

Satellite Image Analysis

Last change of document: March 12, 2013

Supervisor: Christian Melsheimer, room N3371, tel. 62181, Georg Heygster, room N3330, tel. 62180

Lab: room N3280

Accompanying material to this document:

- Map of the City of Hamburg and its vicinity
- Basic Literature

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 the Indian Remote Sensing Satellite IRS-1C, which was launched on December 28, 1995. It has a quasi-polar and sun-synchronous orbit at an altitude of 817 km with an equator crossing time at 10:30 a.m. local time for the descending node (this is the equator crossing time *by definition*). One orbit takes 101 minutes.

The image was taken by the sensor LISS-III (Linear Imaging Self Scanning Sensor III) of the IRS-1C which has 4 spectral channels. Their properties are summarised in the following table:

ch.	colour	wavelength	res.	typical application
2	green	0.52-0.59 μm	23 m	green reflectance of heavy vegetation
3	red	0.62-0.68 μm	23 m	chlorophyll absorption of plants
4	NIR	0.77-0.86 μm	23 m	biomass surveys and water bodies
5	SWIR	1.55-1.75 μm	70 m	

Although all the channels of the image are provided by the LISS-III in a 7 bit quantisation, an 8 bit quantisation is used for the image presentation, i.e., the values of the pixels vary between 0 and 255.

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 is 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 typically range from 0 to 255. They are stored in 1 byte, i.e., 8 bit. Low values are typically represented as dark tones, high values as bright tones. This type of image is called halftone image. They are typically represented in black and white.

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 half-tone values. The false colour is composed from this information. Therefore, a false-colour image needs three times the storage of a half-tone image. There are three ways of representing false-colour images in an array: channel interleaved, line interleaved and pixel interleaved. The IDL representations would be

Type	Declaration
channel interleaved	<code>image = BYTARR (c, l, 3)</code>
line interleaved	<code>image = BYTARR (c, 3, l)</code>
pixel interleaved	<code>image = BYTARR (3, c, l)</code>

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 appear in colours as seen normally by eye (“natural” colours). An image appears in “natural” colours only in the case that 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 used also for a half-tone image where arbitrarily chosen colours are attached to the grey values. So before using this term 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. The production of true colour images is not possible because the blue channel is missing for the LISS-III sensor.

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).
- http://ccrs.nrcan.gc.ca/resource/tutor/fundam/index_e.php
- http://www.esa.int/SPECIALS/Eduspace_EN/ – choose “Remote Sensing Principles” from the sidebar on the left.

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, tape, or through the Internet. Here, the image data are already on the hard disk.

3 Experimental procedure and tasks

3.1 Experiments

The analysis of the image consists of the following tasks. You will find detailed instructions for their execution in chapter 5.

1. Read the complete image channel by channel from disk into an array `img(1300, 1300, 4)`. For each channel image, i.e., `img(*, *, n)`; $n=0, 1, 2, 3$, do the following:
 - (a) Reduce the size to 650×650 pixels by averaging each 2×2 pixels into one using the command `rebin`.
 - (b) Change the orientation by using the command `rotate`.
 - (c) Display the channel image. Is the range of intensities well used?
 - (d) Display the histogram of the channel image. Select upper and lower limits for a rescaling of the intensities.
 - (e) Calculate a contrast enhanced channel image, store it in an array with name and dimensions `imgsc1(650, 650, 4)`.
2. 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
4 (NIR)	R
3 (red)	G
2 (green)	B

Display and print out the resulting image.

3. Identify the geographic region of the image on the accompanying map. 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?
4. Which surface types can you recognise visually (6 types, including clouds and cloud shadows)?
5. Do a pixel-based 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. See chapter 5 for programming details.

- (b) Surface types in histograms: For each of the identified surface types, determine the ranges of intensities occupied in the red and NIR channels. Indicate lower and upper bound, each. See chapter 5 for programming details.
- (c) Try do refine the lower and upper bounds of the intensities. See chapter 5 for programming details.

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.

