Assignment 1
Photometric Stereo & Colour

Maartje de Jonge
0194107
University of Amsterdam
maartjedejonge@gmail.com

Lea van den Brink
10909419
University of Amsterdam
leavdb@hotmail.com

Introduction

Computer Vision is a still growing field that is gaining more and more popularity over the years. In this first assignment, we experiment with photometric stereo, colour spaces, intrinsic image decomposition and colour constancy.

First, in Section 1 we apply the photometric stereo algorithm to synthetic objects as well as a real object (human face). We discuss the resulting albedo, normal map and reconstructed surface. Next, in Section 2, we show different colour spaces for an image and discuss their merits. Then, in Section 3, we decompose an image into its formation components, namely albedo and illumination. Using these components we reconstruct the image with a different albedo for the shown object. Finally, in Section 4 we apply the grey-world algorithm to manipulate an image so that it appears to be taken under a white light source which results in more natural looking colours.

1 Photometric Stereo

We implemented the photometric stereo algorithm to reconstruct the albedo and surface height map from a set of images in a fixed view under different illuminants. First we calculate the albedo and unit normal for each pixel by solving a linear system of equations. Then we calculate the partial derivatives of the height function from the normal vectors. Finally we reconstruct the surface by integrating over these partial derivatives.

1.1 Estimating Albedo and Surface Normal

- **Question 1.1**
  Figure 1 (left) shows the albedo image reconstructed for the five grey taint sphere images. What we expected to see in this image is a circle with uniform light grey and uniform dark grey in the four quarters of the circle. The image indeed shows these quarters. However, the image also shows a thin, slightly darker circle at the exterior border. We expect this to be an artefact of our reconstruction method. Points at the outside are more likely to be in shadow for one or the other light, which may have caused the darker colour taint. The dark circle becomes thinner for when more images are used, as can be seen in Figure 1 (right) for twenty-five images instead of five.
- **Question 1.2**
  We calculate the albedo and the unit normal vector by solving the linear system $\mathbf{i}(x, y) =$

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

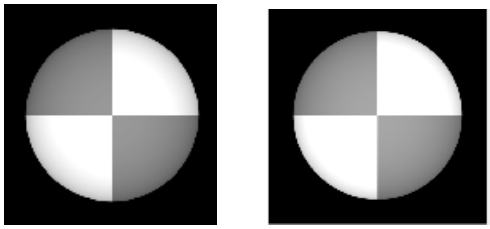

Figure 1: Albedo reconstructed for the spherical object from five (left) and twenty-five (right) images.

$V \cdot \mathbf{g}(x, y)$. Since the vector g has three dimensions, the mathematical minimal amount of images needed to solve this equation is also three. However, a point can be completely in shadow in one or more of these images. Therefore, in practice we need more images and the light sources should be positioned properly. This is especially true for objects with a complicated shape.

- **Question 1.3**
  Shadows make a point look darker than a point with the same albedo that is not in shadow. The variations in brightness occurring under different light sources make it possible to reconstruct the shape of the surface in photometric stereo.

  A possible problem occurs when a point is completely shadowed by one or more images. In this case the shadowed point does not expose any information about its albedo. A possible solution is provided by the 'shadow trick'. When four or more images are available, the normal vector is overdetermined and the images causing a shadowed pixel can be discarded. The mathematical trick to discard the zeros is to solve the equation $I\mathbf{i}(x, y) = IV \cdot \mathbf{g}(x, y)$ instead of $\mathbf{i}(x, y) = V \cdot \mathbf{g}(x, y)$.

  We compared the results of using the shadow trick with the results of not using the shadow trick for five and twenty-five images. The results for the normal image are in Figure 2. The results show that, in the case of five images, the calculated normal is more smoothed when using the shadow trick, which indicates a higher quality. For twenty-five images, however, we can hardly detect any differences in the results with and without the shadow trick. We think that this is because in the case of twenty-five images the least square solution is not really affected by a small number of shadowed points. We did not notice a significant difference in the albedos when comparing images with and without the shadow trick.

