NTNU

TTK4255

Robotvision

Hyperspectral imaging

Mads Formo

March 24, 2020

Contents

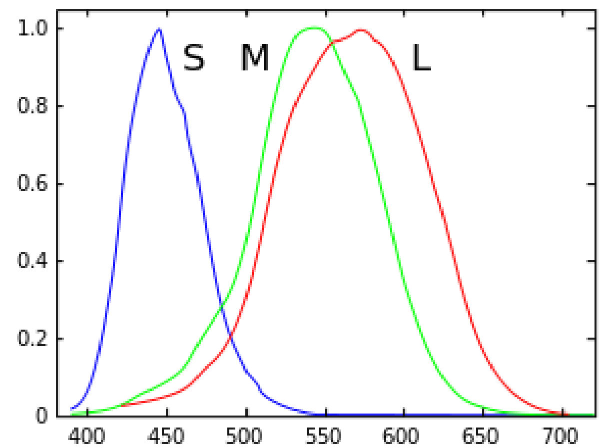

Figure 1: Graph for the human color sensitivity curves, according to Wikipedia [1]

1 Getting familiar with the data

1.1 Finding the spectral resolution

To find the spectral resoulution of the dataset, we load the *hico_wl* array, which contains the wavelength corresponding to band i. We loop through the array and compare each wavelength i with the the previous wavelenght *i-1* and we find that the average distance between the wavelengths is $5.728nm$, which seems to be constant between all wavelengths.

1.2 Relation to human color perception

The color sensitivity of the human eye is shown in fig. 1. As we can see, blue color has a peak around $450nm$ (S-curve), green peaks at $550nm$ (M-curve), and red at $600nm$ (L-curve).

1.3 Create a pseudo RGB image from the hyperspectral bands

From the *hico_wl* array, we find that Blue (450nm) is located at index $i = 8$, green (550nm) at $i = 25$, and finally red (600nm) at $i = 34$. We combine these indices from the HICO dataset and show it as an image to create a pseudo RGB image, shown in fig. 2.

1.4 Representative spectra for selected points

We want to look at the representative spectra of the points (20,20), (100,70) and (400,30), which is in deep water, shallow water and vegetation respectively. As we can see in fig. 3, we see that there is a clear difference in the spectra between water and vegetation. Both have amplitude peaks at the lower end of the spectra and then drop off in power as the wavelength increases. Vegetation however increases again in power at a wavelength of around 700nm, while the water is still decreasing. The findings here seem to agree to the findings of Lucke et al [2] as the general shape of the curves matches those of figure 12 in that report.

