Computer Vision 1 Homework 1

Dmitrii Krasheninnikov
University of Amsterdam
11719230
dmkr0001@gmail.com

Akash Raj Komarlu Narendra Gupta
University of Amsterdam
11617586
akashrajkn@gmail.com

1 Photometric Stereo

1.1 Estimating Albedo and Surface Normal

After estimating the albedo and the surface map, we were surprised to see the albedo slightly shadowed (visible on white parts of the ball), since we expected the ball to be uniformly black and white. This effect can be diminished by using the shadow trick, described further in this section.

A minimum of 3 images are needed to estimate albedo and surface normal [1]. However, due to shadows, more images may be needed if not all parts of the object are illuminated in at least two images. For example, Figure 1 shows the albedo reconstruction we would get (the rightmost image) if only the 3 images on the left are used for albedo estimation. The results of running the algorithm with more images are discussed in the next section (Test of Integrability).

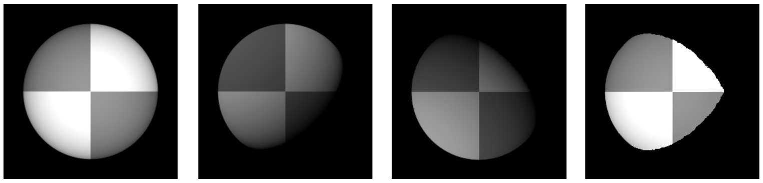

Figure 1: Albedo reconstruction (right) from the 3 images on the left.

To deal with shadows in the case of no ambient illumination [2, Sec. 5.4] introduce the following trick: both sides of equation

$$\boldsymbol{i}(x, y) = \mathcal{V}\boldsymbol{g}(x, y)$$

are multiplied by a diagonal matrix $\mathcal{I}$ formed from the from the image vector:

$$\mathcal{I}\boldsymbol{i}(x, y) = \mathcal{I}\mathcal{V}\boldsymbol{g}(x, y).$$

This trick removes the contributions from shadowed regions, because the elements of $\mathcal{I}$ corresponding to the points that are in the shadow are equal to zero.

Our intuition was that the shadow trick would be more useful when there is less images. This intuition is confirmed by our results – the results obtained from 5 images are very slightly lower quality (the albedo itself appears slightly shaded) without the shadow trick, while the results for 25 images are indistinguishable regardless of the shadow trick.

1.2 Test of Integrability

As shown in Figure 2, the outliers only appear on the boundary of the sphere, where the first order derivatives go through a sharp change and thus the second order derivatives are undefined.

After running the algorithm with 5 and 25 images, we observe that the number of outliers decreases and the reconstructed image becomes sharper as the number of images is increased. Interestingly,

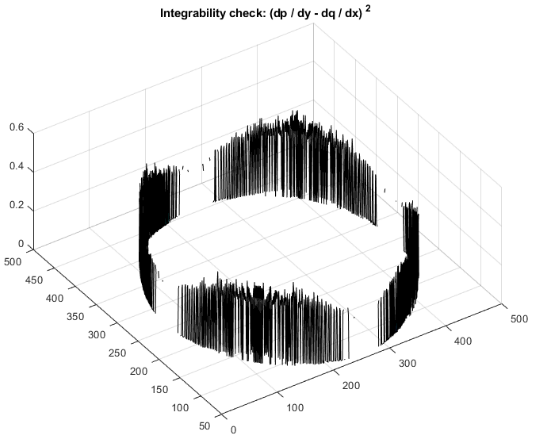

Figure 2: The outliers only appear on the boundary of the sphere, while the area corresponding to the surface of the sphere does not have any outliers.

the shadow trick adds outliers: we get 1865 outliers for 5 images without the shadow trick and 3430 outliers for 5 images with shadow trick; 1788 outliers for 25 images without the shadow trick and 2704 outliers for 25 images with shadow trick.

