NASA Computational Case Study,
Hyperspectral Data Processing: Cryospheric Change Detection
Nargess Memarsadeghi and Thomas Doggett

Abstract

In this case study we learn how remotely-sensed hyperspectral data of NASA's Earth Observing-1 satellite are processed to detect features such as ice, water, and snow on Earth.

Keywords: Remote sensing, hyperspectral data processing, Earth Observing-1, classification;

1 Remote sensing via hyperspectral imaging

Remote sensing refers to collection of information about an object without being in physical contact with it [11]. This information can be gathered via satellites, cameras on airplanes, or sensors that are distributed over an area. Depending on the data acquisition method used, there are various ways of processing and interpreting the obtained information. Different matter absorb and emit light at different wavelengths of the *electromagnetic spectrum* (Figure 3). Our eyes are sensitive to the emitted light only in the narrow Visible range of this spectrum, indicated Figure 3. Measuring the emitted energy of an object at different wavelengths can often times uniquely identify various properties such as its composition, temperature, humidity level, etc. Studying the relationship between matter and its absorbed/emitted energy at different wavelengths is called *spectroscopy*. For example, jewelers use devices that distinguish diamonds from fake ones based by measuring the materials' radiated energy at particular wavelengths and matching it with the known spectral signature of diamond at that wavelength. Such a device is called a *spectrometer*, while a similar device for astronomy applications is often called a *spectroscope*. Similarly, one approach to remote sensing is imaging spectroscopy. Scientists can identify water, snow, different vegetation types, fire, sand, ... via images obtained at different wavelength ranges of the light spectrum, or via *hyperspectral* images.

Hyperspectral data processing is widely used for detection and identification of surface, topographical, and geological features in earth and planetary sciences [3]. In a hyperspectral image, each point in the spatial domain is represented as an n-dimensional pixel, where each dimension represents measurements made at a different range of the light spectrum. For hyperspectral imaging, n can be more than 200. For n values of 10 or less, the data is often referred to as a *multispectral* one [10]. Therefore, hyperspectral/multispectral sensors on board a satellite collect information about each region as a set of n images. Each image is called a *spectral band* or *channel*. The physical area that each pixel represents in the spatial domain is called the *spatial resolution*.

The spectrum range at which measurements of each spectral channel (or each dimension of a pixel) were taken, is called the *spectral resolution* of that pixel.

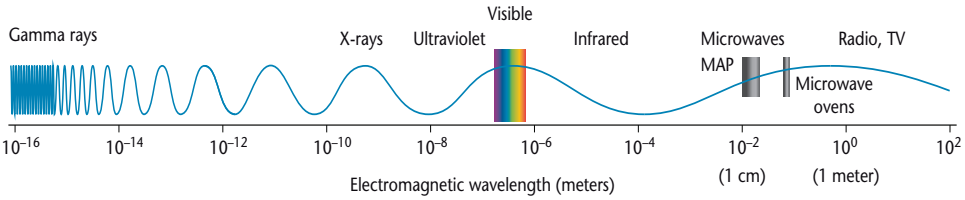

Figure 1: The wavelengths of electromagnetic radiation, displayed here in meters, are used to categorize different parts of the electromagnetic spectrum. Image credit: http://dawn.jpl.nasa.gov/DawnCommunity/flashbacks/fb_10.pdf

Earth Observing 1 (EO-1) is a NASA satellite that was launched on November 2, 2000 [1,3]. It has three instruments, one of which is Hyperion. Hyperion collects hyperspectral data from more than 220 bands with 30 meters spatial resolution and 10nm (1nm = 10^{-9}m) spectral resolution covering wavelengths from 0.4 to 2.5 μm (1 $\mu m = 10^{-6}$ meter). Each image recorded by the instrument covers a 7.5 km by 100 km land area. Scientists process Hyperion data for identifying and detecting various features on earth or its atmosphere. Some algorithms detect clouds, smoke, and different gases, and others identify fire, water, ice, snow, and different vegetation types.

