Exploit Pleiades PHR data with the ORFEO ToolBox

Manuel Grizonnet (CNES), Julien Michel (CNES), Jordi Inglada (CNES)

modified by Mickaël Savinaud (CS-SI)

17/06/2013

Draft version: could be modified if Spot4-Take5 are available for LTC

Table of Contents

1. Foreword

1.1. About the data

The images used during these exercise are extracts from Pleiades demonstration products, made available for evaluation purpose. To get the full products please refer to this website. Products used are:

- `Pleiades Pan-sharpened ORTHO Compression REGULAR`
- `Pleiades TRISTEREO Bundle PRIMARY`

They are covered by a cnes copyright.

Other data needed for some exercises, as well as solution scripts can be found in the data package.

1.2. About the software

To perform the exercises, you will need to have the following software installed:

- **Orfeo ToolBox** 3.14 or later, including applications
- **Monteverdi** 1.6 or later
- **QGis** 1.8 or later

For **Orfeo ToolBox** and **Monteverdi** installation, you can refer to the installation from the Orfeo ToolBox Cookbook.

For **QGis** installation, please refer to **QGis** documentation, which can be found on the project website.

All these elements are available into the dedicated OTB virtual machine.

2. Exercises

2.1. Monteverdi ***and*** *OTB applications*

2.1.1. Description

2.1.1.1. Abstract

This exercise will get you familiar with the use of **Monteverdi** and **OTB applications**.

2.1.1.2. Pre-requisites

- Basic knowledge of remote sensing and image processing,
- Basic knowledge of command-line invocation.

2.1.1.3. Achievements

- Visualize data in **Monteverdi**,
- Basic processing in **Monteverdi**,
- Basic processing with **OTB applications** in graphical mode,
- Basic processing with **OTB applications** in command-line mode.

2.1.2. Steps

2.1.2.1. Monteverdi: data opening and visualisation

In this part of the exercise, you will use the following data: `phr_pxs_melbourne.tif`

1. Run **Monteverdi**: open a terminal and run the following command:

   ```
   $ monteverdi
   ```
2. Open the image (use the *File/Open dataset* menu)
3. Find how to display the image (there are two ways of doing it).
4. Navigate into the image:
   1. Change the full resolution displayed area
   2. Change the zoom displayed area,
   3. Change the zoom level,
   4. What are the information displayed about the current pixel under mouse pointer ?

5. Using the viewer control panel, in the *Setup* tab:
   1. Change set-up to visualize the 4th band,
   2. Change set-up to visualize in false color mode (screen red: near infra-red channel, screen green: red channel, screen blue: green channel).
   3. Change set-up to come back to natural colors
   4. Enhance contrast with respect to the full area,
   5. Enhance contrast with respect to the zoom area,
   6. Come back to default contrast enhancement.

Tips and Recommandations:

- You can use keyboard arrows to navigate into images as well,
- Pleiades bands order is red channel, green channel, blue channel, near infra-red channel.

2.1.2.2. Monteverdi: basic processing

In this part of the exercise, you will use the following data:

`phr_xs_melbourne.tif`

1. Open the image in **Monteverdi**.
2. Find the *BandMath* module in the menu. Open the image in this module. What kind of processing is offered ?
3. Using this module, compute the NDVI of the image: NDVI = \frac{NIR-RED}{NIR+RED}

Visualize the input image and the mask in the same viewer.

1. Using this module, build a mask of pixels whose Digital Number (DN) in the NIR channel is lower than 150. Visualize the input image and the mask in the same viewer.
2. Using this module, build a mask of pixels whose DN is upper than 1000 in all spectral bands. Visualize the input image and the mask in the same viewer.
3. Using the *Concatenate* module, build a composite RGB image with the mask of high values in the red channel, the mask of low NIR values in the blue channel and the NDVI in the green channel.
4. Using the *Color Mapping* Module, build a composite RGB image of the NDVI that allows for better image interpretation.

Tips and Recommandations:

- NDVI values are within -1 and 1, but the range can be much more narrow.

2.1.2.3. OTB applications: Graphical and command-line mode

1. Run the following command:

```
$ otbcli_OrthoRectification
```

   And then

```
$ otbgui_OrthoRectification
```

What do you observe ?

2. How many **OTB applications** are currently available ?
3. How can you get help and documentation about applications ?