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 for preparation

1. Satellite:

- (a) What is a sun-synchronous orbit?
- (b) What does an 'equator crossing time of 10:30 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 8 bit quantisation range between 0 and 255?
- (c) What is the range of a 7 bit quantisation?
- (d) What happens with the distribution of the grey values if a 7 bit quantisation image is stretched into an 8 bit quantisation? How does the new histogram look like? What happens in the opposite case (8 bit to 7 bit)?
- (e) What happens with the distribution of the grey values if this 8 bit image (i.e. 7 bit changed to 8 bit) is averaged by 2×2 pixels (reduction of the images size by a factor of 2 in both directions)?

4. Physics:

- (a) Which part of the electromagnetic spectrum is used by the LISS-III 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) Why do many of the spaceborne sensors not have a blue channel?

5 Detailed Instructions for the Lab Exercise: Satellite Image Analysis

The objective of this lab exercise is to display and analyse high-resolution multispectral satellite images acquired in the visible (VIS), near infrared (NIR), and short wave infrared (SWIR).

But before we start, some basic information.

The tasks according to the lab exercise description are:

- reading of the data from disk
- preparation of the image for analysis and display
- display of single image channels
- analysis of the grey value distribution (histogram) of each channel
- rescaling of image grey values for appropriate image representation using histograms
- generation of so-called false-colour images
- selection of image areas representing typical surface types and estimation of lower and upper bounds of the image values of these surface types in the red and near infrared channel
- refining of the bounds

5.1 Description of the data:

Satellite: IRS-1C Sensor: LISS-III

LISS-III channels:	Resolution:	Pixel size:	IDL array index
2 - green	23 m	25 m	0
3 - red	23 m	25 m	1
4 - near infrared	23 m	25 m	2
5 - short wave infrared	70 m	25 m	3

Research area: City of Hamburg and its vicinity

And now let's start: Good luck and have fun!

But caution!!! Before typing any of the commands of the following instructions please read all instructions and comments above and below each section of commands very carefully!!!

You need to be logged in at one of the computers set up for that lab exercise. Your supervisor will do that for you. The IDL program should be started from within a terminal. This will usually also be done by your supervisor.

5.2 Tasks 1a+b

Read the image data from disc by using the following commands:

```
infile='~/FP13/data/hamburgcity_xdr.smp'
```

```

nxpos=1
nypos=1
nsz=1
openr,1,infile,/xdr
readu,1,nxpos,nypos,nsz
print,nxpos,nypos,nsz
img=bytarr(nsz,nsz,4)
readu,1,img
close,1

```

`nsz` is the original size of the image (number of pixels in both directions). Find out this number by typing the command `print,nsz`.

Now the 4 channels of the image are stored in a 4-dimensional array variable named `img`. In this form the data are not well suited for further analysis and display purposes. Reduce the image size by a factor of 2 in both dimensions and change its orientation using the following commands:

```

img=rebin(img,nsz/2,nsz/2,4)
for n=0, 3 do img(*,*,n) = rotate(img(*,*,n),7)

```

The expression `nsz/2` calculates the new size of the image. Find out this number by typing the command `print,nsz/2`.

5.3 Task 1c

Open a display window by typing (you need to do this only once!):

```

window,0,xsize=nsz/2,ysize=nsz/2,retain=2

```

Display each channel of the multispectral image on the screen by using the following command:

```

tvsc1,img(*,*,n)

```

The variable `n` has to be replaced by the number of the channel you want to display. The numbering is in IDL convention (IDL array index, see above)!!!

5.4 Task 1d+e

Use the following commands to calculate and display histograms of each channel and choose appropriate lower and upper limits:

```

hst0=histogram(img(*,*,n),min=0,max=255)
window,2,xsize=550,ysize=300,retain=2
plot,hst0

```

As above, the variable `n` has to be replaced by the number of the channel you want to analyse. This procedure will occur more often if you proceed in this lab exercise. (Numbering always in IDL-convention, i.e., IDL array index!!!)