2 Cryospheric change detection

In this case study, we learn about a *cryospheric* change detection algorithm by processing remotely-sensed hyperspectral data. The cryosphere is the component of the Earth that is composed of ice in the form of snow, floating ice, glaciers, and the soil at or below water's freezing point which dynamically interacts with atmosphere, climate, planet's crust and water cycle [2]. First, we process a subset of a Hyperion dataset that was obtained over Mapam Yumco in Tibet at 4:56 on March 27, 2004. The latitude and longitude for this dataset is 31° to the north and 81°34′ to the east.

Activity 1. Read Hyperion band number 8, 21, 31, 51, and 150 of the mentioned dataset from data files titled `b_8.tiff`, `b_21.tiff`, `b_31.tiff`, `b_51.tiff`, and `b_150.tiff` respectively. Report the size of images and their pixels' data type. Display and save a color image in a .jpg file using bands 31, 21, and 8 as the Red (R), Green (G), and Blue (B) channels respectively.
Hint 1: You can use a MATLAB routine called `imread`, or implement its equivalent, for reading .tiff images.

The first step in most sensor data processing algorithms, such as the one we will present here, is to calibrate the collected raw *Digital Numbers* (DNs) to represent correct sensor *radiance* values by compensating for the gain factor and biases of the instrument's radiation. That is DN values in band number i, DN_i, should be divided by a gain factor G_i, which is a constant for each channel/band, to obtain the corresponding radiance images L_i for that band (Eq. 1):

$$L_i = \frac{DN_i}{G_i}. \tag{1}$$

For Hyperion data, Visible and Near Infra Red (VNIR) bands, which are band numbers less than 70, have a gain factor of 40, while the gain factor for the remaining Short Wave Infra Red (SWIR) bands is 80 [1]. Then, the radiance values for some algorithms should be converted to *reflectance* values as described in Eq. 2. While radiance is a measure of the amount of light or radiant energy received by the sensor, reflectance measures the amount of energy emitted and reflected from the surface being sensed and depends on the properties of that surface and the geometric relationship between the surface and the Sun at the time of the imaging [9].

$$\rho_i = \left(\frac{\pi d^2}{\cos(\theta) E_{SUN_\lambda}}\right) L_i. \tag{2}$$

Earth-Sun distance d is in Astronomical Unit (AU) which is about 149,597,870.7 kilometers, the approximate mean of the Earth-Sun distance. Distance d is known and different for each day of the year [8].

E_{SUN_λ} is the incident solar flux (irradiance), the amount of electromagnetic energy from sun incident on a surface per unit area and per unit time in Watts/$(m^2 \times \mu m)$. It is a function of the wavelength λ at which the measurement is taken. Solar irradiance values for different Hyperion bands are provided in `hyp_irradiance.txt` file [12].

The *solar zenith angle* , θ, is the angle between a vertical line on the data location on earth and the line of sight to the sun. It can be obtained from the date and time of the data acquisition and knowing the location of the observed data on earth. This site [7] provides the solar elevation or altitude for a given location at a given time. Then, one can derive the solar zenith angle using the relationship that the sum of the solar zenith and solar elevation angles is 90 degrees. These angles are also sometimes provided by the instrument's telemetry.

Activity 2. Convert the provided radiance values to reflectance for the given dataset.

Hint 1: Use Equation 1 to first obtain radiance values by calibrating DNs, and

then use Equation 2 to derive reflectance values from radiance values.
Hint 2: What are the solar flux values that you are going to apply to images from each band?
Hint 3: What is the solar zenith angle for the given dataset?
Hint 4: What is the Earth-Sun distance for the given dataset? Use information provided in [8] and the data acquisition date.