1.3 Shape by Integration

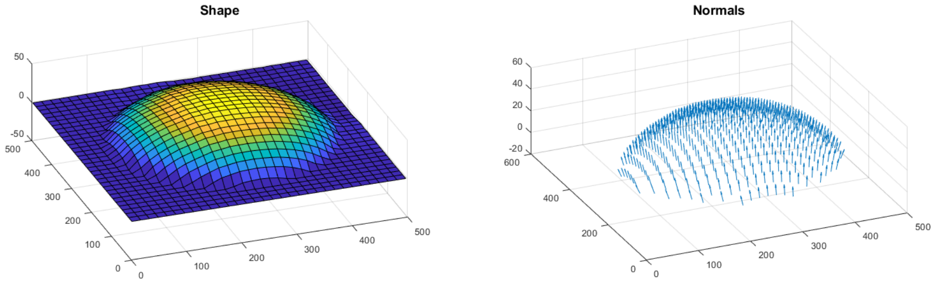

Figure 3: Shape of the sphere recovered by integration, and the surface normals. (25 images used; integration: average of column and row paths.)

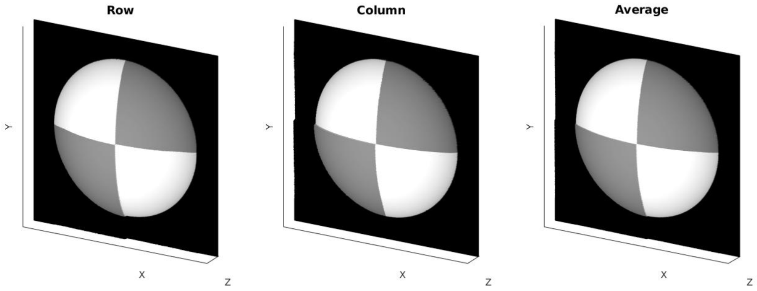

Figure 4: Surface reconstruction of the Sphere (5 images used) using row-major, column-major and average path integration.

In figure 4, we note that the errors in the height reconstruction occur when edges are perpendicular to the row (for row-major) and to the column (for column-major). This error is reduced when we take the average of the two methods. Our intuition is that averaging over multiple paths will reduce the errors in the derivative estimates.

The error in reconstruction depends on the symmetry of the original image. For the MonkeyColor dataset, we observe in Figure 8 the surface height reconstructed using the average method is worse than the column-major method. The row-major path reconstruction performs the worst of the three and its error is propagated to the image reconstructed using the average method.

For the SphereGray dataset, when 25 images were used to estimate the albedo and normal, the surface height map constructed is more accurate compared to the case where only 5 images were used. We observe that the reconstructed image has smoother edges.

1.4 Experiments with different objects

The albedo result on the MonkeyGray dataset contains more outliers than in the case of the SphereGray. This is likely due to the existence of more complex edges in the MonkeyGray dataset. In some of the images, the shadow of a part of the face (for example, the eyes) overlaps with the shadow cast by a different part of the face. In Figure 5, the right ear is completely not visible due to the shadow. A larger number of images in the dataset would reduce the albedo errors. In case of small dataset, we believe that removing such images might improve the surface height reconstruction.

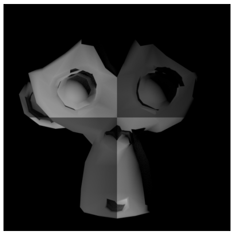

Figure 5: The shadow cast by the face completely covers the right ear.