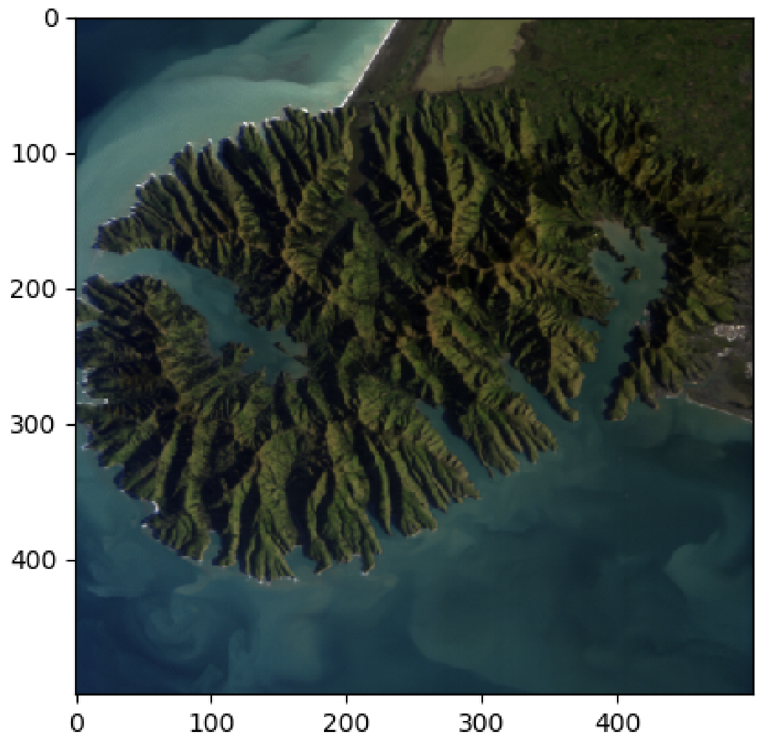

Figure 2: Pseudo RGB image, showing R (600nm), G (550nm), B (450nm)

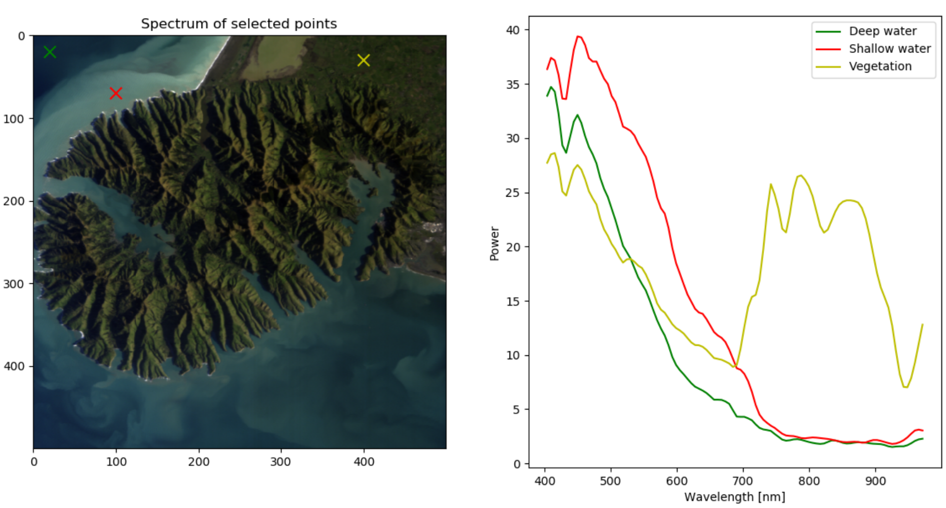

Figure 3: Representative spectra of specific points

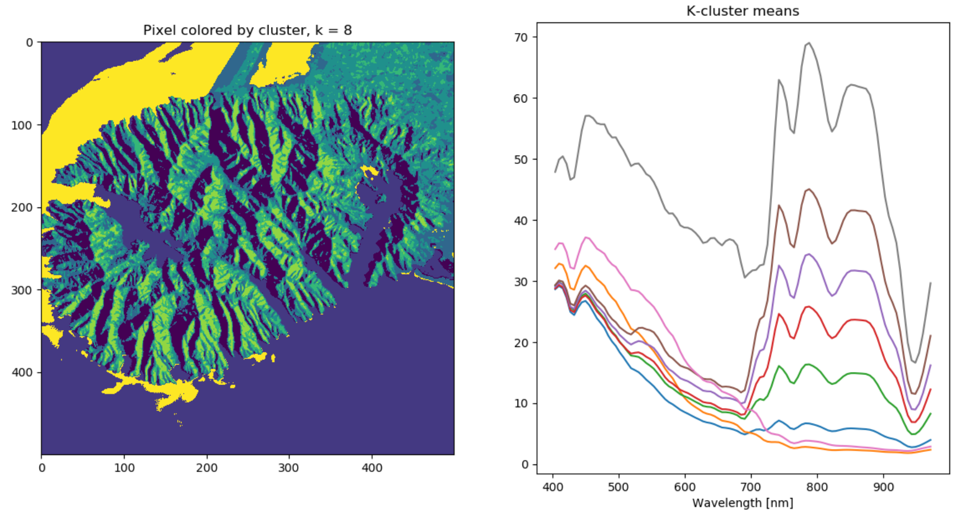

Figure 4: K-mean clusters of the image

2 Classification & Bio-geophysical Parameter Retrieval

2.1 Can we predict where there is chlorophyll through classification?

We will use *K-Means clustering* to classify the data. K-means clustering is a unsupervised learning algorithm that can be used to classify and cluster data into k different clusters. The data points are adjusted iteratively until all points are associated with the nearest cluster. We want to cluster each observation (pixels, with n spectral channels) into a specific cluster (environment class, ie. deep water, shallow water, vegetation).