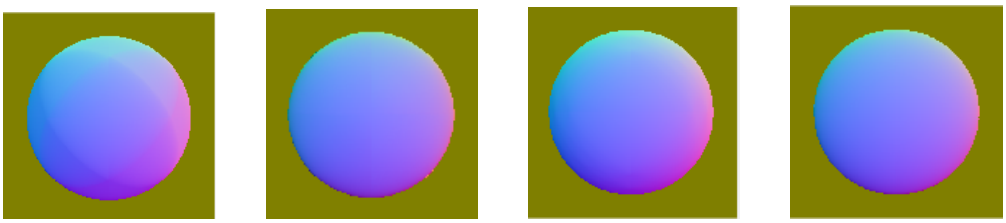

Figure 2: Normal map for (left to right): five images, five images with shadow trick, twenty-five images, twenty-five images with shadow trick.

1.2 Test of Integrability

- **Question 2.1**
  See implementation.
- **Question 2.2**
  We checked our computation by checking that the mixed second partial derivatives are approximately equal. We took 0.005 as a threshold and plotted the outliers. The results are shown in Figure 3.

  We see that all errors are at the border, which is explainable from the fact that the height of the sphere changes more rapidly at the borders than at the top, for changes in the x and y direction. Therefore, the calculated error is expected to be bigger on average. Another

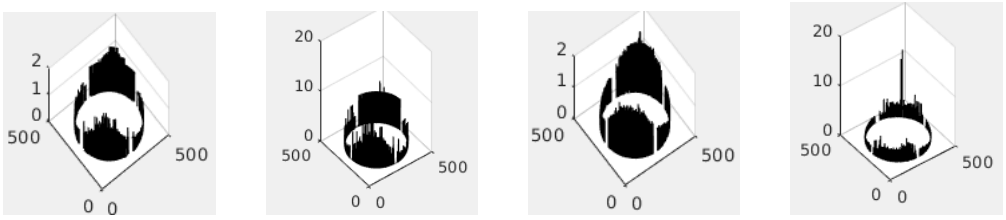

Figure 3: Outliers comparing mixed second derivatives for (left to right): five images, five images with shadow trick, twenty-five images, twenty-five images with shadow trick.

possible reason is that the calculations for points at the border are less accurate because these points are more likely to be in shadow for one or more images. The darker outside circle noticed in Q1.1 supports this explanation.

We compared the results for five and twenty-five images, with and without the shadow trick. The percentage of outliers are respectively: 0.013 (five with trick), 0.0068 (five without trick), 0.0094 (twenty-five with trick), 0.0075 (twenty-five without trick). These numbers and the pictures suggest that using the shadow trick results in more and more severe outliers. This seems unexpected at first sight, we can not really explain this phenomenon. We also see that the results for five images with the shadow trick are much worse compared to the results for twenty-five images with the shadow trick. We think that this is because in general using more images reduces the numbers and sizes of outliers.

1.3 Shape by Integration

- **Question 3.1**
  We reconstructed the surface heights for each pixel by numeric integration using the column first path and the row first path. The results for five images are in Figure 4 (left and centre). We compared both figures visually but we do not see a clear difference between them.

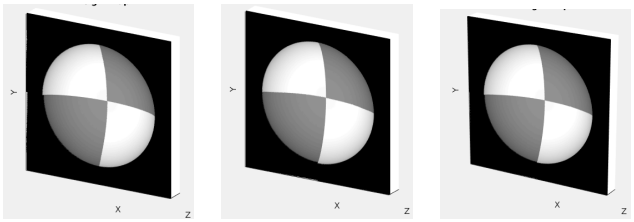

Figure 4: Surface reconstructed from five images using column-path (left), row-path (centre) and average of column- and row- path (right).

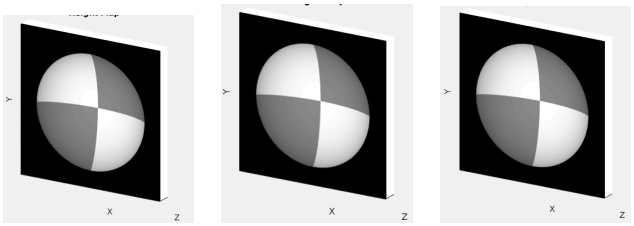

Figure 5: Surface reconstructed from twenty-five images using column-path (left), row-path (centre) and average of column- and row- path (right).

- **Question 3.2**
  We also did a reconstruction using the average of the column and row first paths, see Figure 4 (right). We expected the average path to improve the results of both the row and column path reconstruction. However, visual inspection did not reveal any clear difference between the three figures. We repeated the experiment with twenty-five images shown in Figure 5.

