30540 - Mapping from Aerial and Satellite Images
Partial Unmixing

Anders Richard Pankoke Holm
s144125

13/02-2019

Constrained Energy Minimization (CEM)

Within the field of remote sensing, we deal with analyzing for example objects photographed from far away. That could be the land, ocean etc. which we capture data of from satellites or through aerial photography. The data we get, are images in the different light spectra. In our everyday life, we see images displayed using red, green and blue (RGB). An object appearing red, say a rose, will absorb most of the green and blue light, whereas the red light is reflected, thus making the rose appear red. But these are only the colors we are able to see with the naked eye. The visible part of the electromagnetic spectrum spans only a limited amount of the entire spectrum. This means that light of higher or lower wavelengths than the visible light are also being reflected, but not observed by humans. We instead need special equipment to observe for example ultra-violet or infrared light.

The type of light an object reflects, can tell a lot about the object, and when taking images from several kilometers above the ground, it can be very difficult to distinguish objects, when only looking at the visible light. For example, vegetation often give off a lot of infrared radiation. Different types of plants would reflect different wavelengths of light, and in different amounts, such that an object can be characterized by it's own spectrum. For this exercise we will be working an image where several spectra are provided. Each spectrum is an image itself, built up as a matrix, with the respective values of observed light for each pixel. Thereby when we layer all these images, we can extract a spectrum for each pixel. With this information, we are able to locate objects with similar characteristics to that of a chosen pixel.

In order to extract this sort of information, and compare the pixels, we will used a method called Constrained Energy Minimization. The value of the reflected wavelength of light in each pixel spectrum is called an end member (see figure 1).

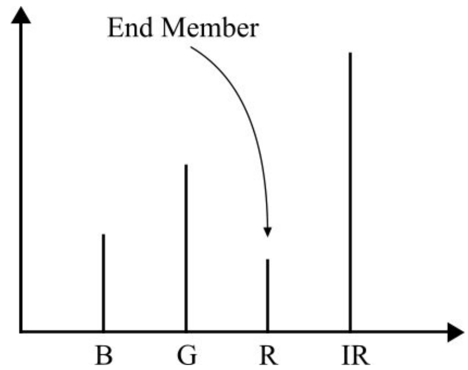

Figure 1: Example of a pixel spectrum consisting of infrared, red, green and blue light.

Such a spectrum is said to contain 100 % of the abundances of chosen object. If we know all the spectra for all end-members in our image, then we are able to perform *Full Unmixing*. Unfortunately, we wont always have this information, so as an alternative, we can do a *Partial Unmixing*. Here we arrange our known end-members into the column vector $\boldsymbol{d}$, along with corresponding abundances α_p, and the unknown end-members are put into a matrix $\boldsymbol{U}$ along with the corresponding abundances γ. Our observations are then given by:

$$\boldsymbol{r} = \boldsymbol{d}\alpha_p + \boldsymbol{U}\gamma + \boldsymbol{n} \tag{1}$$

where $\boldsymbol{n}$ is noise. We want the projected ideal abundances (abundances very similar to chosen pixel's) to be 1, that is $\boldsymbol{w}^T\boldsymbol{d} = 1$, the output to have an expected value of 0, such that $E\{\boldsymbol{w}^T\boldsymbol{r}\} = 0$, and at last we want to minimize the squared difference between actual output and expected output $E\{(\boldsymbol{w}^T\boldsymbol{r} - 0)^2\}$. This can be done using Lagrange multiplier -2λ. This gives us, that $\boldsymbol{w}$ can be found by:

$$\boldsymbol{w} = \frac{\Sigma^{-1}\boldsymbol{d}}{\boldsymbol{d}^T\Sigma^{-1}\boldsymbol{d}} \tag{2}$$

where Σ, is the variance co-variance (or dispersion) matrix of the spectral bands.

Discussion of results

For doing CEM, two MATLAB scripts were provided along with data containing a 30 spectral bands AVIRIS image (ranging from visible green at 520 nm to near-infrareds bands at 2.33 μm). Given an image's spectral bands arranged into a 3D matrix, where the third dimension is the layered spectra, the function "`imshowrgb.pdf`, allows us to display an image using 3 spectral bands. As mentioned earlier, we normally display images using RGB colors, but for cases where the reflected light is from the non-visible part of the spectrum, we can display these images in false colors. In the script

cem30540.m we perform the CEM operation. The data is loaded in as binary file of the format uint8, meaning that for every pixel in every spectrum, a value is given for the reflected light as a number between 0 and 255. With our $N = 30$ variables we calculate the dispersion matrix Σ by arranging all the pixels of one spectrum into a vector. The resolution of our images are 180×360, thus we get a matrix $\boldsymbol{X}$ that has dimensions 64800×30. The mean of each spectrum is then subtracted from it's respective column, and we get the a new matrix $\overline{\boldsymbol{X}}$, from which the dispersion matrix is finally computed. $\boldsymbol{d}$ is extracted by taking all end members of a chosen pixel spectrum, and arranging them as a column vector. Finally, $\boldsymbol{w}$ is computed from eq. (2), which will give us our abundances through $\alpha = \overline{\boldsymbol{X}}\boldsymbol{w}$. This is rearranged back into the same index the pixel originally had.

