Grey-World Algorithm fails with insufficient color variation in the image.

6 Conclusion

From the experiments we see that surface map reconstruction becomes increasingly less accurate when the assumptions of Lambertian surface, static image and object fail. We notice that RGB colour space can be substituted by the other colour spaces depending on the use case. Also, we find that the Grey-World Algorithm fails when there is not much color variation in the image and new color constancy algorithms should be applied.

References

[1] Adams, J., Parulski, K. and Spaulding, K. *Color processing in digital cameras.* IEEE micro, 18(6), pp.20-30, 1998.

[2] Leo M. Hurvich and Dorothea Jameson *An Opponent-Process Theory of Color Vision* Psychological Review, Vol. 64, No. 6, 1957.

[3] Jian Yang, Chengjun Liu and Lei Zhang *Color space normalization* Pattern Recognition,vol 43, 1454–1466, 2010.

[4] Edith A Feisner and Ronald Reed *Color Studies* Fairchild Books, New York, 2014.

[5] Joanna Marguier *Exploiting Redundancy in Color Images* PHD THÈSE NO 4582, ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE, 2009.

[6] Saarinen, Kari. *Comparison of decimation and interpolation methods in case of multiple repeated RGB-YCbCr colour image format conversions.* In Proceedings of IEEE International Symposium on Circuits and Systems-ISCAS'94, vol. 3, pp. 269-272. IEEE, 1994.

[7] Christopher Kanan and Garrison W. Cottrell Color-to-Grayscale: Does the Method Matter in Image Recognition? 2012, doi: 10.1371/journal.pone.0029740

[8] Connolly, C. and Fleiss, T. *study of efficiency and accuracy in the transformation from RGB to CIELAB color space* IEEE Transactions on Image Processing, 6(7), pp.1046-1048, 1997.

[9] S. Beigpour, M. Serra, J. van de Weijer, R. Benavente, M. Vanrell, O. Penacchio and D. Samara *Intrinsic Image Evaluation on Synthetic Complex Scenes* IEEE International Conference on Image Processing (ICIP'2013), 2013.

[10] Carlo Innamorati, Tobias Ritschel, Tim Weyrich Niloy and J. Mitra *Decomposing Single Images for Layered Photo Retouching* Eurographics Symposium on Rendering, Vol. 36, 4 , 2017, DOI: 10.1111/cgf.13220

[11] Soma Biswas and Rama Chellappa *Pose-robust Albedo estimation from a single image* Conference Paper in Proceedings, CVPR, IEEE, July 2010

[12] Sk. Mohammadul Haque, Avishek Chatterjee and Venu Madhav Govindu *High Quality Photometric Reconstruction using a Depth Camera* The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 2275-2282

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