As we can see from fig. 3, we know that those three different points have distinctly different spectra, thus it should be possible to classify them accordingly. The results of a K-mean clustering, run with Spectral Python's *kmeans* function [3], can be seen in fig. 4. We clearly see different classes for water, land, and vegetation, the latter containing lots of chlorophyll. We also see a very distinct class along the coast on the upper part of the image. This may very well be a collection of chlorophyll, but it might also just be shallow water, or more likely a combination of both.

2.2 How well can we directly estimate the chlorophyll content?

We use the NASA OBPG algorithm, defined in equation 4 in the assignment [4], as well as the parameters given there, to try to visualize the chlorophyll contents. Using the closest available wavelengths in the dataset, $\lambda_{green} = 553$ (i = 26) and $\lambda_{blue} = [444, 490, 507]$ (i = [7, 15, 18]). The results can be seen in fig. 5. We can clearly see high concentrations on the north west coast, same place as in fig. 4, but now we also see quite a bit on the southern coast as well. Thus it seems that this algorithm performs better than the k-means clustering.

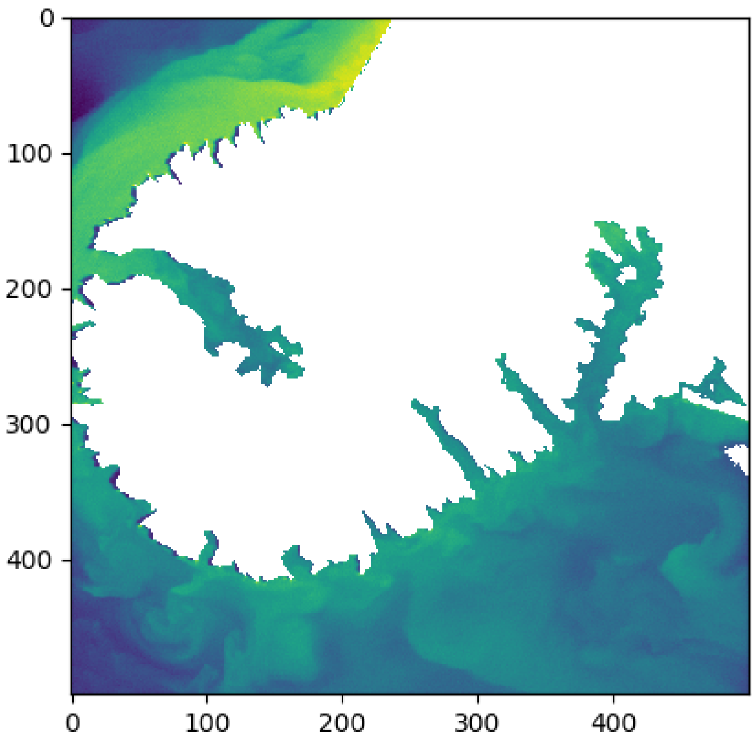

Figure 5: Results from the NASA OBPG algorithm, showing chlorophyll concentrations