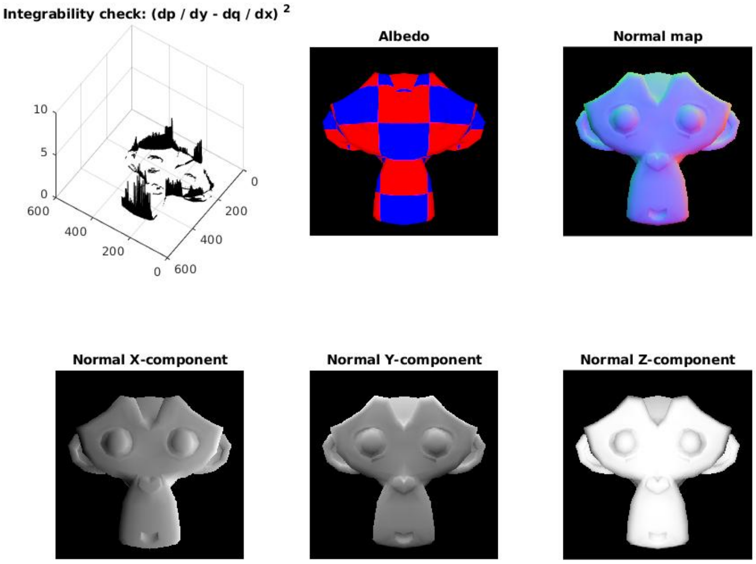

Figure 6: Integrability, albedo, normals for the MonkeyColor dataset

Figure 7 shows the results for SphereColor dataset. To work with 3-channel RGB images, three 1-channel albedos are constructed (one for each channel). For the surface normals, the maximum normal of RGB channels is considered. Generally since there are 3 zero-pixel values, it is not clear which one should be used for the computation of the normals. The maximum normal trick helps to get around this issue.

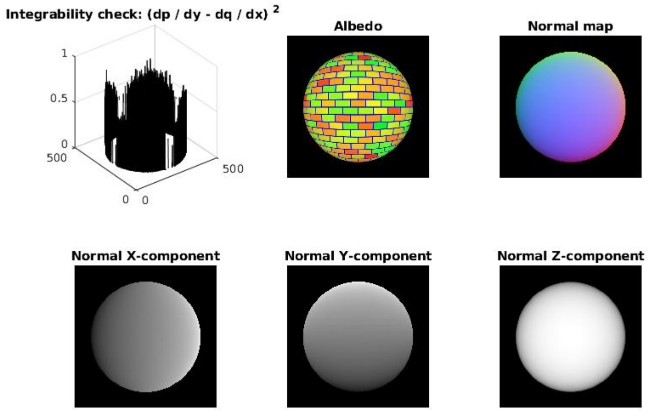

Figure 7: Integrability, albedo, normals for the SphereColor dataset

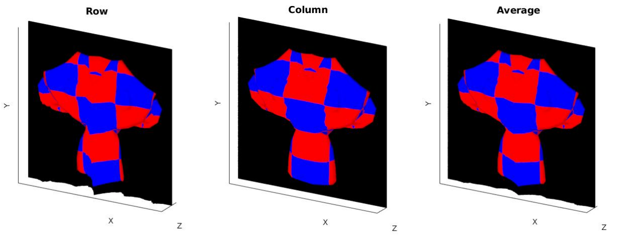

Figure 8: Reconstructed image (MonkeyColor) using row-major, column-major and average path integration.

1.4.1 Experiments on the Yale Face dataset

We ran our algorithm on the Yale Face Dataset. Figure 9 shows the results (albedo and normals). In the image decomposition process, we assume that the object is fixed and only the light source moves around. In reality, this might not always be true. When photographing a human face, it is probable that there are slight changes in the expression, eye direction, etc. Figure 10 shows two images where the assumptions are violated. In the right image, the eye is completely dark.

Figure 12 shows the height maps generated before and after the removal of the problematic images. The left image shows distortions (visible in the form of dark lines) along x-axis. These distortions do not appear in the right image. Figure 11 shows the height map constructed without shadow trick after the removal of the problematic images.

2 Color Spaces

2.1 RGB Color Model

The RGB color model is used as a basis of digital photography due to:

1. its additive nature: the colors are *added* to black (as opposed to subtracted from white), which is natural for recording light. This is also the reason why most mammals have RGB cones in their eyes.
2. historical reasons: the first color photographs appeared after James Clerk Maxwell's 1860 discovery that color photographs could be created using B&W film and R, G, and B filters.

The standard digital camera captures the full RGB color image via a CMOS sensor which has a subpixel-like structure for capturing R, G and B channels. Each "subpixel" consists of a filter that

