TTK4255 Robotic Vision, Spring 2020

Hyperspectral Imaging Project

Sivert Bakken, Joe Garrett and Simen Haugo

Introduction

Just as imaging with a red-green-blue (RGB) camera reveals information that is not visible in a grey-scale (monochromatic) imagery, hyperspectral imaging acquires a much more distinct spectral signature of the materials being imaged relative to an RGB imager. Whereas RGB imagery acquires three spectral bands (colors), a hyperspectral imager acquires around 100 different bands. Achieving this requires customized camera geometries, such as the one depicted in Figure 1. Because spectral images have both spatial (two-dimensional) and a spectral dimension, they are called image cubes.

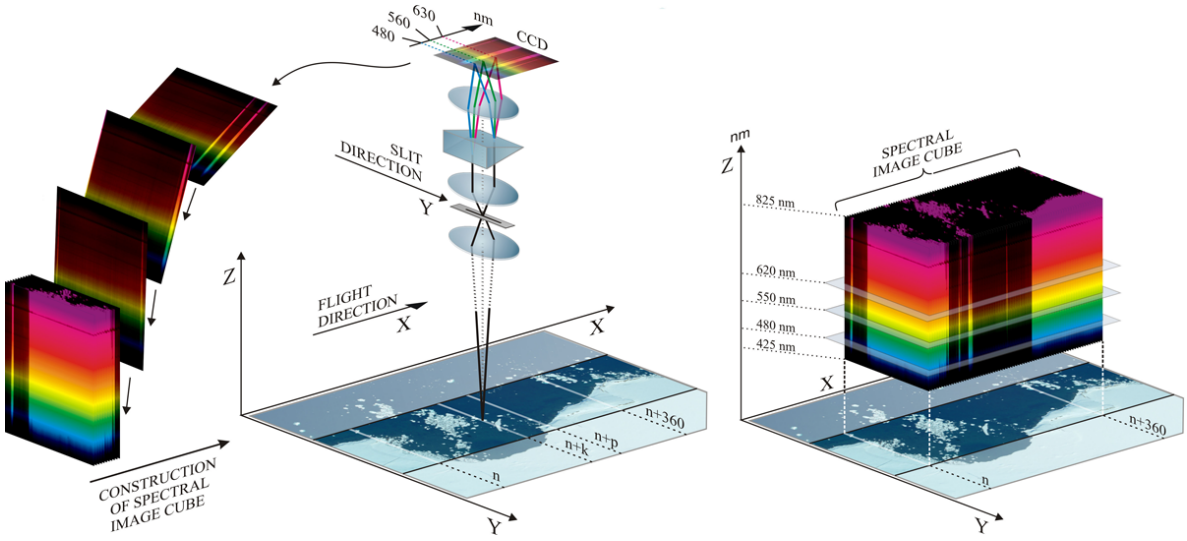

Figure 1: Visualization of image cubes acquired using a push broom hyperspectral imager, which is often the instrument of choice for airborne missions. Source: Zsolt Volent, Geir Johnsen, and Fred Sigernes "Kelp forest mapping by use of airborne hyperspectral imager," Journal of Applied Remote Sensing 1(1), 011503 (1 December 2007).

Analysis of an image cube can be difficult for several reasons, e.g. spectra are not only dependent on the materials in the scene, but they also depend on the shape of the objects, the spectra of the light source and the light position. Operationally, we'll only be able to measure the apparent optical properties, while what we're interested in are the inherent optical properties. This assignment will not put much focus on these challenges.

Some common applications for hyperspectral and multispectral imagery are food security, defense technologies, Earth sciences and health. All these fields can benefit from remote, non-intrusive measurements of the object of interest. If you want to learn more about hyperspectral imaging some books are available through the internet. [1]

Cube and matrix representation

Linear algebra is the typical tool of choice for analyzing hyperspectral data. Although any single spectral channel (band) of the data cube is amenable to linear algebra, it is common to flatten the whole data cube into a matrix representation. We then ignore the original spatial layout of the pixels, and simply consider the data as a set of spectral measurements. This is shown in equation (1) and (2), where each $\mathbf{x}_{i,j}$ is a vector of spectral measurements (for different wavelengths λ) at pixel i, j.