We expected that a higher number of images would lead to a smoother reconstructed surface. However, we could not visually confirm this expectation. A suggestion for further research is to compare the reconstruction results by looking at how well they respect the symmetric properties of the sphere.

1.4 Experiments with Different Objects

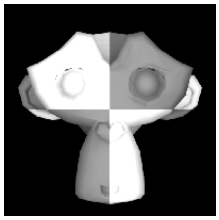

Figure 6: Albedo

Figure 7: Normal map

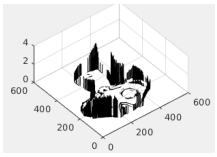

Figure 8: Integrability check

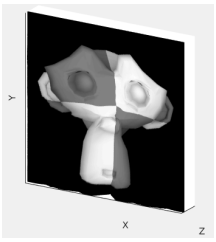

Figure 9: Reconstructed Surface

- **Question 4**
  We applied the photometric stereo algorithm to the synthetic monkey object described by 121 images. The results are illustrated in Figure 6 to 9.

  The albedo result for the monkey shows errors near the exterior border and near the borders of the eyes (Figure 6). These errors appear as a darker grey than the actual grey taint of these points. We expect these errors to occur at these points because they are more likely to be in shadow for different light sources. As a possible solution, we suggest adding more images using light sources that specifically illuminate these problematic points.
- **Question 5**

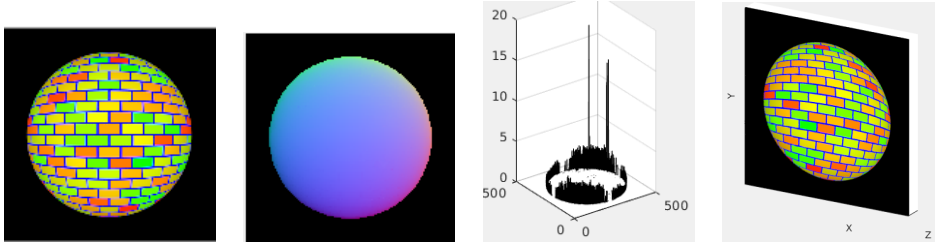

Figure 10: From left to right: Albedo, normal map, second order derivative outliers, and reconstructed surface for the coloured sphere.

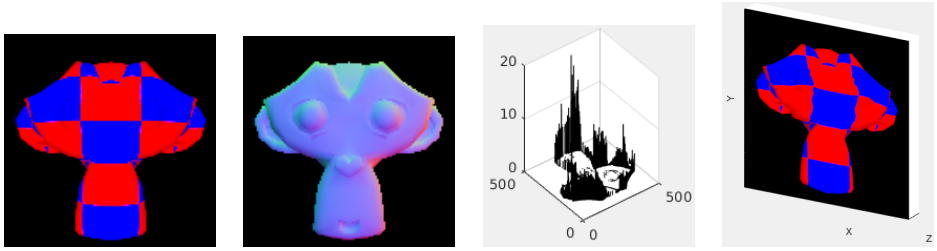

Figure 11: From left to right: Albedo, normal map, second order derivative outliers, and reconstructed surface for the coloured monkey.

We applied the photometric stereo algorithm to colour images. First, we split the image into separate channels and calculate their albedo and input normals individually. We reconstruct the albedo by simply combining the albedos calculated for the individual channels. We

reconstruct the surface normal by adding the normal vectors for the individual channels and then normalising the resulting vector.

We added a small fix to handle the situation that a pixel value in a channel is zero or NaN. In that case, we set the albedo and the surface normal to 0, resp. $\mathbf{0}$ in that channel. Since we construct the final surface normal from the three individual channels, the effect is that the channel with the zero value is ignored.

Figure 10 shows the results for applying the adapted photometric stereo algorithm to the coloured sphere, while Figure 11 shows the resulting images for the coloured monkey. The images illustrate that the adapted algorithm is able to reconstruct albedo and surface for coloured images.

- **Question 6**
  We applied the reconstruction algorithm to a subject consisting of 64 images of the Yale Face Database. Figure 12 shows the albedo, the normal map and the second order partial derivative errors. Figure 13 shows the reconstructed surfaces for the column- and row-integration paths.

