Assignment 1

Dana Kianfar - Jose Gallego Posada

11391014 - 11390689

{dana.kianfar, jose.gallegoposada}@student.uva.nl

1 Photometric Stereo

Description

Photometric stereo is a technique withing a broader class of procedures called Shape from Shading in which we are interested in recovering the shape of an object of which multiple pictures have been obtained. In the case of photometric stereo, several pictures of the object are taken under different lighting conditions.

As a result of this method we expect to recover the 3 dimensional representation of the object as well as its albedo at every point. This final outcome allows to circumvent the restrictive assumption that the reflectance of the object is the same everywhere employed in traditional Shape from Shading.

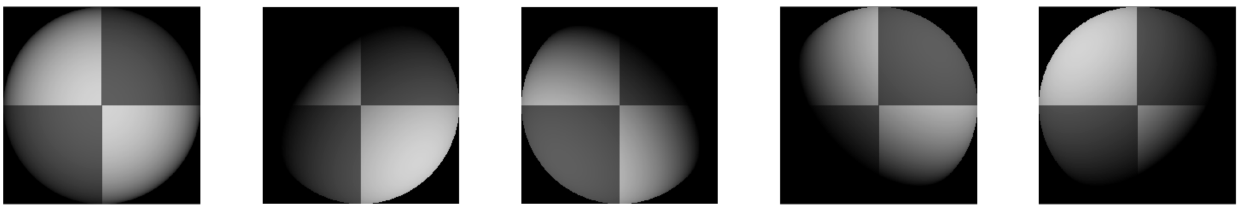

Figure 1: Pictures of a sphere under various lighting conditions.

For this exercise we considered five pictures of a sphere taken with a fixed orthographic camera and several light source locations. For the sake of simplicity, we assumed that the light sources in the sample images are far way and, respectively, frontal, right-below, left-below, right-above, left-above, with equality at all pixels, as shown in Figure 1.

We implemented algorithm 5.1 described in [1], which can be summarized by the following equations. Assume that the response of the camera is linear in the surface radiosity, so the value of a pixel at (x, y) is:

$$I(x, y) = \mathbf{g}(x, y) \cdot \mathbf{V} = \mathbf{g}(x, y) \cdot k\mathbf{S},$$

where $\mathbf{g}(x, y) = \rho(x, y)\mathbf{N}(x, y)$ describes the surface, ρ is the albedo, $\mathbf{N}$ is the normal vector, $\mathbf{S}$ is the light source vector, and k is the constant connecting the camera response to the input radiance.

Let $\{I_i\}_{i=1}^n$ be a set of $n = 5$ images. We can define the matrix $\mathscr{V}$ as:

$$\mathscr{V} = \begin{pmatrix} \mathbf{V}_1^T \\ \vdots \\ \mathbf{V}_n^T \end{pmatrix}$$

In the current experiment, for example, the second row of the matrix $\mathscr{V}$ corresponds to the vector to the source vector $\frac{1}{\sqrt{3}}[1, -1, 1]^T$, since the second image in Figure 1 has the light source in position right-below.

Then, for each point, we can gather the measurements from the different images as:

$$\mathbf{i}(x, y) = \begin{pmatrix} I_1(x, y) \\ \vdots \\ I_n(x, y) \end{pmatrix} \qquad \mathscr{I}(x, y) = \begin{pmatrix} I_1(x, y) & \dots & 0 \\ 0 & \ddots & 0 \\ 0 & \dots & I_n(x, y) \end{pmatrix},$$

and build the linear system

$$\mathscr{I}\mathbf{i}(x, y) = \mathscr{I}\mathscr{V}\mathbf{g}(x, y).$$

A least squares solution to the previous linear system yields the $\mathbf{g}$ vector at the point (x, y). Therefore, we can compute the albedo as $\rho(x, y) = |\mathbf{g}(x, y)|$ and the normal vector as $\mathbf{N}(x, y) = \frac{1}{\rho(x,y)}\mathbf{g}(x, y)$.

Finally, we would like to express the surface of the object as $(x, y, f(x, y))$. So, the normal vector found before coincides with:

$$\mathbf{N}(x, y) = \frac{1}{\sqrt{1 + \frac{\partial f}{\partial x}^2 + \frac{\partial f}{\partial y}^2}} \begin{pmatrix} -\frac{\partial f}{\partial x} \\ -\frac{\partial f}{\partial y} \\ 1 \end{pmatrix}$$

Additionally, imposing the condition that the mixed partial derivatives of the function f must coincide, we ensure that the recovered surface is smooth. This verification is called the integrability test, as we will reconstruct the surface by:

$$f(x, y) = \oint_C \left[\frac{\partial f}{\partial x} \frac{\partial f}{\partial y}\right] \cdot d\mathbf{l} + c$$

Note that the integrability condition ensures that we have a conservative vector field. Thus, the reconstructed surface is independent of the path C selected.