$$\mathbf{X}_{\text{Cube}} = \begin{bmatrix} \mathbf{x_{1,1}} & \mathbf{x_{1,2}} & \cdots & \mathbf{x_{1,W}} \\ \mathbf{x_{2,1}} & \mathbf{x_{2,2}} & \cdots & \mathbf{x_{2,W}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x_{H,1}} & \mathbf{x_{H,2}} & \cdots & \mathbf{x_{H,W}} \end{bmatrix}, \tag{1}$$

$$\mathbf{X}_{\text{Matrix}} = \begin{bmatrix} \mathbf{x_{1,1}} & \cdots & \mathbf{x_{1,W}} & \mathbf{x_{2,1}} & \cdots & \mathbf{x_{2,W}} & \cdots & \mathbf{x_{H,1}} \cdots & \mathbf{x_{H,W}} \end{bmatrix} \tag{2}$$

For convenience, we will later in the text refer to the columns of $\mathbf{X}_{\text{Matrix}}$ without a double subscript, and simply write $\mathbf{x}_i$, where i ranges from 1 to $N = HW$. That is,

$$\mathbf{X}_{\text{Matrix}} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_N \end{bmatrix} \tag{3}$$

Datasets

For most of the tasks you will work with the HICO.mat dataset provided in the zip file for this project. It contains a subset of data acquired over the coast of New Zealand near Christchurch, on June 16, 2011, by the HICO (Hyperspectral Imager for the Coastal Ocean) instrument, onboard the international space station. For this project, wavelengths in the range 400 to 1000 nm were kept, but bands more prone to noise were removed.

The HICO.mat file contains the data entries listed in Table 1. In Matlab, you can read .mat files using the load function, which will make the data entries visible in the workspace. In Python, you can read .mat files using scipy.io.loadmat.

The data directory also includes calibrated reflectance spectra for deep and shallow seawater. These are provided for the same wavelengths as the HICO data, although values for wavelengths outside the range 438 nm to 730 nm are not valid. This is specified by a boolean mask array indicating whether the wavelength of band i has a valid value.

In addition to the provided dataset, several other datasets are available online at Hyperspectral Remote Sensing Scenes, but you are not required to consider these.

[1] Eismann, Michael Theodore. "Hyperspectral remote sensing." Bellingham: SPIE, 2012.

Name	dims(x, y, λ)	Comment
HICO_original	500, 500, 100	Original data set
HICO_noisy	500, 500, 100	Data with noise
HICO_wl	1, 1, 100	HICO wavelengths
deep_water_Rrs	1, 1, 100	Calibrated reflectances for deep seawater
shallow_water_Rrs	1, 1, 100	Calibrated reflectances for shallow seawater
valid_bands_Rrs	1, 1, 100	Boolean validity mask array

Table 1: Top three are provided in HICO.mat. Bottom three are in the data directory.

Programming language and provided code

You are free to use any programming language for this project. We recommend using Matlab or Python, as these have good libraries for working with hyperspectral images and for the tasks in this project. For this project, you probably want the following libraries:

- *Matlab*: Image Processing Toolbox (you may already have this installed).
- *Python*: SpectralPy and SciPy.

An included script (example.py/m) shows how to load the hyperspectral data in Python and Matlab. It also shows how to convert between the cube and matrix representation with appropriate usage of the reshape function.

Evaluation

Your project score is determined based on your answers to each question. The score attainable per question is written next to the title as the percentage in parenthesis. Answering fully or partially the questions in your report can give up to 100% score for the project, which counts toward 20% of your final grade.

1 Get familiar with the data (5%)

The images you have worked with so far in the course have contained three channels. We have loosely described these as red, green and blue (RGB) channels, but we have not thought too deeply about what these channels are actually measuring, how they are obtained by the sensor or how they are related to physical quantities. The science behind these things is known as radiometry, and becomes important when working with hyperspectral data.

(a) Find the spectral resolution

The HICO_original data contains 100 single-channel 500 x 500 pixel images. Each image corresponds to a band of wavelengths. The sensor measured wavelengths between 400 and 1000 nm. The distance between the bands is called the spectral resolution. The hico_wl array contains the physical wavelength corresponding to band i. Load the array and find the spectral resolution by inspecting the wavelength values. Is it the same between each band?

(b) Relation to human color perception

Human color perception is enabled by having three types of receptors on our eyes, called cones, which are selectively sensitive to parts of the spectrum. The sensitivity of each cone is described by a spectral response curve. Use Google to find a graph for the human color sensitivity curves. At which wavelengths do the sensitivity curves peak? Which curve is associated with red, green and blue color perception, respectively?