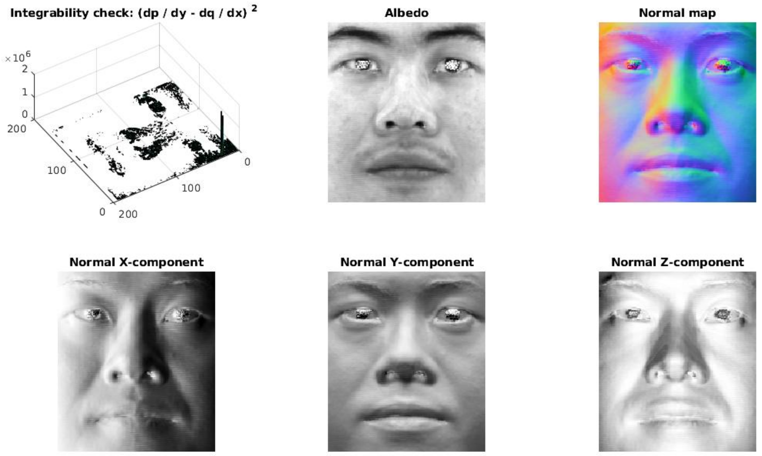

Figure 9: Albedo and Normals for the Yale face dataset

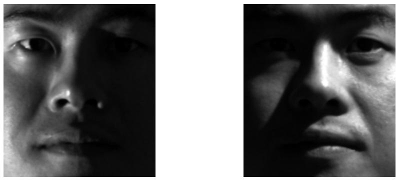

Figure 10: Two problematic images in the Yale Face dataset.

lets only one color through, and a light intensity sensor under the filter. Figure 13 shows a schema of such CMOS sensor.

2.2 Color Space Conversion and Color Space Properties

We implemented functions that convert an image from RGB color space to either of the Opponent, Normalized RGB (rgb), HSV, YCbCr and 3 variants of Grayscale color spaces. Each of the aforementioned color spaces is described below.

2.2.1 Normalized RGB color space

Each pixel of an image in the normalized RGB color space is that pixel's *chromaticity* vector. Chromaticity is used in many computer vision algorithms such as algorithms for white balance correction.

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

2.2.2 HSV color space

The HSV color space represents colors as (**H**ue, **S**aturation, **V**alue). The most common use of HSV is color selection tools, since HSV color representation feels more natural than RGB. However, there are other uses for HSV. For example, it is useful to have a separate intensity component for procedures such as histogram equalization of a color image; thus, such procedures are usually done on HSV images.

2.2.3 Opponent color space

The Opponent color space represents color in 3 components: the red-green channel O_1, the blue-yellow channel O_2 and the luminance channel O_3. Opponent color spaces natural axes facilitate

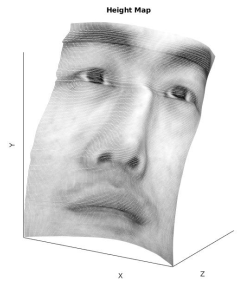

Figure 11: Height maps after removing the two problematic images (without shadow trick)

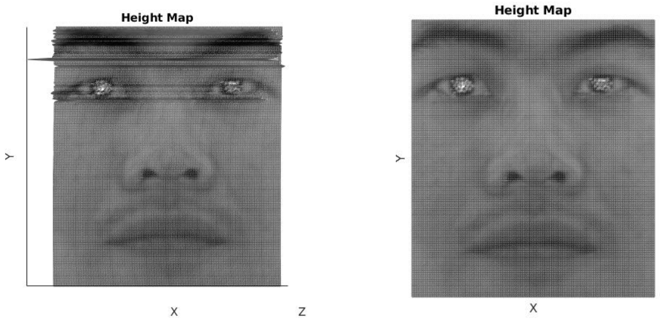

Figure 12: Height maps before and after removing the two problematic images (with shadow trick)

HSV-style color selection, and it is also designed to work well for computational applications such as color transfer, where HSV and RGB falter [3].

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

2.2.4 YCbCr color space

YCbCr is a color space that encodes a color as the luminance Y and two chroma channels C_b and C_r. YCbCr was designed with image storage and transmission goals in mind, and gets rid of some of the redundancy of the RGB color space. An example application of YCbCr is chroma subsampling: decreasing the resolution allocated to ”color” compared to ”light and dark” in the image. YCbCr is more naturally suited for this since it has separate channels for chroma and luminance.

2.2.5 Grayscale

Different methods can be used to convert an image to grayscale. Figure 18 shows the results of various methods used to generate the grayscale image.

Lightness: this method averages the most and the least prominent colors to get the intensity i for a pixel. This method tends to reduce contrast.

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

Average: for each pixel, the intensity is the average of R, G and B channels.