Results

After completing the implementation in MATLAB, we executed an experiment to determine an adequate value for the parameter k. For this, we performed the photometric stereo using different values of k and chose the one for which a higher contrast in the colors of the quadrants of the sphere was observed, which is $k = 200$. This parameter was used in the subsequent experiments in this section. The results of this search are displayed in Figure 2.

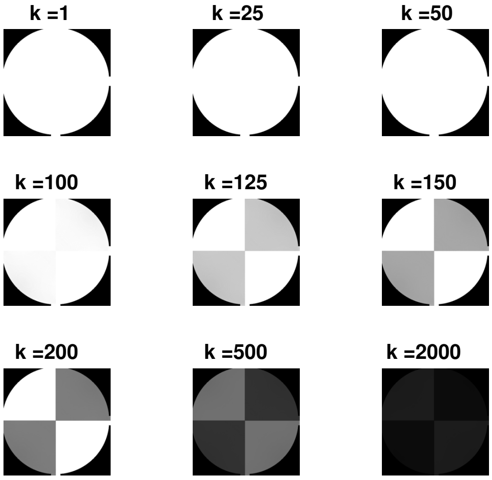

Figure 2: Albedos for different values of the scaling parameter k.

Figure 3 shows the results for the integrability test. For an improved readability we display only those points for which $\left(\frac{\partial p}{\partial y} - \frac{\partial q}{\partial x}\right)^2 > 0$. The points clearly correspond to the boundary of the sphere (as from the viewpoint of the camera), in which the gradient is ill-defined, for which differences in the mixed partial derivatives are encountered. However, note that for all the points in the circle defined by that circumference, the integrability est is successful.

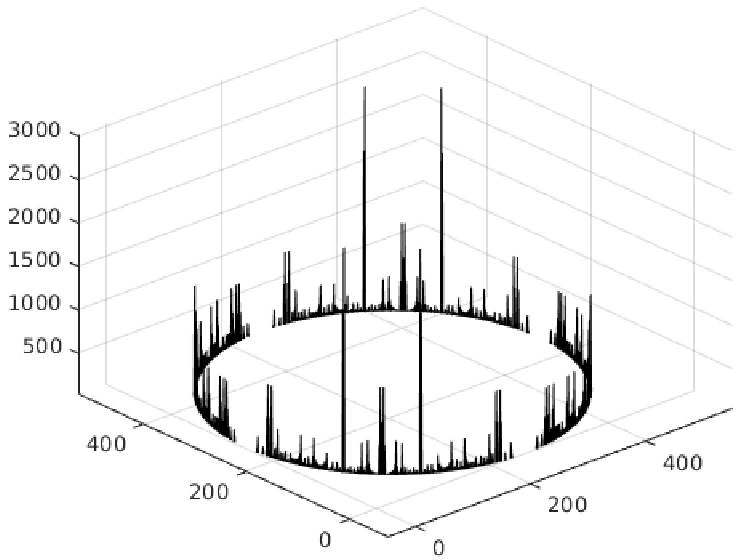

Figure 3: Squared difference of second derivatives at each pixel for $\left(\frac{\partial p}{\partial y} - \frac{\partial q}{\partial x}\right)^2 > 0$.

Finally, the reconstructed surface and the corresponding normal vector field are presented in Figures 4 and 5. The spherical shape obtained is consistent with the expected outcome of the experiment.