There are often other pre-processing algorithms one needs to apply to images before being able to analyze them. For example, removing the noise resulting from the detector fluctuations and geo-coding (knowing the geographic coordinate of each pixel) are two common preprocessing algorithms, some of which maybe performed on-board the spacecraft. Also, to ensure that data represent surface reflectance, atmospheric effects such as vapor, cloud, and smoke are accounted for (atmospheric correction). For this case study, we are working with Level 0.5 Hyperion data which has only been partially processed (please see Figure 2 in [2]), as this is the product that can be readily produced with available on-board processing power. Also, we are not concerned with geo-coding the data or the cloud detection algorithm since the area is covered with small amounts of cloud (0-9 percent based on data's meta data).

Now we start applying the classification algorithm for identifying ice, snow, and water. This exercise is based on an algorithm that was developed for automatic detection of such features on board EO-1 [2]. This algorithm only requires processing 5 out of Hyperion's 242 bands, listed in Table 1, which are provided to you. The algorithm consists of a set of conditions that are evaluated for each pixel and decisions are made based on these evaluations. This algorithm is demonstrated in Figure 2 as well as Table 2. Since the algorithm was designed

Band	Type	Wavelength (μm)
8	Short Wavelength Visible	0.43
21	Mid-wavelength Visible	0.56
31	Red	0.66
51	Near Infrared	0.86
150	Short wave infrared	1.65

Table 1: Hyperion bands used by the Cryospheric change detection algorithm in this case study based on the algorithm in [2].

for use on-board the spacecraft, it relies only on some simple pixel level arithmetic. In Table 2, ρ_x represents reflectance values of the band at wavelength x, while L_x represents radiance values of the band at wavelength x. The algorithm also uses the Normalized Difference Snow Index (NDSI) values at various steps (Eq. 3).

$$NDSI = \frac{(\rho_{0.56} - \rho_{1.65})}{(\rho_{0.56} + \rho_{1.65})} \quad (3)$$

