Practical Environmental Measurement Techniques / F-Praktikum Physik:

Satellite Image Analysis

Last change of document: April 17, 2024
Supervisor:
Christian Melsheimer, room N3370, tel. 62181
Marcus Huntemann, room N3250, tel. 62182

1 Background of the experiment

The goal of this lab exercise is to give a first idea of the usage of satellite images taken in the visible and infrared part of the electromagnetic spectrum. Today, a variety of satellites is in operation providing such images at different spatial resolutions ranging between several kilometres and below 1 m. In this lab exercise you will learn how different surface types can be identified and distinguished in satellite images.

We will analyse an image of Landsat 8, which was launched on February 11, 2013. It has a near-polar and sun-synchronous orbit at an altitude of 705 km with an equator crossing time at 10:11 a.m. local time for the descending node (this is the equator crossing time *by definition*). One orbit takes 98.9 minutes.

The image was taken by the Operational Land Imager (OLI) on Landsat 8, which has 9 spectral channels. Their properties are summarised in the following table:

ch.	colour	wavelength	res.	typical application
1	Ultra Blue	0.435 - 0.451 μm	30 m	coastal / aerosol
2	Blue	0.452 - 0.512 μm	30 m	
3	Green	0.533 - 0.590 μm	30 m	green reflectance of heavy vegetation
4	Red	0.636 - 0.673 μm	30 m	chlorophyll absorption of plants
5	NIR	0.851 - 0.879 μm	30 m	biomass surveys and water bodies
6	SWIR 1	1.566 - 1.651 μm	30 m	
7	SWIR 2	2.107 - 2.294 μm	30 m	
8	Panchromatic	0.503 - 0.676 μm	15 m	
9	Cirrus	1.363 - 1.384 μm	30 m	Clouds

Although all the channels of the image are provided by OLI in a 12 bit quantisation, an 16 bit quantisation is used for the image presentation, i.e., the values of the pixels vary between 0 and 65535.

The data are taken sequentially by the sensor on the satellite and represent the intensity of the reflected (or scattered) radiation (called radiance) of an area of the earth surface. The data are stored digitally line by line and each line is subdivided into area elements called pixels. Thus, a digital image is stored as a rectangular arrangement of numbers. In mathematics such a rectangular arrangement of numbers is called a matrix. Henceforth, we will think of an image as an image matrix.

The matrix elements here (i.e. Landsat OLI) range from 0 to 65535. They are stored in 2 byte, i.e., 16 bit. Low values are typically represented as dark tones, high values as bright tones. They are often shown in a black and white color scale, with different values as different grey levels.

For the representation of a so-called false-colour image, the red, green, and blue (r,g,b) information is needed for each pixel, i.e., three values. The false colour is composed from this information. Therefore, a false-colour image needs three times the storage of a single channel image image. There are three ways of representing false-colour images in an array: channel-interleaved, line-interleaved and pixel-interleaved. Using the *numpy* package of the programming languge *Python*, the representations would be

Type	Declaration
channel interleaved	`image = np.array(c,l,3)`
line interleaved	`image = np.array(c,3,l)`
pixel interleaved	`image = np.array(3,c,l)`

where *l* and *c* represent the numbers of lines and columns, respectively. Here, we will use the channel-interleaved representation.
Remark: The term "false-colour image" means that the three "basic" colours red, green and blue of a representation (e.g., the display on the screen) are attached to three information channels (sensor channels) of a multichannel image. Obviously such an image does not necessarily appear in colours as seen normally by eye ("natural" colours). An image appears in "natural" colours only if the three colours of the representation are attached correctly to the red, green, and blue channels of a sensor. This is a so-called "true colour image". Unfortunately, the term "false-colour image" is sometimes also for a single channel image where arbitrarily chosen colours are attached to the grey values. The correct term for this should be "indexed coloours", "palette colours" or "mapped colours". So before using this term "false-colour image" it has to be specified clearly in which context it will be used. In this lab exercise all false colour images consist of three information channels. While Landsat OLI has a blue channel, a number of satellite instruments do not have a blue channel, so they cannot produce true colour images.

1.1 Recommended reading

- Any textbook on satellite remote sensing and digital image processing (specifically: classification), e.g., R. A. Schowengerdt, Remote Sensing, Models and methods for image processing, Academic Press 1997 (chapter 1 and 9.1–9.3).
- https://www.nrcan.gc.ca/maps-tools-publications/satellite-imagery-air-photos/tutorial-fundamentals-remote-sensing/9309
- http://www.esa.int/SPECIALS/Eduspace_EN/ – choose "Remote Sensing Principles" from the sidebar on the left.
- Some details about the instrument OLI on Landsat-8 can be found here: https://www.usgs.gov/media/files/landsat-8-data-users-handbook – the Landsat 8 User Handbook (PDF), look at Section 2

See also the tasks for preparation at the end of section 4

2 Experimental setup

While initially specific hardware was required for image processing, nowadays most PCs are suitable for this task. An extended memory and a large screen are recommended. The images are generally provided on CD, DVD, or through the Internet. Here, the image data are already on the hard disk or can be easily downloaded.

3 Experimental procedure and tasks

3.1 Experiments

The experimental work, i.e., the analysis of a satellite image, is done using the programming language *Python*. More specifically the web-browser frontend *Jupyter Lab* is used, with a kind of worksheet that contains the required Python commands ("code") and the necessary documentation. The code sections can be modified.
The main tasks for the analysis of the image are shown below – detailed instructions for their execution are in the Jupyter Notebook file.