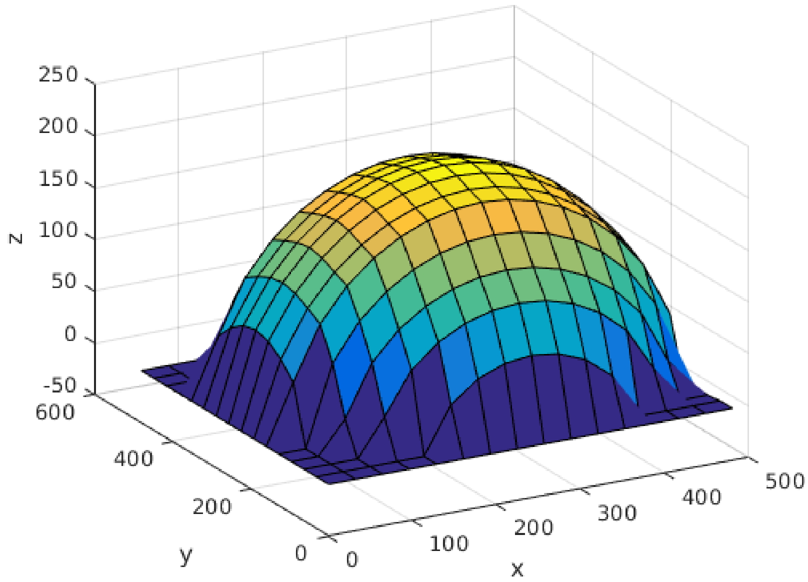

Figure 4: Reconstructed surface.

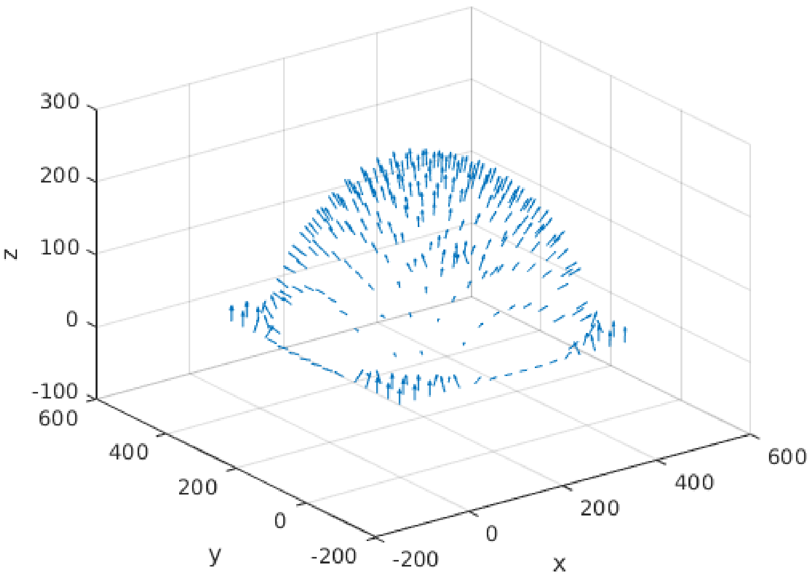

Figure 5: Normal vectors for the reconstructed surface.

2 Color Spaces

Color spaces are schemes by which we can quantitatively describe and create colors. Different color spaces are better for different applications and devices. In this section, we explore various color space transformation of an RGB image which is shown in Figure 6. We selected this image due to its qualitative colorfulness and its ability to highlight differences between the color spaces.

Each transformation from RGB to another color space is visualized. Specifically, after each conversion we display the image for each of the three channels separately. Additionally, for comparison to RGB we present the color distribution of each space. The RGB color distribution for the image is shown in Figure 7.

Figure 6: Original image in RGB.

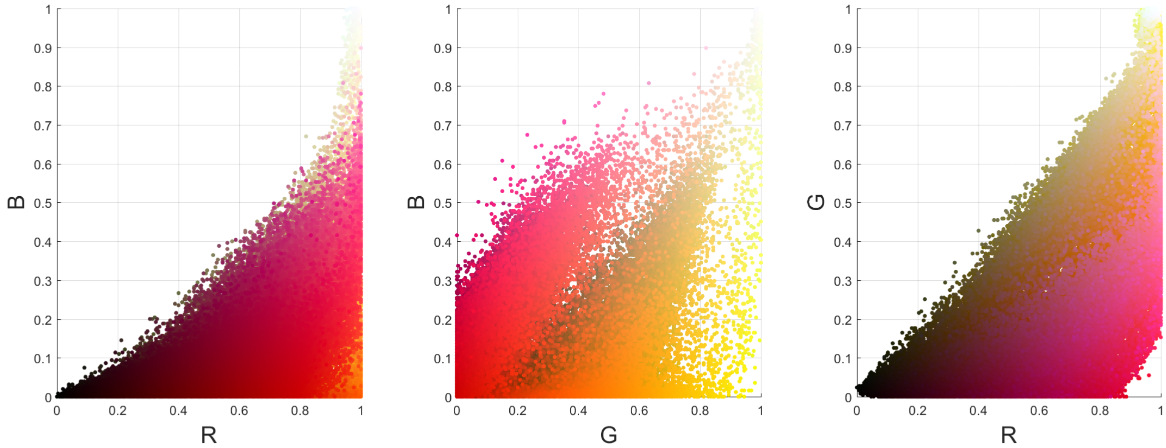

Figure 7: Color distribution of image in RGB space.