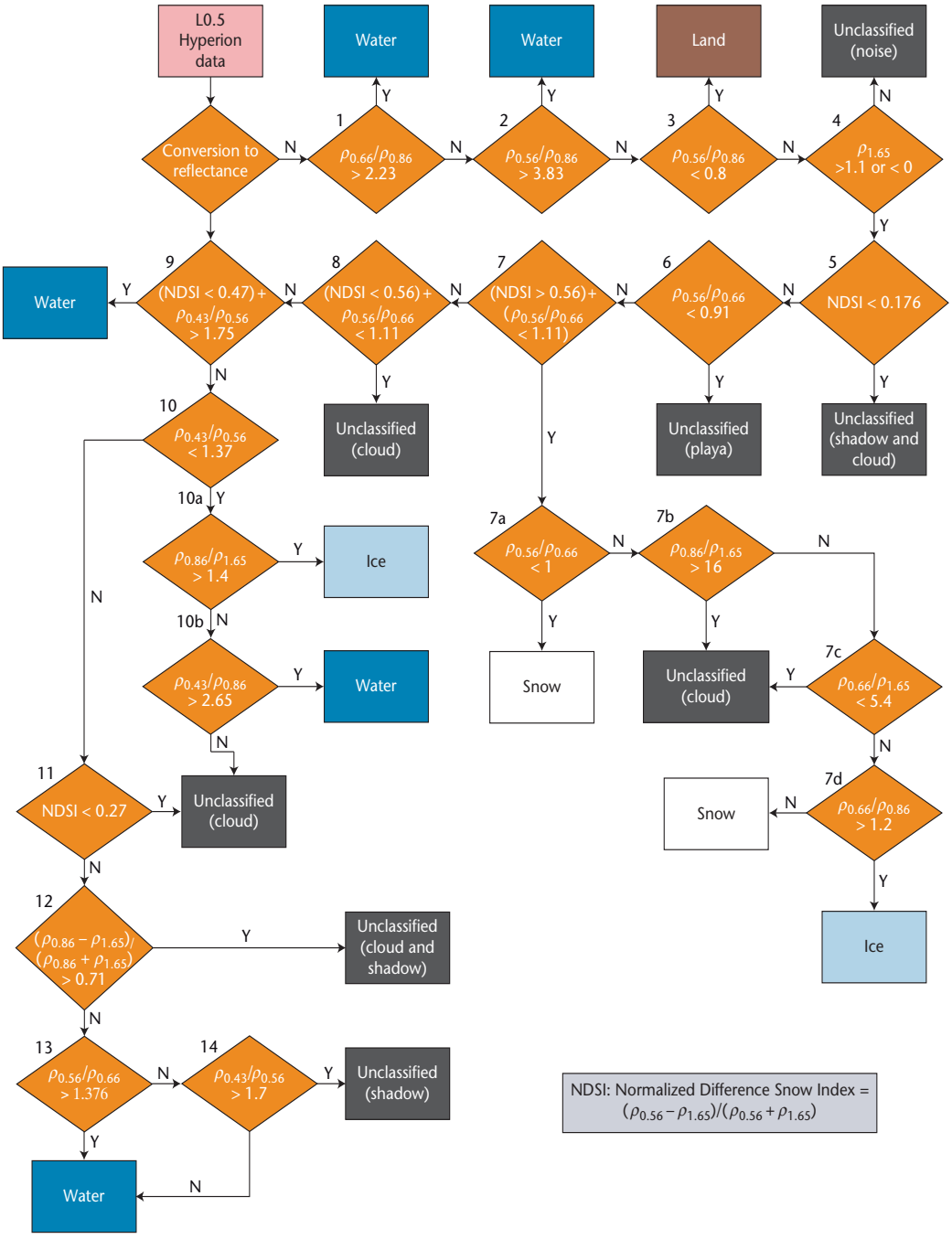

Figure 2: Flowchart of the cryospheric change detection algorithm, developed by Thomas Doggett and his colleagues [2], for identifying ice, snow, land and water given L0.5 Hyperion data. The raw data in five bands is first converted to reflectance values and those are used in an iterative series of empirically derived band ratios (see Table 2) to classify each pixel.

Step	Algorithm	Yes	No
1	$(\rho_{0.66}/\rho_{0.86}) > 2.23$	Water	step 2
2	$(\rho_{0.56}/\rho_{0.86}) > 3.83$	Water	step 3
3	$(\rho_{0.56}/\rho_{0.86}) < 0.8$	Land	step 4
4	$\rho_{1.65} > 1.1$ or $\rho_{1.65} < 0$	Unclassified	step 5
5	NDSI < 0.176	Unclassified	step 6
6	$(\rho_{0.56}/\rho_{0.66}) < 0.91$	Unclassified	step 7
7	(NDSI > 0.56) and $(\rho_{0.56}/\rho_{0.66}) < 1.11$	step 7a	step 8
7a	$(\rho_{0.56}/\rho_{0.66}) < 1$	Snow	step 7b
7b	$(\rho_{0.86}/\rho_{1.65}) > 16$	Unclassified	step 7c
7c	$(\rho_{0.66}/\rho_{1.65}) < 5.4$	Unclassified	step 7d
7d	$(\rho_{0.66}/\rho_{0.86}) > 1.2$	Ice	Snow
8	NDSI < 0.56 and $(\rho_{0.56}/\rho_{0.66}) < 1.064$	Unclassified	step 9
9	NDSI < 0.47 and $(\rho_{0.56}/\rho_{0.66}) > 1.76$	Water	step 10
10	$(\rho_{0.43}/\rho_{0.56}) < 1.37$	10a	step 11
10a	$(\rho_{0.86}/\rho_{1.65}) > 1.4$	Ice	step 10b
10b	$(\rho_{0.43}/\rho_{0.86}) > 2.65$	Water	step 11
11	NDSI < 0.27	Cloud	step 12
12	$(\rho_{0.86} - \rho_{1.65})/(\rho_{0.86} + \rho_{1.65}) > 0.71$	Unclassified	step 13
13	$(\rho_{0.56}/\rho_{0.66}) > 1.376$	Water	step 14
14	$(\rho_{0.43}/\rho_{0.56}) > 1.73$	Unclassified	Water

Table 2: The cryospheric change detection algorithm used in this case study based on the algorithm and Table 5 in [2].