(c) Create a pseudo RGB image from the hyperspectral bands

If we select three bands out of the hyperspectral data which are close to the three primary colors, then we can combine them to produce a pseudo RGB image. Use the hico_wl array and the peak values you found in (b) to extract three channels from HICO_original, corresponding to red, green and blue. Normalize the data to the range $[0, 1]$ and display the RGB image. Include the figure in your report.

(d) Representative spectra for selected points

The points (20,20), (100,70) and (400,30) are in deep water, shallow water and vegetation, respectively. Plot the RGB image and the three points on top. Next to it, plot the spectral measurement at each point. NB! The points are (column, row) but image data is accessed by (row, column). Include a legend for both plots ("deep water", "shallow water", "vegetation"), and label the x and y axes of the spectral data with appropriate units (the hyperspectral data measures radiance in units $\mathrm{W/m^{-2}}\mu\mathrm{m^{-1}sr^{-1}}$). Include the figure in your report.

The HICO instrument and mission was detailed in this paper, which is also included in the zip. Do your results above match Fig. 12 in the paper?

2 Classification & Bio-geophysical Parameter Retrieval (45%)

Bio-geophysical parameter retrieval, i.e. inferring physical properties of a scene based on light, is one application of hyperspectral imaging. For example, a scientist might be interested in where there is significant chlorophyll in the ocean. In this section, we will investigate a few strategies for extracting useful information from a hyperspectral image.

(a) Can we predict where there is chlorophyll through classification? (10%)

k-means clustering is a general machine learning algorithm. First, explain how it can be applied to hyperspectral image data and what information can be obtained. Specifically, what are the "observations" and what are the "clusters"? You may use your plot from Task 1d to aid your explanation.

Next, find an implementation of k-means clustering. For example, imsegkmeans (Matlab) or kmeans (Python/SpectralPy). Run k-means clustering on HICO_original. Generate an image where pixels are colored based on their class label and plot the k cluster means next to the image. Try values of k between 1 and 10. What do the clusters represent? How many different classes do you expect to find? Do you see a distinct class that relates to chlorophyll? Include a figure for one value of k in your report.

(b) How well can we directly estimate the chlorophyll content? (5%)

The NASA OBPG algorithm [2] can be used to compute the near-surface concentration of chlorophyll. It is based on an empirically-derived relationship between chlorophyll concentration and reflectance in the blue-to-green spectrum:

$$\log_{10}(\text{chlor}_a) = a_0 + \sum_{i=1}^{4} a_i \left(\log_{10} \left(\frac{\max_j R_{rs}(\lambda_{\text{blue},j})}{R_{rs}(\lambda_{\text{green}})} \right) \right)^i \tag{4}$$

Here, chlor_a is the concentration of chlorophyll-a in mg/m^3, $a_{0\ldots4}$ are empirically-derived coefficients, and $R_{rs}(\lambda)$ is the remote sensing reflectance at wavelength λ at the point of interest. For this task, assume that R_{rs} is equal to the sensor measurement (i.e. the values stored in HICO_original).

Implement the NASA OBPG algorithm to estimate the chlorophyll content of the water bodies in HICO_original and generate a chlorophyll intensity image. Use the coefficients from the OC4 SeaWiFS analysis (found in the top row on the website in the footnote)

$$a_0 = 0.3272, \quad a_1 = -2.9940, \quad a_2 = 2.7218, \quad a_3 = -1.2259, \quad a_4 = -0.5683$$

and use the wavelengths $\lambda_{\text{green}} = 555$ and $\lambda_{\text{blue}} = (443, 490, 510)$. Use the bands that are closest to the wavelength center of the relevant bands. Does this algorithm seem to show the location of chlorophyll better than k-means classification?

[2] https://oceancolor.gsfc.nasa.gov/atbd/chlor_a/

(c) How can we estimate the reflectance from the surface of the ocean? (10%)

In (b) we assumed that the remote sensing reflectance R_{rs}, a property of the surface being measured, was equal to the values stored in our hyperspectral image. However, the HICO instrument actually measures the radiance exiting the top of the atmosphere (TOA). These two are not equal, and recovering R_{rs} from TOA measurements is known as atmospheric correction. Here you will use the empirical line (ELM) method. In this method, $R_{rs}(\lambda)$ is computed from a simple model of the form

$$R_{rs}(\lambda) := \rho(\lambda) = \frac{L(\lambda) - b(\lambda)}{a(\lambda)} \tag{5}$$

where $L(\lambda)$ is the measured TOA radiance (i.e. the values stored in the HICO image cube), and a and b are terms that model the absorption of light as it passes through the atmosphere and the contribution from atmospheric scattering.