We also perform a number of grayscale conversions. For each conversions, we present the empirical distribution of pixel intensities. In all experiments, we cast the original RGB values to floating points for numerical stability.

2.1 Opponent Color Space

For each RGB triple in the image, we performed the transformation to the opponent color space and obtained three channels O_1, O_2, O_3 as follows.

$$\begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix} = \begin{pmatrix} \frac{R-G}{\sqrt{2}} \\ \frac{R-G-2B}{\sqrt{6}} \\ \frac{B}{\sqrt{3}} \end{pmatrix}$$

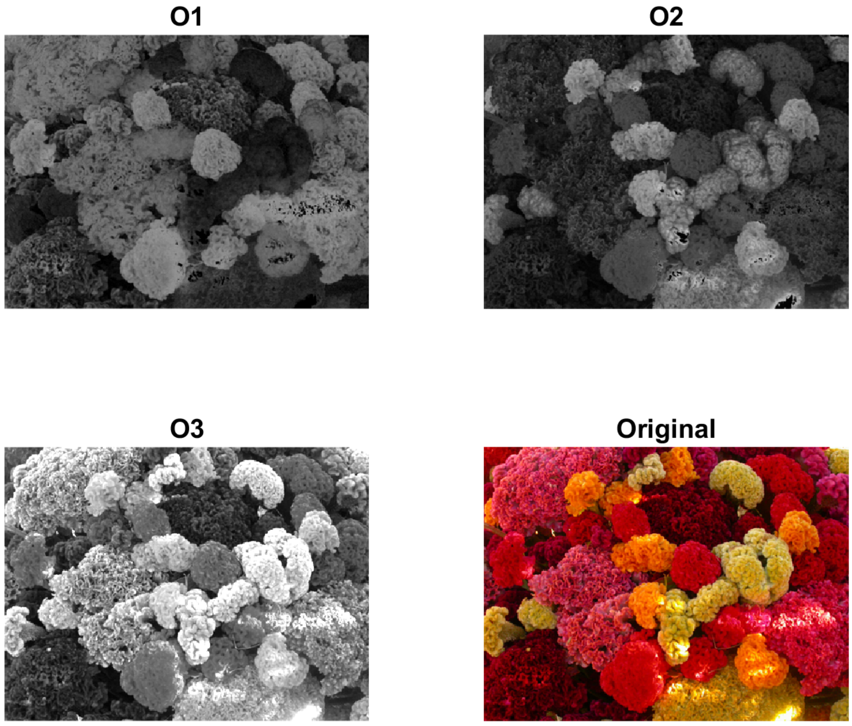

Figure 8: Opponent Color Space.

Figure 8 shows the channels of the image in the opponent color space. We present the projection of the color distribution on each channel pair in Figure 9. Qualitatively, while the shape of the projected distributions vary, we observe that adjacent colors are not significantly different than in RGB.

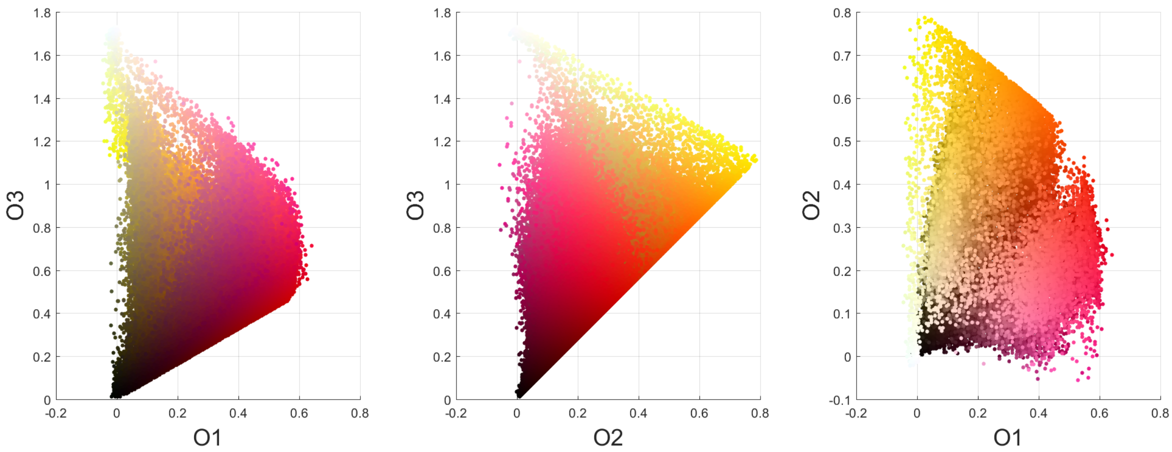

Figure 9: Opponent Color Space color distribution.