Activity 3. Perform the cryospheric classification algorithm as described on the given dataset. Display and save the results in a .jpg file.
Hint 1: Use the suggested color coding in Table 3 for visualization of different classes.
Hint 2: What would be the numeric code for RGB channels for each color?

Figure 3: Scientist Thomas Doggett mapping snow and ice over on Lake Mendota, Wisconsin in January 2004 to ground truth the cryosphere classification algorithm.

So far we have performed a *classification* algorithm on hyperspectral data. That is, we not only have grouped points which are similar to each other into classes (clustering), we also know what each class represents and could label them (classification) as water, ice, etc. As you can see by now, classification algorithms rely on some knowledge of the data or properties of the classes. For example, the algorithm you developed for this case study was designed based on research results of earth scientists and their knowledge of reflectivity levels of different matter. If such an expert knowledge about objects and their reflectivity levels is absent, often times one can perform an unsupervised clustering algorithm on data, and then use *ground truth* points for each class to assign correct labels to them and turn the clustering result to a classification one. Ground truth represents prior knowledge of data and can help us make correct decisions

about other similar locations where such knowledge is lacking. For example, we know the location of seas and big rivers on a given image. If other points are grouped in the same cluster with a known sea on an aerial image, we can conclude that location contains water. Ground truth is also often used to verify and test the correctness of newly designed algorithms (e.g. Did we design the correct algorithm and there is really sand where it claims there is?). Often times scientists or students go to the field of study and collect ground truth information about vegetation types or other surface properties with a spectrometer and a hand-held Global Positioning System (GPS) device. Figure **??** shows scientist Thomas Doggett gathering collecting ground truth data for verification of his snow, water, ice, and land classification algorithm. Field data usually makes good school assignments and field trips for students.

Next, we explore the clustering approach. One of the widely used clustering algorithms, where k is the number of desired classes, is the k-means algorithm [4]. This algorithm aims to group points that are close to each other according to a similarity function so that sum of squared distances from all points to their cluster center is minimized. This is how this algorithm achieves this objective:

1. Select k random data points as initial cluster centers.
2. Calculate distances of all data points from these k cluster centers.
3. Assign each point to its closest cluster center.
4. Calculate the mean of all points in each cluster, and assign it as the new cluster center.
5. Repeat steps 2-4 till the algorithm converges, that is when the cluster centers do not change anymore.

Class	Color
Land	Brown
Water	Blue
Ice	Turquoise
Snow	White
Unclassified	Black

Table 3: Suggested color codes for visualizing the classification results.

Question 1. Is this algorithm guaranteed to converge? Why or why not?
Hint 1: What is the objective function that is being minimized?
Hint 2: Does this function have a global minimum? If yes, calculate it; if no, why not?
Note: To learn more about this algorithm, you can first go through another case study on this topic [5, 6].

Activity 4. Apply the k-means algorithm for clustering of the data into 5 classes. Display the results by using your knowledge of the data from previous Activities.
Hint 1: What is the dimensionality of each point for this dataset?
Hint 2: What is the value k, number of desired classes? (see Table 3).
Hint 3: Use result of the cryospheric algorithm as your ground truth to label each class.

Question 2. What are the advantages and disadvantages of each approach?
Hint 1: How do the two approaches compare in terms of the number of required calculations per pixel?
Hint 2: How do the two approaches compare in terms of the running time?
Hint 3: How do the two approaches compare in terms of the required memory?
Hint 4: How does each approach rely on the user's knowledge of the data?
Hint 5: How does the two algorithms compare in terms of the quality of results? Do they group the same set of points as a cluster?
Hint 6: Can you classify each pixel independently of others in each approach?