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

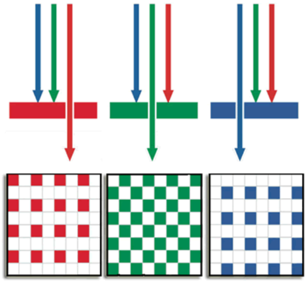

Figure 13: Schema of a CMOS sensor used in a standard digital camera. Image source: http://www.shortcourses.com/guide/guide1-3.html

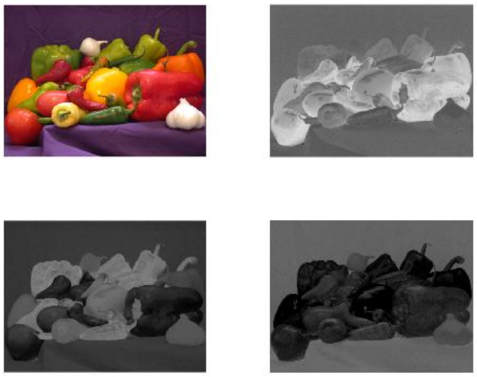

Figure 14: The three chroma components of the image in the nRGB color space.

Luminosity: in this method, a weighted average of the R, G and B channels is used to obtain the intensity of each pixel. The weights account for human perception; green is weighed more heavily compared to red or blue as humans are sensitive to the green channel.

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

Matlab rgb2gray: Matlab's inbuilt function converts RGB values to grayscale values by forming a weighted sum of the R, G, and B components

$$i = 0.2989R + 0.587G + 0.114B$$

2.3 More on Color Spaces: CMYK Color Space

The CMYK color space stores the values for Cyan, Magenta, Yellow and Black. CMYK is a *subtractive* color space – it works by subtracting the colors from white, as opposed to adding the colors to black like RGB does. The subtractive nature of makes it ideal for printing on white paper.

Due to the extra channel CMYK images are one third bigger than RGB images; however, despite the difference in the file size, the information content of an RGB image is the same as that of a CMYK image.

3 Intrinsic Image Decomposition

3.1 Other Intrinsic Components

Other than reflectance and shading, images can be decomposed into:

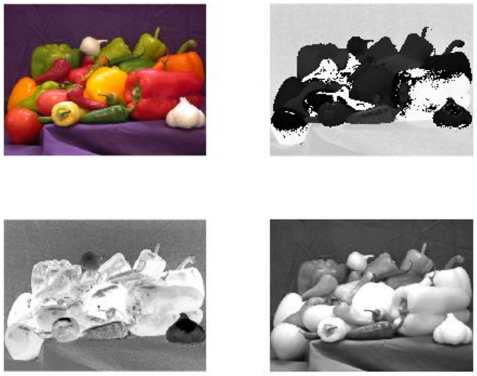

Figure 15: Hue, Saturation and Value components of the image in HSV color space (top-left).

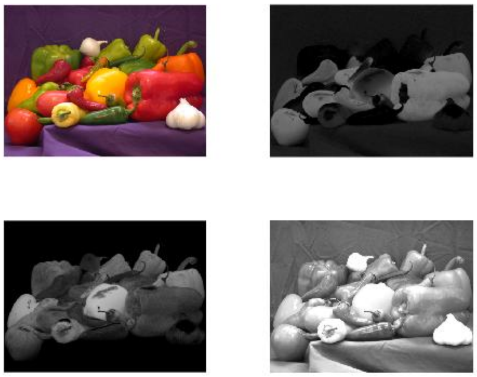

Figure 16: The red-green channel, the blue-yellow channel and the luminance channel of the image in the Opponent color space (top-left).

1. *Texture* and *Piecewise Smooth Content*; such decomposition is used in in image analysis and synthesis, and image coding. A particular application where such decomposition would be useful is filling in missing parts of an image. For example, [4] provide a method for such decomposition.

2. *Components of different frequencies*; typically achieved by different kinds of wavelet transformations. This type of decomposition is often used in image compression and analysis.

3.2 Synthetic Images

The reason most of the intrinsic image decomposition datasets are composed of synthetic images is that it is very hard to extract the ground truth reflectance component from natural images: special lighting conditions are required for this to be possible. This is the reason the famous MIT dataset [5] of ground truth image decompositions contains only 20 images. With synthetic images there is no such problem as the ground truth is known, and thus it is possible to generate millions of correct image decompositions.