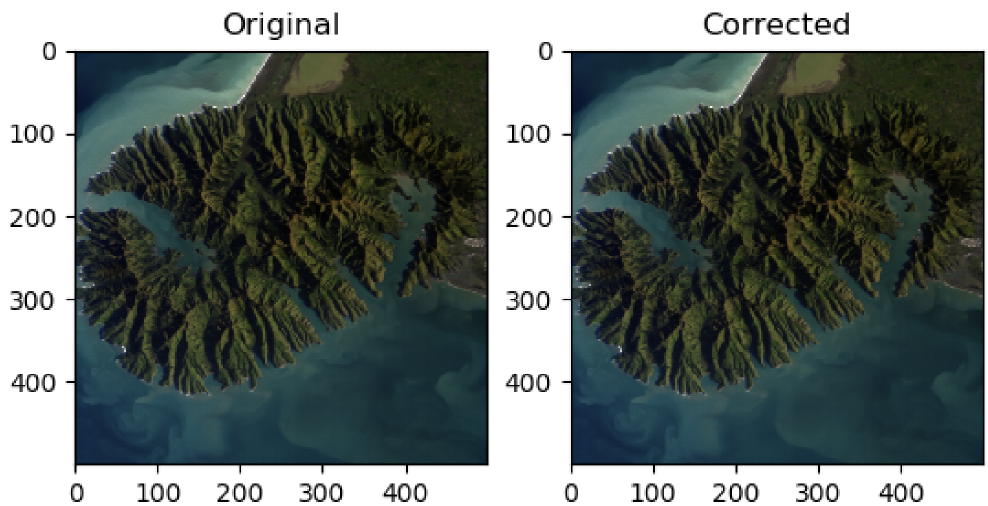

Figure 6: Pseudo RGB image of the atmosphere corrected image

2.3 How can we estimate the reflectance from the surface of the ocean?

The data in the HICO dataset actually contains the measurement of radiance exiting the top of the atmosphere, and not the radiance of the water directly. Therefore we must recover the radiance R_{rs} of the water from the top of atmosphere (TOA) measurements. This is done with the empirical line (ELM) method, as described in [4], on the form eq. (1). Where a and b are terms that model the absorption of light in the atmosphere, which is to be estimated, and L is the measured value in the HICO dataset.

$$R_{rs}(\lambda) = \frac{L(\lambda) - b(\lambda)}{a(\lambda)} \tag{1}$$

After performing the atmospheric correction, the resulting image is shown in fig. 6. It is very difficult to see a clear difference between the two by only looking at the image, but if we look at the pixel values for the red, green and blue channels separately, we see that the blue color channel is only about 2% of the original, uncorrected pixel value, while the red and blue channels are about 9-10% of their original values. Thus it would seem that the atmospheric correction removes some of the blue color of the image.

2.4 Compute chlorophyll concentration using atmosphere-corrected data

2.5 Classify land versus water

2.6 Other bio-geophysical parameters

2.7 Alternative atmospheric correction methods

3 Dimensionality Reduction & Noise Filtering

3.1 What is dimensionality reduction?

3.2 Principal Component Analysis (PCA)

3.3 How does dimensionality reduction via PCA afect classification?

3.4 Maximum Noise Fraction

3.5 Maximum Noise Fraction on HICO noisy

3.6 Discuss your results

3.7 How can we best use the subspace?

4 Fun but definitely hard problems

4.1 Deep learning

4.2 Multispectral-hyperspectral image fusion

4.3 Spatial-spectral methods

4.4 Locating methane emissions

References

[1] Wikipedia. *Spectral sensitivity*. Jan. 2020. URL: https://en.wikipedia.org/wiki/Spectral_sensitivity.

[2] Robert L. Lucke et al. "Hyperspectral Imager for the Coastal Ocean: instrument description and first images". In: *Appl. Opt.* 50.11 (Apr. 2011), pp. 1501–1516. DOI: 10.1364/AO.50.001501. URL: http://ao.osa.org/abstract.cfm?URI=ao-50-11-1501.

[3] Thomas Boggs. *SpectralPython*. 2014. URL: https://www.spectralpython.net.

[4] Sivert Bakken, Joe Garret, and Simen Haugo. "Hyper Spectral Imaging Project". In: *TTK4255 Robotic Vision, NTNU* (Feb. 2020). URL: https://ntnu.blackboard.com/bbcswebdav/pid-881058-dt-content-rid-25343529_1/xid-25343529_1.