In this case study we performed cryospheric change detection algorithm on a Hyperion dataset of NASA's EO-1 satellite. In fact, the same processes your learned about here currently run on board of the EO-1 spacecraft! Then, products similar to what you have generated are downlinked to Earth.

Nargess Memarsadeghi is a computer engineer at NASA Goddard Space Flight Center, where she processes large scientific datasets with applications in earth sciences, planetary sciences, and astrophysics. She also leads the Educational NASA Computational and Scientific Studies (enCOMPASS) project. Her research interests include image processing, optimization, and data fusion algorithms. She has a PhD in computer science from the University of Maryland at College Park. Contact her at Nargess.Memarsadeghi@nasa.gov or visit `http://encompass.gsfc.nasa.gov`.

Thomas Doggett is an adjunct instructor at Northern Virginia Community College, where he teaches Physical and Historical Geology. His research interests include the geology of icy worlds in the outer solar system and remote sensing of the terrestrial cryosphere. He has an MS in Geological Sciences

from the Arizona State University and is a member of the Autonomous Sciencecraft Experiment science team. Contact him at tdoggett@nvcc.edu or visit http://ase.jpl.nasa.gov.

Acknowledgements

We thank the Autonomous Sciencecraft Experiment (ASE: `http://ase.jpl.nasa.gov`) team of the EO-1 spacecraft for the datasets used in this case study and their help and guidance on the algorithm. EO-1 mission, part of NASA's New Millennium Program (NMP), is operated by NASA's Goddard Space Flight Center.

References

[1] Pamela Barry. *EO-1/Hyperion science data user's guide, level 1-B.* TRW Space, Defense, and Information Systems., Redondo Beach, CA, May 2001.

[2] T. Doggett, R. Greeley, S. Chien, R. Castano, B. Cichy, A. G. Davies, G. Rabideau, R. Sherwood, D. Tran, V. Baker, J. Dohm, and F. Ip. Autonomous detection of cryospheric change with hyperion on-board earth observing-1. *Remote Sensing of Environment*, 101(4):447–462, April 2006.

[3] Michael K. Griffin, Su M. Hsu, Hsiao hua K. Burke, Seth M. Orloff, and Carolyn A. Upham. Examples of EO-1 Hyperion data analysis. *Lincoln Laboratory Journal*, 15(2):271–298, November 2005.

[4] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data.* Prentice Hall, Englewood Cliffs, NJ, 1988.

[5] Nargess Memarsadeghi and Dianne P. O'Leary. Classified information: The data clustering problem. *Computing in Science and Engineering*, 5(5):54–57, September/October 2003. Solutions are in 5(6): 64–70.

[6] Nargess Memarsadeghi and Dianne P. O'Leary. *Scientific Computing with Case Studies*, chapter 11, Classified Information: The Data Clustering Problem, pages 149–155. Society for Industrial and Applied Mathematics (SIAM), 2009.

[7] Naval Meteorology and Oceanography Command. Naval oceanography portal: Sun or moon altitude/azimuth table: U.s. cities and towns. `http://www.usno.navy.mil/USNO/astronomical-applications/data-services/alt-az-us`. Accessed February 2, 2012.

[8] NASA. Landsat 7, science data users handbook, chapter 11. `http://landsathandbook.gsfc.nasa.gov/pdfs/Landsat7_Handbook.pdf`. Accessed February 2, 2012.

[9] Sr. Nicholas M. Short. Remote sensing tutorial. `http://rst.gsfc.nasa.gov`. Accessed February 2, 2012.

[10] John A. Richards and Xiuping Jia. *Remote Sensing Digital Image Analysis: An Introduction.* Springer, fourth edition, December 2005.

[11] Floyd F. Sabins. *Remote Sensing Principles and Interpretation.* W.H. Freeman and Company, San Francisco, third edition, October 1996.

[12] United States Geological Survey. USGS, EO-1 Website, Frequently Aked Questions (FAQs), Question 21, Table 3. `http://edcsns17.cr.usgs.gov/eo1/faq`. Accessed February 2, 2012.