2.2 Normalized RGB Color Space

For each pixel in the RGB images, we calculated the transformation to the normalized RGB color space and obtain the three channels R_n, G_n, B_n as follows:

$$\begin{pmatrix} R_n \\ G_n \\ B_n \end{pmatrix} = \begin{pmatrix} \frac{R}{R+G+B} \\ \frac{G}{R+G+B} \\ \frac{B}{R+G+B} \end{pmatrix}$$

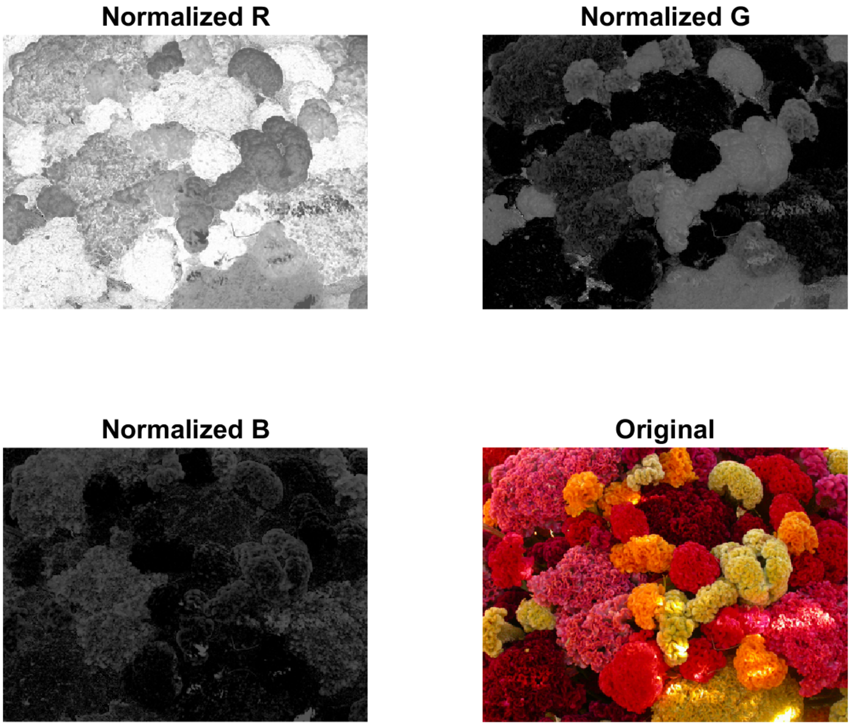

Figure 10: Normalized RGB Color Space.

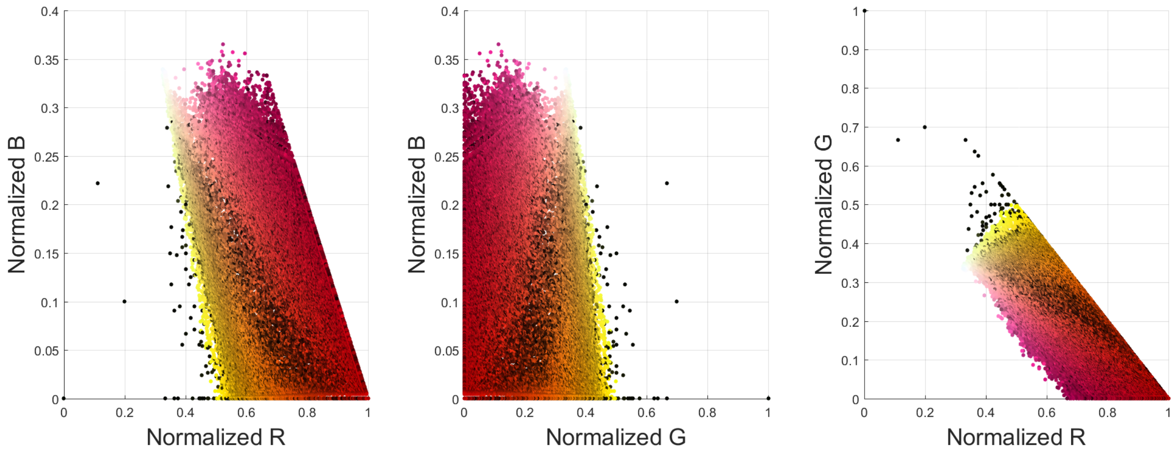

Figure 11: Normalized RGB Space color distribution.