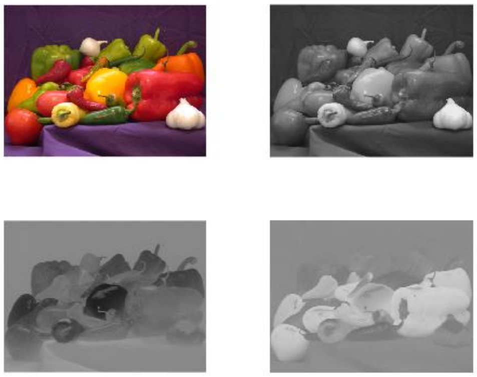

Figure 17: Luminance Y and two chroma components of the image in YCbCr color space (top-left).

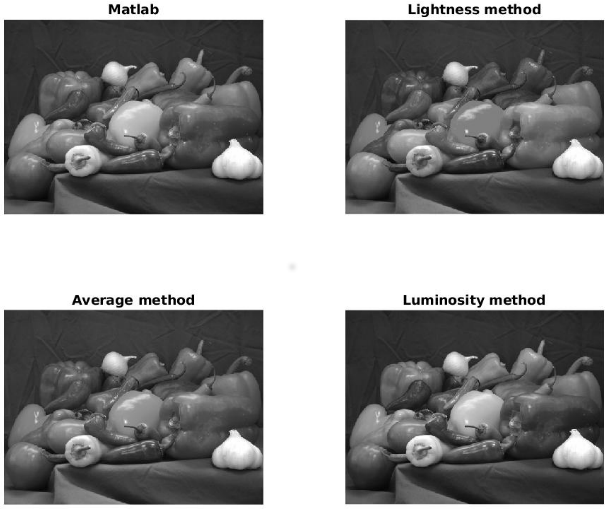

Figure 18: Four methods are used to generate the grayscale images from the original image

3.3 Image formation

Figure 19 shows the original image, its reflectance and shading components, and the image reconstructed from these components. We observe the high quality of the resulting reconstruction.

3.4 Recoloring

The true color of the ball material has the RGB values of (184 141 108). Figure 20 shows the images with the ball recolored into pure green (0, 255, 0), and magenta (255, 0, 255).

Although the ball was recolored with pure colors, the colors of the resulting images are not pure due to shading, which reduces the intensity of the colors. The shading component is not uniform, and thus the colors of the resulting images are not uniform either.

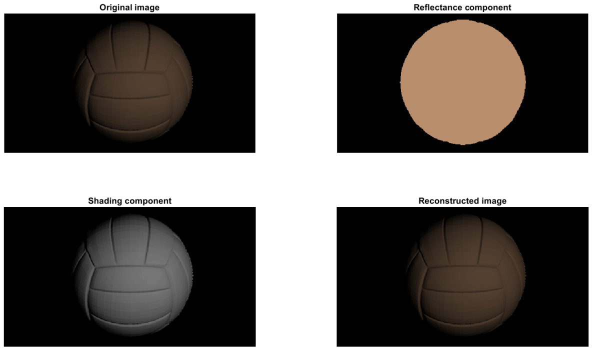

Figure 19: Image reconstruction from its reflectance and shading components.

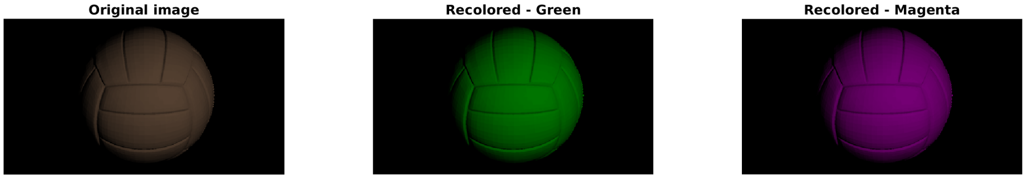

Figure 20: Recolored images.

4 Color Constancy

4.1 The Grey-World algorithm

The Grey-World algorithm assumes that *the average reflectance of surfaces in the world is achromatic*, that is, the mean color of the correct image is grey. This works well in cases when there are multiple objects of different colors in the image, as demonstrated in Figure 21. However, when the RGB channels of the image are not approximately balanced, the results of Grey-World algorithm are very far from optimal. Figure 22 demonstrates such a violation of the Grey-World assumption.