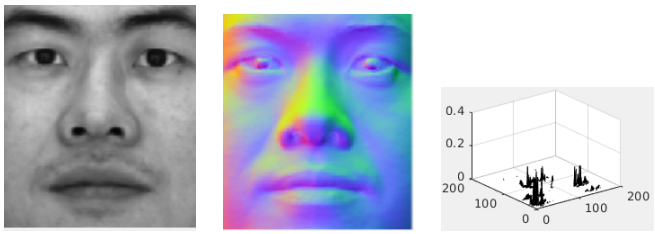

Figure 12: From left to right: Albedo, normal map and second order derivative outliers for the Yale face subject.

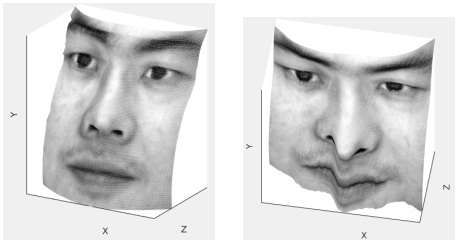

Figure 13: Reconstructed surface for the Yale face subject using the column- (left) and row- (right) integration path.

The reconstruction results show that the column path leads to a much better result than the row path, especially the mouth is much more realistic for the column path. A likely explanation is that the row vertical integration path over the nose gives problems because of the steep altitude differences at the lower end of the nose. The column path avoids integrating vertically over the nose and therefore results in a more smooth reconstructed image.

Shape from shading methods depends on the assumption that all images are in a fixed view under different light sources. The Yale Face image is likely to violate this assumption in a subtle way since it is hard to keep your face in exactly the seem position and expression for a longer period of time. However, we could not visually detect images in which the face had a position or an expression that was clearly deviating from the others.

What we did find however were two images with a different light source, that is, parallel striped light instead of a far distance point source (Figure 14). We repeated the experiment after removing these problematic images. We could not however visually detect a clear improvement in either the reconstructed surface or the albedo and normal map. We guess that removing 2 from the 64 images does not have a big impact on the final results.

Figure 14: Yale Face image with deviating illumination.

2 Colour Spaces

For this part of the assignment we implemented the colour space conversions discussed below. For the opponent colour space and the normalized RGB colour space we used the formulas known from the literature, for HSV and YCbCr we used the built-in matlab functions. For the grey scale conversions we followed the methods described in `https://www.johndcook.com/blog/2009/08/24/algorithms-convert-color-grayscale/`.

- **RGB Colour Model**
  An RGB colour model is used since the colours it uses are seen as additive colours, meaning, having equal parts of red green and blue will result in white. This is similar to colours on monitors. Since the monitor emits light, it makes sense to use additive colours that become lighter when there is more of it.
- **Colour Space Properties**

  – Opponent Colour Space
  The opponent colour space exists of three elements O_1, O_2 and O_3. O_1 is the luminescence of the image, O_2 is the red-green element, and O_3 is the blue-yellow element. This can be seen in figure 15. On the left is the full opponent colour space. To its right is the O_1 element. The third is the O_2 element of the colour space and O_3 is shown on the right. The opponent colour space is stated [2] to be useful for compressions.

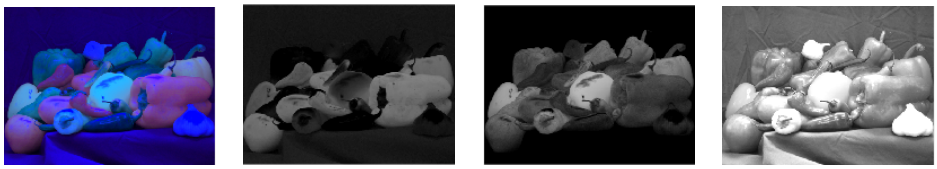

Figure 15: Colour Spaces: Opponent Colour Space

  – RGB Normalised Colour Space
  This colour space is used for the times that one does not want to have the colours of the image to be influenced by shadows, certainly helpful in some situations. It divides the pixel by the total spectrum of pixels. This can be seen in figure 16. The full RGB normalised can be viewed in the left image. The red, green and blue component can be viewed in the second, third and fourth image respectively.