1. Read the complete image into an array `img`. Please keep in mind that channel 8 is not included in the loaded dataset.

2. Crop the image to the city of Hamburg and surroundings using the following steps with an example channel

   (a) Display a histogram

   (b) Decide for a reasonable upper and lower limit for image data of this channel from the histogram

   (c) Display the channel data in a grey-scale image with the chosen upper and lower limits

   (d) Determine the minimum and maximum x and y coordinates for the crop around Hamburg

3. With the cropped image, for each channel image, i.e., `img[:,:,`*n*`];` *n*`=0,1,2,...`, do the following:

   (a) Display the channel image. Is the range of intensities well used?

   (b) Display the histogram of the channel image. Select upper and lower limits for a rescaling of the intensities.

   (c) Calculate a contrast enhanced channel image, store it in an array with the name `imgScl`.

4. Generate a false-colour image by attaching the colours red, green and blue of the additive colour representation normally used in computers to three of the four channels. Select the channels in a way that a maximum of details becomes visible. Hint: By convention, pixels with much vegetation should be displayed in red. Try the following combination:

channel	colour
5 (NIR)	R
4 (red)	G
3 (green)	B

Display and the resulting image.

5. Identify the geographic region of the image on a map (e.g., https://www.openstreetmap.org. Mark the displayed region in the map. What are the dimensions of the displayed region? Determine the dimensions of one pixel from the map. Does it agree with what you know from the data?

6. Which surface types can you recognise visually (6 types, including clouds and cloud shadows, if present)?

7. Do a pixel-based supervised classification of surface types, based on the red and NIR pixel values in the following way:

   (a) Select rectangular blocks containing mixed as well as possibly only pure surface types in the image. Plot the intensity values of these blocks in the red-NIR-plane. This type of representation is called scatter plot.

   (b) Surface types characterisation from scatterplot: For each of the identified surface types, determine the ranges of intensities occupied in the red and NIR channels from such scatterplots (this means you define a rectangle in the Red-NIR plane that encloses the most dense part of the scatter plot). Record/write down the lower and upper bound in the red and NIR channel for each surface type.

   (c) Display all pixels with red and NIR values within these bounds

   (d) Try to refine the lower and upper bounds of the intensities.

8. Do an unsupervised classification

3.2 Data analysis

Discuss the results of tasks 4, 5, 6, and 7: Explain the visual appearance of the different types (e.g., why does the water look dark). Prepare a list of lower and upper bounds separating the surface types. Do that for the bounds before and after refining. Sample form of the matrix of bounds:

	surface type 1		surface type 2		surface type 3		…
ch. 3 (red)	low	high	low	high	low	high	…
ch. 4 (NIR)	low	high	low	high	low	high	…

Also represent that matrix graphically (the lower and upper bounds in the red and NIR channel for each type define a rectangle in the red-NIR plane).
Discuss the matrix of the refined bounds and its graphical representation: Which surface types can be easily separated, which ones cannot? Explain why.
Compare the result of the unsupervised classification with the result of the supervised classification.

3.3 Special precautions

As you will be working in a research environment, do not modify the computers, and do not pull any plugs! In case of any problems, ask the scientists around. Do not drink or eat when working on the computers!

4 Questions and Tasks for preparation

1. Satellite:
   - (a) What is a sun-synchronous orbit?
   - (b) What does an ‘equator crossing time of 10:11 a.m.’ mean?
   - (c) For which node is the equator crossing time defined?
2. Sensor:
   - (a) What is the spatial resolution of a remote sensing sensor?
   - (b) What is the difference between the spatial resolution of a sensor and the pixel size of an image taken by the sensor?
   - (c) What is the spectral resolution of a sensor?
3. Image:
   - (a) What is the histogram of an image?
   - (b) Why do the pixels of an 16 bit quantisation range between 0 and 65535?
   - (c) What is the range of a 12 bit quantisation?
   - (d) What happens with the distribution of the grey values if a 12 bit quantisation image is stretched into an 16 bit quantisation? How does the new histogram look like? What happens in the opposite case (16 bit to 12 bit)?
   - (e) What happens with the distribution of the grey values if this 16 bit image (i.e. 12 bit changed to 16 bit) is averaged by 2 × 2 pixels (reduction of the images size by a factor of 2 in both directions)?
   - (f) What is a RGB image? Why do you have to convert from 16 to 8 bit for a RGB representation?
4. Physics:
   - (a) Which part of the electromagnetic spectrum is used by the OLI sensor?
   - (b) What is the source of this radiation? Which physical process(es) allow to image the earth surface from space in this spectral range?
   - (c) Although this sensor has one, why do many of the spaceborne sensors do not have a blue channel?
5. Analysis & Data:

(a) Landsat data is freely available. Use the Earthexplorer under https://earthexplorer.usgs.gov/ to get a feeling for the data available. Assuming we want to analyse the city of Hamburg, which aspects should be taken into account when selecting the image?

(b) Look at http://scikit-learn.org/stable/modules/clustering.html for an overview of automatic clustering algorithms. Have a look at sections 2.3.1 and 2.3.2. Which algorithms might be useful for a unsupervised analysis of satellite images. Assume an image size of approx. 2000 by 2000 pixels and no more than 10 different surface types.

(c) The KMeans algorithm is a very simple clustering algorithm. For an interactive example see https://www.naftaliharris.com/blog/visualizing-k-means-clustering/. Try the algorithm on the default set with different starting positions for the centroids and with different point sets. What are possible drawbacks of this algorithm and how are they avoided?

(d) What is the correlation coefficient?