Before we can use this model, we need to estimate a and b. ELM does this as follows. At a given λ, assume we have a set of n known (calibrated) reflectance values ρ_i and corresponding measured radiance values L_i, $i = 1...n$. By rearranging the above model, we can set up n equations of the form

$$a\rho_i + b = L_i, \quad i = 1..n \tag{6}$$

which can be written as a linear system of equations $\mathbf{Ax} = \mathbf{y}$, where

$$\mathbf{A} = \begin{bmatrix} \rho_1 & 1 \\ \vdots & \vdots \\ \rho_n & 1 \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_n \end{bmatrix}, \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} a \\ b \end{bmatrix} \tag{7}$$

The best estimate of a and b, in the least squares sense, can then be computed by solving the system for $\mathbf{x}$. This gives the value of a and b at one wavelength λ. To obtain the complete set of a and b values, you must solve a new linear system at each wavelength λ of interest.

Calibrated remote sensing reflectances, $R_{rs}(\lambda)$, for deep water and for shallow water are provided in the data directory for the same wavelengths as the HICO data (see Table 1). Values for wavelengths outside the range 438-730 nm are invalid, however this is not an issue as the tasks only use a few specific wavelengths within this range.

Use the calibrated reflectances and sample the hyperspectral image cube at two appropriate points to obtain the corresponding top-of-atmosphere radiance $L_i(\lambda)$ for deep and shallow water, respectively. This gives a system of $n = 2$ equations, which can be solved for a and b at each wavelength of interest. You can then apply the atmosphere correction to each band of interest in the HICO image cube using equation 5.

Apply the atmospheric correction for each valid band (corresponding to 438-730 nm), and display a pseudo RGB rendering of the scene before and after atmospheric correction. Include the figure in your report.

(d) Compute chlorophyll concentration using atmosphere-corrected data (5%)

Use the atmosphere-corrected data to compute the chlorophyll concentration image again. Include a figure comparing the atmosphere-corrected results with your results in (b) and comment on the differences.

For correctness, you may compare your results with NASA (clickable link) for the same scene. Using the method presented here, you will be able to get within the same order of magnitude (i.e. chlorophyll concentrations between 1 and 10 mg/m^3 in the water areas). Click the color map button in the top-left for how to interpret their chlorophyll concentration image. Note that their chlorophyll image is plotted with logarithmic scale, i.e. they are plotting $\log_{10}(\text{chlor}_a)$ instead of chlor_a.

For easier visualization, you may want to mask away the land as is done on the linked website. In the next task you will do this yourself, but for this task you may use the provided land mask image. See the example code for how to apply the mask. You will also want to adjust the minimum and maximum values (see documentation for imshow in Matlab and Python). Based on the NASA data, you can infer that appropriate minimum and maximum values are around 0.01 to 10 mg/m^3 (linear scale), or equivalently -2 to 1 (logarithmic).

Also, run k-means clustering on the atmosphere-corrected data within the valid bands. Does atmosphere correction make any difference to the k-means clustering? Include a figure in your report.

(e) Classify land versus water (5%)

In the above tasks you computed chlorophyll concentration for all pixels, but what we were interested in was the chlorophyll content of the water bodies. You should therefore find a way to classify land pixels versus water pixels and generate a binary mask image (each pixel is 0 or 1). You may use the k-means clustering algorithm with an appropriate value of k, or find an alternative method, e.g. Support Vector Machine (SVM). In your report, describe how you performed the classification and include a figure showing:

- The RGB image
- The RGB image with land pixels set to 0
- The chlorophyll concentration image with land pixels set to 0

(f) Other bio-geophysical parameters (5%)

The chlorophyll concentration is only one example of a bio-geophysical parameter. There are plenty of others that can be derived from the available spectral measurements (400-1000 nm). Can you find some examples of other bio-geophysical parameters? You may need to search online.

(g) Alternative atmospheric correction methods (5%)

What are the limitations of ELM? What might an alternative way of estimating the effect of the atmosphere be? Again, this task requires you to look online for relevant information.