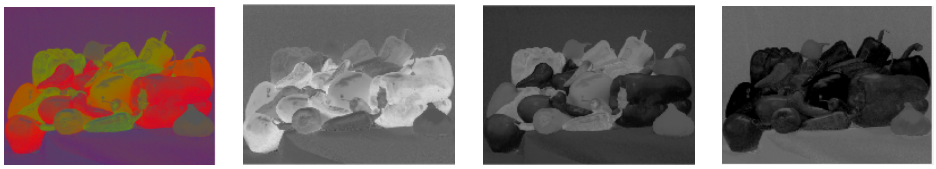

Figure 16: Colour Spaces: RGB normalised Colour Space

– HSV Colour Space
The HSV (Hue Saturation Value) Colour space exists of three parts. The hue, meaning which colour is used. The saturation, meaning how intense the colour is. The third, the value of brightness, states how black or how white the colour is. These can be seen in figure 17. The left is the full HSV image, to its right is the hue, third is the saturation and finally the value. This can be helpful when one would like to remove shadows.

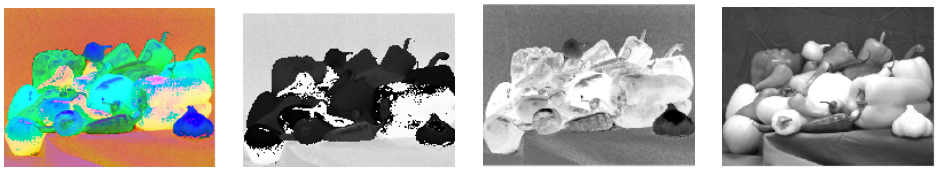

Figure 17: Colour Spaces: HSV Colour Space

– YCbCr Colour Space
In the YCbCr Colour space, The Y stands for the luminescence, Cb is the blue-difference and Cr is the red-difference. Both are chroma components, hence the C. This can be seen in figure 18. The left shows the full YCbCr image, to its right, the luminescence. Third shows the blue-difference and finally, the red-difference is shown. YCbCr uses the fact that humans do not perceive as much information in the right two images in figure 18 than the second one to store images more efficiently.

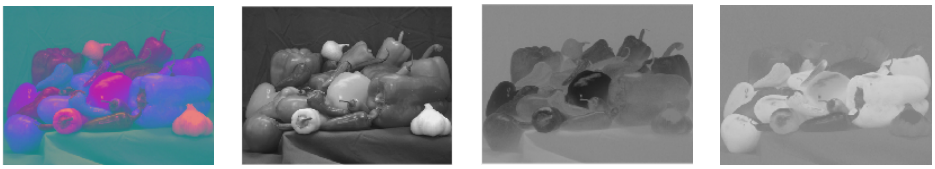

Figure 18: Colour Spaces: YCbCr Colour Space

– Grayscale
There are three algorithms to change an image into a grayscale one. The first is called the lightness method, where it averages the most and the least standing out colours. Second, the average method, which takes the average of the values for red green and blue. Third, the luminosity method. This is similar to the average method, however, the colours that are most striking to us are enhanced. These can be seen in figure 19. The left image is made by using the lightness method, the image to its right is made with the average method. The third demonstrates the luminosity method and the fourth used the built-in MATLAB function $rgb2gray$ which uses only the luminescence information of the image. This makes that image look like the one on its left.

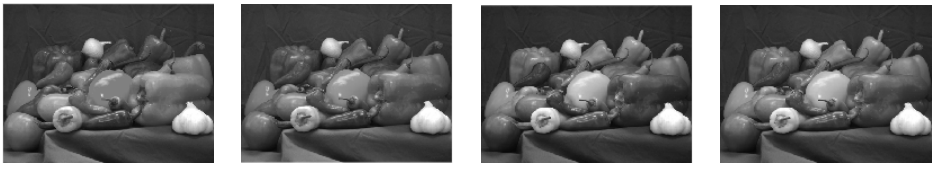

Figure 19: Colour Spaces: Grayscale

- **More on Colour Spaces**
Another colour space is called the CMYK Colour Space, where the abbreviation stands for the four colours used, cyan, magenta yellow, and black. This colour space is used in printing, so it is made to accommodate the white paper it is printed on.

3 Intrinsic Image Decomposition

An image can be approximately reconstructed from it's reflectance and shading by taking the element-wise product. In this part of the assignment we reconstruct an image from it's intrinsic components and use this idea to recolor the image by modifying its reflectance component.