Figure 10 shows the different channels of the image in the opponent color space. The Normalized R channel appears brighter as the majority of pixels in the RGB image have a red-like hue. We present the projection of the color distribution on each channel pair in Figure 11. Qualitatively, we observe that adjacent colors are different than in RGB.

2.3 HSV Color Space

The HSV color space separates luminance from the image and is considered to represent colors with closer resemblance to how humans perceive color. We use MATLAB's `rgb2hsv` function and obtain results shown in Figure 12. Qualitatively we observe a large different between how colors are clustered, shown in Figure 13, than RGB.

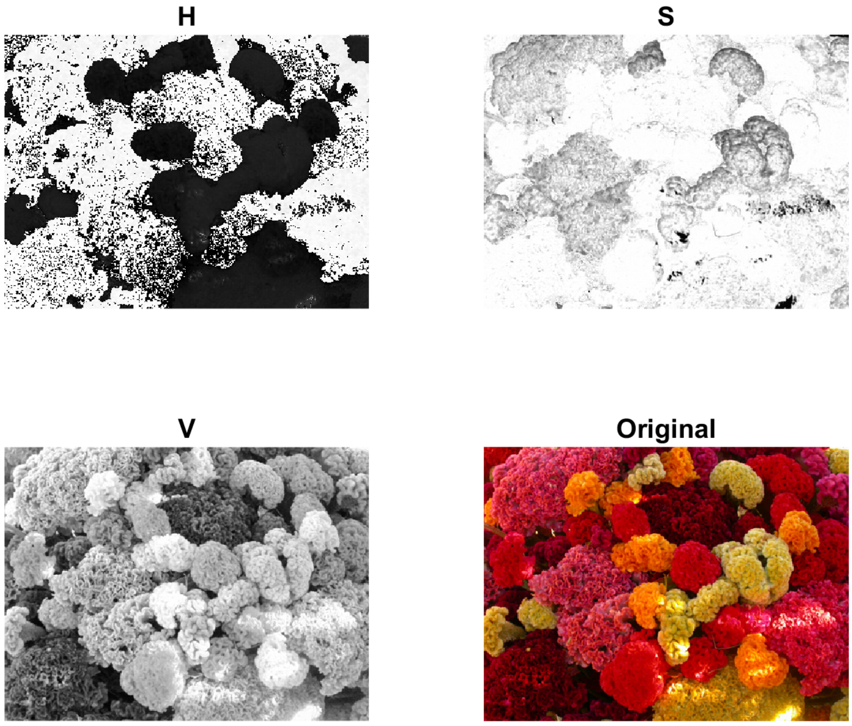

Figure 12: HSV Color Space.

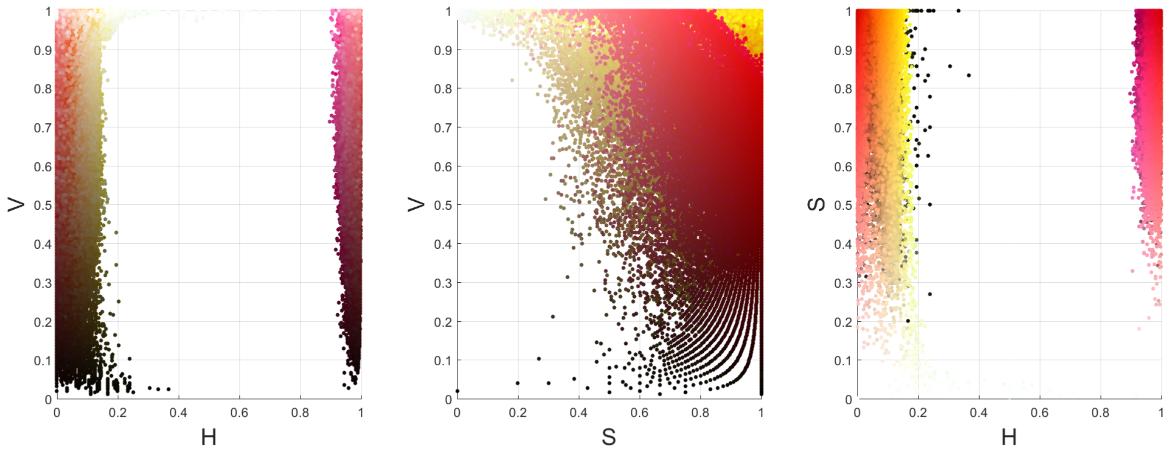

Figure 13: HSV Color Space color distribution.