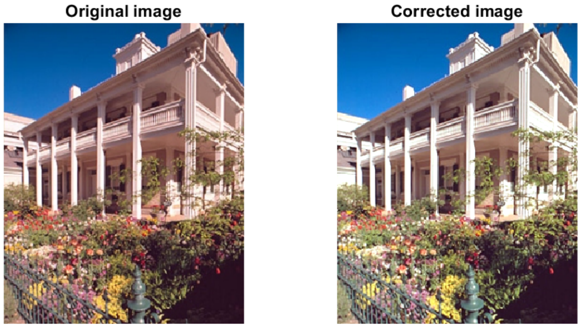

Figure 21: Application of the Grey-World algorithm to an image.

4.2 Alternatives to the Grey-World algorithm

This report shortly describes two alternatives to the Grey-World algorithm: the *max-RGB algorithm* and the *Grey-Edge algorithm*.

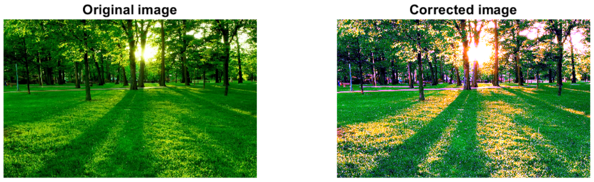

Figure 22: The image contains significantly more green compared to red and blue; the Grey-World algorithm performs poorly on such an image.

The max-RGB algorithm assumes that the *maximum reflectances of all channels are equal*:

$$\max_{\boldsymbol{x}} R(\boldsymbol{x}) = \max_{\boldsymbol{x}} G(\boldsymbol{x}) = \max_{\boldsymbol{x}} B(\boldsymbol{x}) = k\boldsymbol{e}$$

This method is very similar to the White-Patch algorithm, with the only difference being the possibility of the maxima of the separate channels being in different locations of the image. The rest of the correction procedure is exactly the same as the White-Patch procedure: the corrected image is obtained by multiplying each channel by the ratio of the desired value of the max pixel (1 for all channels) to the maximum value of the channel in the original image.

The main assumption of the Grey-Edge algorithm [6] is that *the average of the reflectance differences in a scene is achromatic*. The color of the light source is derived by a normal averaging operation over the derivatives of the channels:

$$\frac{\int |\boldsymbol{f}_{\boldsymbol{x}}(\boldsymbol{x})| d\boldsymbol{x}}{\int d\boldsymbol{x}} = \int_w e(\lambda) \left(\frac{\int |s_{\boldsymbol{x}}(\lambda, \boldsymbol{x})| d\boldsymbol{x}}{\int d\boldsymbol{x}} \right) \boldsymbol{c}(\lambda) d\lambda,$$

where $|\boldsymbol{f}_{\boldsymbol{x}}(\boldsymbol{x})| = (|R_{\boldsymbol{x}}(\boldsymbol{x})|, |G_{\boldsymbol{x}}(\boldsymbol{x})|, |B_{\boldsymbol{x}}(\boldsymbol{x})|)^T$. The corrected image is obtained by multiplying each channel by the ratio of the desired illuminant of the channel (0.5 for all channels) to the illuminant of the channel in the original image.

References

[1] Robert J Woodham. Photometric method for determining surface orientation from multiple images. *Optical engineering*, 19(1):191139, 1980.

[2] David A Forsyth and Jean Ponce. *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference, 2002.

[3] Margarita Bratkova, Solomon Boulos, and Peter Shirley. orgb: a practical opponent color space for computer graphics. *IEEE Computer Graphics and Applications*, 29(1), 2009.

[4] Jean-Luc Starck, Mikael Elad, and David L Donoho. Image decomposition: Separation of texture from piecewise smooth content. In *Wavelets: Applications in Signal and Image Processing X*, volume 5207, pages 571–583. International Society for Optics and Photonics, 2003.

[5] Roger Grosse, Micah K. Johnson, Edward H. Adelson, and William T. Freeman. Ground-truth dataset and baseline evaluations for intrinsic image algorithms. In *International Conference on Computer Vision*, pages 2335–2342, 2009.

[6] Joost Van De Weijer and Theo Gevers. Color constancy based on the grey-edge hypothesis. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 2, pages II–722. IEEE, 2005.