- **Other Intrinsic Components**
  Other components that an image can be decomposed to are specular reflection, depth [3] and structure-texture separation as mentioned in [4]. According to [4], the texture cannot be fully captured by other intrinsic components, like shadow and reflection. This is why the structure-texture separation component has been introduced, making it possible to also capture that important part of the image.
- **Synthetic Images**
  The difference between synthetic images and real-world images is that with the synthetic images all influences, like light sources, can be fully controlled. This is not the case with real-world images. To make images like the synthetic ones in the real world, a lot of time and money need to be spent to construct a database of these.
- **Image Formation**

  We implemented image reconstruction by taking the element-wise product of reflectance and shading. Figure 20 shows the original image, the reflectance and shading components and the reconstructed image.

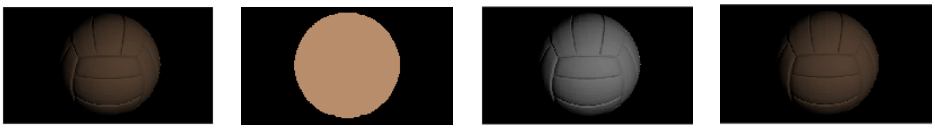

Figure 20: Deconstruction and re-composition of a ball. From left to right: original image, reflectance, shading, and reconstructed image.

- **Recolouring**
  The true material uniform color of the ball is (184, 141, 108), in RGB values describing the reflectance. We recoloured the ball by changing these RGB values. Figure 21 shows the result of recolouring the ball in pure green and magenta. Since this is just one layer, the shadows that will be added to the pure colour make it not pure anymore.

Figure 21: The original ball and the reconstructed ball in green and magenta.

4 Colour Constancy

The grey-world algorithm is used to correct the colors in an image so that the resulting image looks as if it is taken under a white light source. We implemented the grey-world algorithm by changing the RGB channels so that all channel values are multiplied by the factor $128/mean(channel)$, thereby assuming that the average colour in a scene is grey (128, 128, 128).

- **Question 4.1**
  Figure 22 illustrates the effect of automatic white balancing by showing an example image before and after applying the grey-world algorithm. We see that the reddish colour cast on the image is removed and that the colours look more natural.
- **Question 4.2**
  The grey-world algorithm depends on the assumption that under a white light source, the

Figure 22: Example of an image before- (left) and after- (right) automatic white balancing. The image at the right looks more natural since the reddish colour cast is removed.

Figure 23: Example of an image before- (left) and after- (right) automatic white balancing. The result of white balancing looks unnatural since the grey world assumption is violated by the input image with a lot of green objects.

average colour in a scene is grey (128, 128, 128). Therefore, the algorithm will not work well for scenes that violate this assumption. To illustrate this point we apply the algorithm to a nature image in which the green colour is dominant (Figure 23). Indeed the output image looks highly unnatural since the reduced green colour was not caused by a light source but by the scene itself.

- **Question 4.3**
  As described in [1] there are many other colour constancy algorithms. These can be divided into two types, pre-calibrated approaches and data-driven approaches. The grey-world algorithms fall into the second category. Another example of data-driven approaches for colour constancy are machine learning approaches. These are helpful since these can adapt to dynamic illumination conditions, however, it takes long to train. Another approach is called the retinex approach, which is helpful when images are dark.

5 Conclusion

We implemented the photometric stereo algorithm and applied it to synthetic objects (sphere and monkey) and a real object (Yale face). The results were positive, that is we were able to reconstruct the 3D surface and to recover the albedo.

Next, we focused on colours. First, we showed different colour spaces for an image and discussed their merits. We then decomposed an image into its formation components, namely reflectance (albedo) and shading (illumination). Using these components we were able to reconstruct the image and to reconstruct the image with a different albedo. Finally, we applied the grey-world algorithm to correct an image so that it appears to be taken under a white light source. The result is an image that looks more natural from a colour perspective.

References

References

[1] Vivek Agarwal, Besma R Abidi, Andreas Koschan, and Mongi A Abidi. An Overview of Color Constancy Algorithms. *Journal of Pattern Recognition Research*, 1:42–54, 2006.

[2] Silvio Borer and S Süsstrunk. Opponent color space motivated by retinal processing. *First European Conference on Color in Graphics, Imaging and Vision (CGIV)*, 1:187–189, 2002.

[3] Qifeng Chen and Vladlen Koltun. A simple model for intrinsic image decomposition with depth cues. *Proceedings of the IEEE International Conference on Computer Vision*, pages 241–248, 2013.

[4] Junho Jeon, Sunghyun Cho, Xin Tong, and Seungyong Lee. Intrinsic image decomposition using structure-texture separation and surface normals. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8695 LNCS(PART 7):218–233, 2014.