3 Dimensionality Reduction & Noise Filtering (45%)

In this task, you will use different methods to remove redundant information from hyperspectral images. These methods are based on dimensionality reduction and come from the field of multivariate data analysis.

Important: To be able to use these methods, we will view a hyperspectral image cube, with $W \times H$ pixels and L spectral bands, as a matrix $\mathbf{X}$ with L rows and $N = WH$ columns. In this matrix form, each column contains the spectral data for a single pixel.

(a) What is dimensionality reduction? (5%)

A hyperspectral image cube can store WHL potentially different numbers. For example, HICO_original has 500×500 pixels and 100 spectral bands, and can therefore store $500 \times 500 \times 100 = 25$ million numbers. However, in real data we expect a large portion of these to be correlated. Consider your RGB image and spectral data plot from Task 1d. Do you see any potential correlations or opportunities to compress the data in the spatial or spectral domain, respectively?

(b) Principal Component Analysis (PCA) (5%)

The first method you will use is Principal Component Analysis (PCA). PCA approximates each column of $\mathbf{X}$ by a linear combination L-dimensional basis vectors. The basis vectors are shared for all columns, but each column has its own weights. Consider a matrix $\mathbf{X}$ with 100 rows and 1000 columns, and assume that you use 10 basis vectors. How many weights are associated with a single column? How many weights are there in total? How many numbers are needed to describe the approximated data in total? How much do we save compared to the original data? Finally, briefly explain how PCA chooses the basis vectors.

(c) How does dimensionality reduction via PCA affect classification? (10%)

Here you can choose to implement PCA yourself to receive a higher score, or you can find an existing implementation. If you implement PCA yourself, include your code. Run PCA on $\mathbf{X}$ using $P = 10$ components and compute the $P \times N$ matrix representing the compressed data. Run k-means on the compressed data and generate a classification image as in Task 2a. Try different values of P between 1 and 100. Include one or two figures in your report and discuss how PCA affects the k-means clustering.

(d) Maximum Noise Fraction (10%)

The second method you will use is the Maximum Noise Fraction transform (MNF), also known as Noise Adjusted Principal Components (NAPC), as described in the paper (Green et al., 1988), included in the zip. The method is described in Section II of the paper, but we summarize it here using our notation.

Assume that our signal has been corrupted by additive noise. The covariance matrix of our data can then be written as

$$\text{Cov}(\mathbf{X}) = \mathbf{\Sigma} = \mathbf{\Sigma}_s + \mathbf{\Sigma}_n \tag{8}$$

where $\mathbf{\Sigma}_s$ and $\mathbf{\Sigma}_n$ are the covariance matrices of the signal and the noise, respectively. The first step in MNF is to estimate the noise covariance matrix $\mathbf{\Sigma}_n$, but for this task you can assume that it is given to you. Given the noise covariance, the method proceeds to compute the eigendecomposition of $\mathbf{\Sigma}_n^{-1}\mathbf{\Sigma} = \mathbf{V}\text{diag}(\mu_1, ...\mu_L)\mathbf{V}^{-1}$, where the columns of $\mathbf{V}$ are the right-hand eigenvectors and $\mu_1 \leq \mu_2 \leq ... \leq \mu_L$ are the sorted eigenvalues. The paper then defines the MNF transform as

$$\mathbf{Y} = \mathbf{A}^T\mathbf{X} \tag{9}$$

where the rows of $\mathbf{A}^T = \mathbf{V}^{-1}$ are the left-hand eigenvectors. An approximation of the original signal can be reconstructed by multiplying the P first rows of $\mathbf{Y}$ with the P first columns of $\mathbf{V}$:

$$\hat{\mathbf{X}} = \mathbf{V}_{(:,\,1:\text{P})}\mathbf{Y}_{(1:\text{P},\,:)} \tag{10}$$

Before applying the method on the hyperspectral data, you will try it on two smaller datasets (task_3_case_*.mat). Both contain a 200×200×3 image I_noisy and the covariance matrix $\mathbf{\Sigma}_n$, but the covariance matrix is different for the two cases. The data also contains the original image I_original, which you will use to evaluate your results. The data can be loaded and converted into matrix form in the same way as HICO.mat. For convenience, we have included a script task_3_c.py/m that you can base your solution on. In the script you should do the following:

1. Run PCA on $\mathbf{X}$ and reconstruct the original signal using either 1 or 2 components.
2. Run MNF on $\mathbf{X}$ and reconstruct the original signal using the same number of components as for PCA. When implementing MNF, use np.linalg.eig (Python) and eig (Matlab) to compute the eigendecomposition. You need to sort the eigenvalues and eigenvectors in increasing order, as this is not done automatically.
3. Make a figure showing the uncontaminated image, noisy image, and reconstructed image from MNF and PCA. Above MNF and PCA, report the average pixel error between the reconstruction and the original uncontaminated signal. Include the figure in your report. (The figure is generated for you by the provided script).