2.4 YCbCr Color Space

We use MATLAB's `rgb2ycbcr` function and obtain results shown in Figure 14. Qualitatively we do not observe a significant different between how colors are clustered, shown in Figure 15, than RGB.

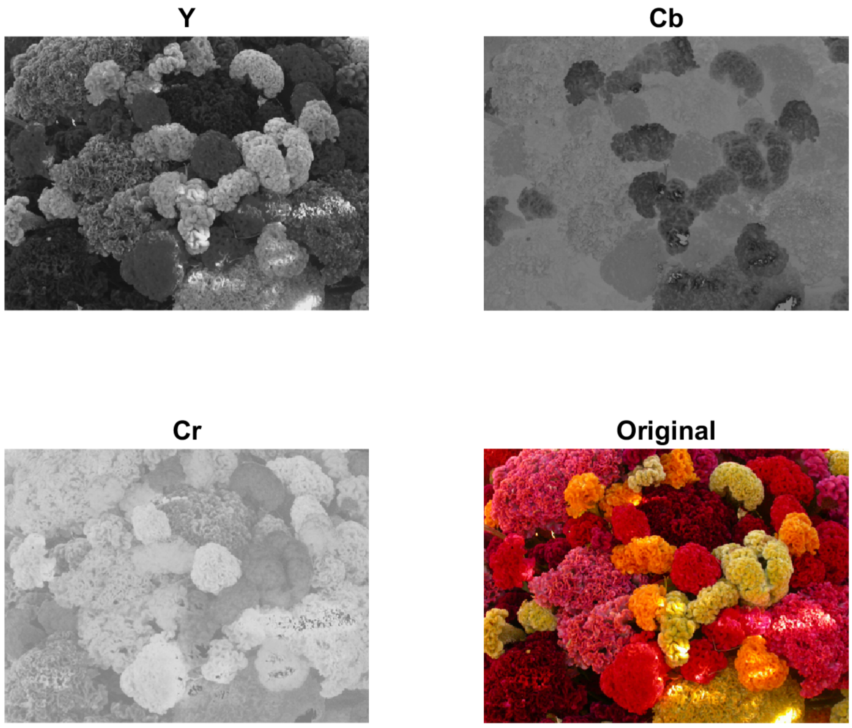

Figure 14: YCbCr Color Space.

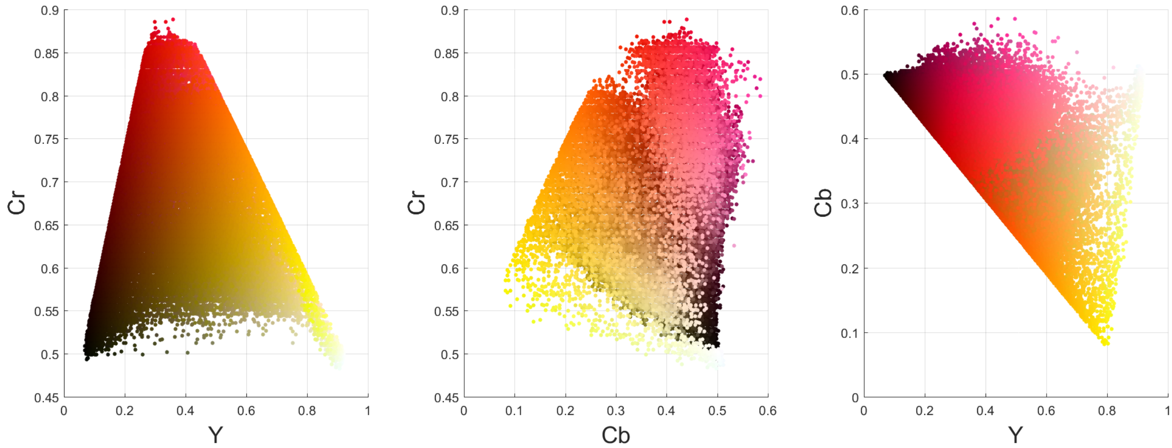

Figure 15: YCbCr Space color distribution.

2.5 Grayscale

We use three different techniques for converting the image to grayscale.