In the cem30540.m script, we get to pick out a pixel that will will provide us with our desired spectrum for computing abundances.

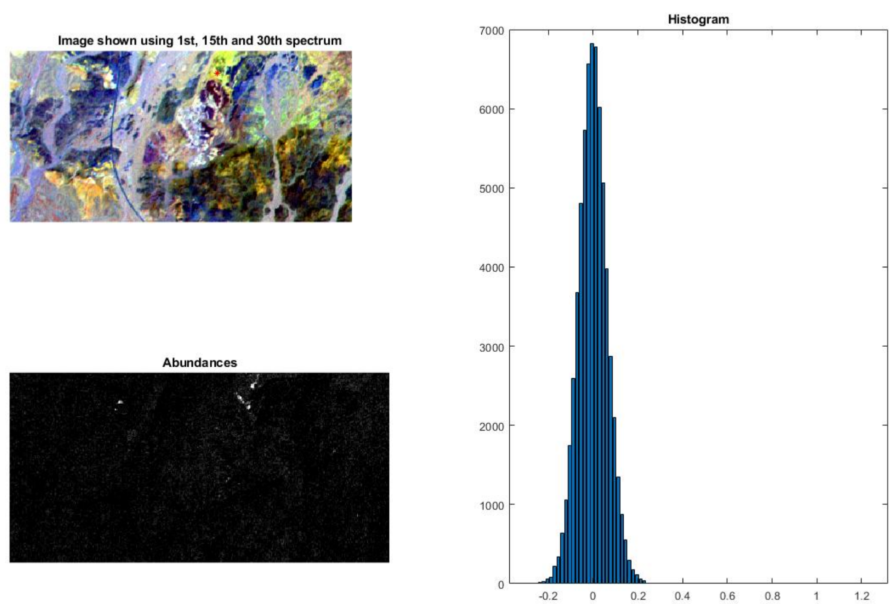

Figure 2: Example of picking a good end-member.

In figure 2 we see an example, where a good end-member was chosen. The end-member we chose has an abundance of 1, and those that are very similar to the chosen pixel come close to 1. We see how we are able to locate similar terrain, for example vegetation in small islands, e.i. it is more structured. The noise in the picture is minimal, which is also illustrated from the histogram, which has many counts centered around 0 (given that we constrained for $E\{\boldsymbol{w}^T\boldsymbol{r}\} = 0$), thus making it narrow.

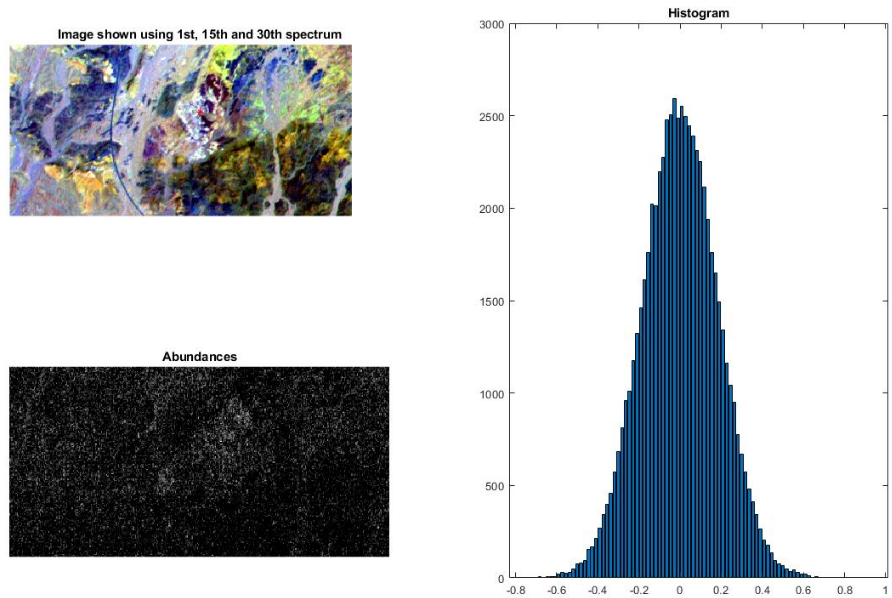

Figure 3: Example of picking a bad end-member.

In this second and final example, a bad end-member was chosen. This time there is a lot of noise in the abundances plot. We also see from the histogram, the distribution is much wider around 0, also indicating a bad end-member.