Display each rescaled channel by using:

```
window, 0, xsize=nsz/2, ysize=nsz/2, retain=2
tv, bytscl(img(*, *, n), lower limit, upper limit)
```

(where n has to be replaced by the number of the channel you are working on, and *lower limit* and *upper limit* have to be replaced, of course, by the respective numbers chosen when examining the histogram) and store the rescaled channel in a new array variable named `imgsc1`. To create this array variable type `imgsc1=bytarr(nsz/2, nsz/2, 4)`. You have to do this only once! The storage is as follows:

```
imgsc1(*, *, n) = bytscl(img(*, *, n), lower limit, upper limit)
```

which you have to do for all 4 channels.

Note: If you use several display windows in IDL (just like now: window 0 for the image, and window 2 for the histogram), only one window at a time is active. All `plot`, `tv` and `tvsc1` commands will put the plot/image into the active window. By default, the last window you opened (with command `window, n, xsize=...`) is the active one.

To switch to another window, type

```
wset, m
```

where m must of course be replaced by the window number.

5.5 Task 2

Prepare the display on the screen by typing the following commands:

```
tv!ct, indgen(256), indgen(256), indgen(256)
window, 0, xsize=nsz/2, ysize=nsz/2, retain=2
```

False-colour images consist of an arbitrary combination of 3 channels of the LISS-III sensor. Create a new array variable called "imgcl" by typing:

```
imgcl=bytarr(nsz/2, nsz/2, 3)
```

This command creates an array variable. Do that only once!

Store 3 of the appropriately rescaled channels from "imgsc1" in `imgcl` by typing:

```
imgcl(*, *, 0) = imgsc1(*, *, n1)    (Colour Red in display)
imgcl(*, *, 1) = imgsc1(*, *, n2)    (Colour Green in display)
imgcl(*, *, 2) = imgsc1(*, *, n3)    (Colour Blue in display)
```

The variables $n1$, $n2$, and $n3$ have to be replaced by a selection of channel numbers (IDL array indices). This is similar to the treatment of the variable n mentioned above. Use the following command to display the coloured image:

```
tv, imgcl, true=3
```

Repeat this procedure combining different channels of the LISS-III sensor until you find a satisfying combination. Before you proceed, ask your supervisor if this is an appropriate choice for the further analysis. Then print out the image with your channel combination typing the command:

```
imageprint, imgcl
```

You need the printout for the handling of task 3. You can use the printout or even better the display on the screen for task 4. The map is a helpful additional data source for the interpretation.

5.6 Tasks 5+6+7

For the handling of task 5 and 6 you can use the program `select_area` by typing:

```
select_area, img, imgscl
```

This program supports the selection of rectangular blocks from the screen by clicking in the display window and provides the according scatter plot of the red and near infrared channels of the LISS-III sensor. The program allows to repeat this procedure on your demand.

For refining or adjusting the lower and upper bounds of the surface types (task 7) use the program `select_values` by typing:

```
select_values, img, imgscl
```

You can enter the boundaries of the surface types for both channels. The program marks the pixels corresponding to your choice of bounds in the image displayed on the screen. You can change these boundaries and check the result. Additionally, the program provides two histogram plots of the red and near infrared channel with an indication of your actual choice of boundaries. Within this program, you can repeat the procedure on your demand.

5.7 Some useful IDL commands in case of problems

Sometimes IDL encounters problems in a subroutine such as `imageprint`, `select_area`, or `select_values`. This happens when you give wrong input, e.g., when you type letters instead of numbers in `select_values`, or when you have called a subroutine without or with wrong arguments, e.g., `select_area, img` instead of `select_area, img, imgscl`. IDL will then abort the subroutine but still remain inside it. The consequence is that it will know only the local variables of that subroutine, while most other variables from the main level (`img`, `imgscl` etc.) are not accessible. If you continue, you will get error messages like “Variable is undefined: IMG”.

In order to get out of such a failed subroutine and return to the main level, type

```
retall
```

and all variables will be there again.

To check in which level you are and what variables are defined, you can always type

```
help
```

It will also tell you the dimensions of arrays.

Hope everything worked well!