2.5.1 Lightness

The lightness method averages the most and least prominent colors to obtain the intensity i for each pixel as follows:

$$i = \frac{\max(R, G, B) + \min(R, G, B))}{2}$$

2.5.2 Average

This method averages the RGB channels to obtain intensity i for each pixel as follows:

$$i = \frac{R + G + B}{3}$$

2.5.3 Luminosity

The luminosity method performs a weighted averages on the RGB channels to obtain the intensity i for each pixel. The weights reflect the sensitivity of the human eye with respect to the RGB channels, with green and blue respectively weighed as the most and least prominent channels.

$$i = 0.21R + 0.72G + 0.07B$$

2.5.4 MATLAB

MATLAB's `rgb2gray` function forms a weighted sum of the R, G, and B components:

$$i = 0.2989R + 0.5870G + 0.1140B$$

2.5.5 Results

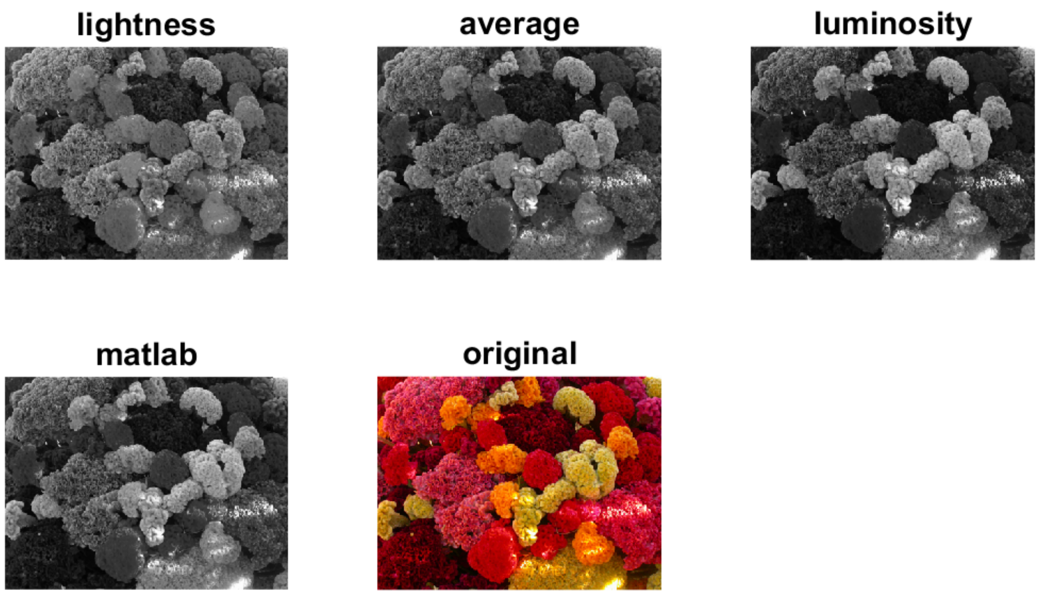

Figure 16: Grayscale Color Space.

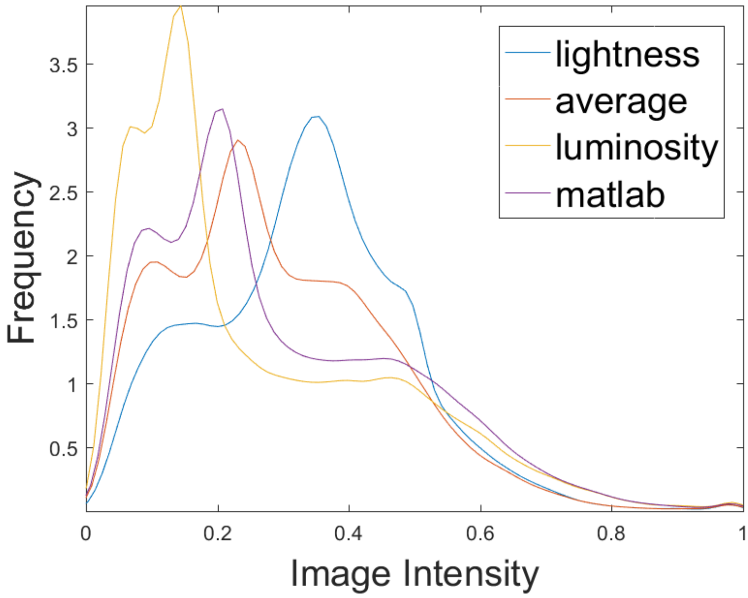

Figure 17: Grayscale Intensity Emprical Distribution.

Figure 17 presents the intensity distributions of the four different conversions shown in Figure 16. We observe that the Lightness method on average has higher intensity, with a peak around 0.4, at the expense of low contrast. The Luminosity method has a right-skewed distribution, with a peak around 0.18, and qualitatively is more representative of the contrast of the original image. The average method and MATLAB's default function perform very similarly and their differences are not noticeable to the naked eye.

References

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