Do the above for both datasets and comment on the following in your report:

Q1: What seems to be the optimal number of components (1 or 2)?

Q2: What error did you achieve with MNF versus PCA?

Q3: Inspect the noise covariance matrix $\mathbf{\Sigma}_n$ in both datasets, and the eigenvalues $\mu_1, ..., \mu_L$ as defined above. Based on this, do your results agree with the two cases in section II.A and II.B in Green's paper (page 3)?

(e) Maximum Noise Fraction on HICO_noisy (5%)

In order to use MNF on our actual data, we first need to estimate the noise component. One method for estimating noise is the between-neighbor difference, which assumes that neighboring pixels have similar spectra, and that their difference is therefore equal to the noise. In other words,

$$\mathbf{X}_n = \begin{bmatrix} \mathbf{x}_1 - \mathbf{x}_2, & \mathbf{x}_2 - \mathbf{x}_3, & \cdots, & \mathbf{x}_{N-1} - \mathbf{x}_N \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{n}}_1, & \hat{\mathbf{n}}_2, & \cdots, & \hat{\mathbf{n}}_{N-1} \end{bmatrix} \tag{11}$$

and $\mathbf{X}_n$ is in turn used to estimate the noise covariance Σ_n. The covariance matrix has shape $L \times L$ and can be estimated using cov (Matlab) or numpy.cov (Python). You will need to transpose the input in Matlab.

Implement the above noise estimation method. Run MNF and PCA on HICO_noisy, and compute the average spectral error over all pixels. Use 100, 50, 30, 10, 5, 3 and 1 components. Present the results in a table and comment on the difference between PCA and MNF.

(f) Discuss your results (5%)

Reflect on the following questions:

Q1: Under what conditions would PCA be optimal, i.e. what are the assumptions made with respect to noise in the data when using PCA?

Q2: What is the primary difference between MNF and PCA?

(g) How can we best use the subspace? (5%)

Multivariate analysis has many potential application areas, but a common trait is that the reduced dimensionality, or subspace, is used to explore underlying and latent information in the data. When compared to other encoding based methods for compression the subspace has some advantages and disadvantages, briefly explain some of these. What is virtual dimensionality and how is it connected to subspace identification? To revisit the previous task, can you use dimensionality reduction to say anything about chlorophyll?

4 Fun but definitely hard problems (5%)

These are a few fun problems to think about. They are current research topics in the literature. If you think you have an good way of approaching one of these topics, contact Sivert or Joe, because it could lead to more research opportunities. For your report, reflect on one or more of the questions below and write an answer if you have one.

(a) Deep learning

Although Deep Learning is one of the most popular analysis techniques, it is difficult to apply to hyperspectral data because of the datacubes' large dimensionality. Can you think of a way integrate hyperspectral imaging with deep learning?

(b) Multispectral-hyperspectral image fusion

The fusion of high-resolution multispectral images with low-spatial-resolution hyperspectral images has become a popular research topic. Can you think of an effective way to use dimensionality reduction in the image fusion process?

(c) Spatial-spectral methods

Recent developments in hyperspectral data analysis have shown that combining spatial and spectral information can lead to much better accuracy. See for example, " Hyperspectral Feature Extraction Using Total Variation Component Analysis" (2016). However, it is also complicated and slow. Can you think of a way to make spatial-spectral analysis faster?

(d) Locating methane emissions

One hyperspectral imaging satellite recently in the news is Claire, also known as GHGSat-D (GHG = GreenHouse Gases). Claire found a large methane leak in Turkmenistan. When the government of Turkmenistan was notified, they were able to fix the gas leak. Thus hyperspectral imaging has world-saving potential. See: https://physicsworld.com/a/from-methane-emissions-to-space-weather-satellite-based-observations-forge-ahead/

The problem for this project, though, is that the data produced by Claire is proprietary. Can you find publicly available multispectral or hyperspectral data that could facilitate geolocating industrial methane leaks? Can you find any leaks?