2.1.2.4. OTB applications: Basic processing

1. Use the **OTB applications** to produce the same results as steps 3 to 7 as with **Monteverdi** in section 2.1.2.2.

2.1.2.5. Homework

1. How can we load or visualize images directly from command-line using **Monteverdi** ?
2. Is there another way to compute radiometric indices like NDVI with the **OTB Applications** ?
3. Learn about the *Python* access to **OTB Applications** and write a python script performing the same steps as in section 2.1.2.4.

2.2. Pre-processing : geometry and radiometry corrections

2.2.1. Description

2.2.1.1. Abstract

This exercise will get you familiar with the geometric and radiometric corrections using **OTB applications**.

2.2.1.2. Pre-requisites

- Basic knowledge of remote sensing and image processing,
- Basic knowledge of command-line invocation.

2.2.1.3. Achievements

- Orthorectification of remote sensing image using **OTB Applications**
- Optical calibration of remote sensing image using **OTB Applications**

2.2.2. Steps

In this part of the exercise, you will use the following data: `phr_xs_melbourne.tif`

2.2.2.1. Geometric corrections principle

This operation allows to go from image index to ground coordinates.

1. Run the command-line and graphical version of the **Orthorectification** application
2. Which parameters allow to manage Digital Elevation Model informations (DEM) in the application? Which modes are available?

3. Open the image with Monteverdi, what is the estimated spacing of the input image?

2.2.2.2. Orthorectification

1. In which UTM zone, is located Melbourne?
2. Use Quantum GIS to determine the corresponding EPSG code for this projection.
3. Perform the orthorectification of the image and open it in Quantum GIS

Tips and Recommandations:

- Use the estimated spacing read in **Monteverdi** to set the output spacing parameters in the **Orthorectification** application
- Use an average elevation of 20 meters

2.2.2.3. Optical calibration

This operation allows to go from Digital Number (DN) to reflectance (physical values)

It includes to convert Top Of Atmosphere to Top Of Canopy reflectance which aim is to compensate the atmospheric effects.

The optical calibration framework is OTB is fully adapted to the pipeline architecture.

1. Use the **OpticalCalibration** application to compute Top Of Atmosphere reflectance.
2. Use the **OpticalCalibration** application to compute Top Of Canopy reflectance.
3. Compare the two images using Monteverdi or OTB applications.

Tips and Recommandations:

- Enable the '-milli' parameter which allow to save the output image in uint16. By default, reflectance image is saved in float values (between 0 and 1)

2.3. Segmentation

2.3.1. Description

2.3.1.1. Abstract

This exercise will get you familiar with the OTB **Segmentation** application. You will learn how to produce a raster segmentation output with different algorithms and how to scale up to larger input images by producing vector outputs.

2.3.1.2. Pre-requisites

- Basic knowledge on OTB applications and QGis usage
- Basic knowledge on image segmentation
- Basic knowledge on GIS vector file formats

2.3.1.3. Achievements

- Usage of the OTB **Segmentation** application,
- Segmentation of large raster and import the results in a GIS software.

2.3.2. Steps

2.3.2.1. Getting familiar with the Segmentation *application*

1. Run the command-line and graphical version of the application
2. Read the documentation. What are the three segmentation methods available ?
3. What are the two output modes ?

2.3.2.2. Simple segmentation in raster mode

In this part of the exercise, you will use the following data:
`segmentation_small_xt_phr.tif`

1. Run the **Segmentation** application in *raster* mode, using the connected components filter and a thresholding condition on the spectral distance
2. View the resulting segmentation in **Monteverdi**. What do you see ?
3. Use the **ColorMapping** application to enhance the rendering of the result:
    1. Try the *optimal* method
    2. Try the *image* method
4. Try different connected components conditions and see how they influence the results. You can try to change the distance threshold for instance, or look into the documentation for other keywords.

Tips and Recommandations:

- Use the **distance** keyword in the expression to denote spectral distance
- Pay attention to the output image type

2.3.2.3. More segmentation algorithms

In this part of the exercise, you will use the following data:

`segmentation_small_xt_phr.tif`

1. Run the **Segmentation** application in *raster* mode again, but this time use the Mean-Shift filter. Use the **ColorMapping** application to visualize the results.
    1. Try the default parameters first
    2. Try to change the parameters and see how it influences the results. The most important parameters are the spatial and the range radius.
2. Run the **Segmentation** application in *raster* mode again, but this time use the Watershed

filter. Use the **ColorMapping** application to visualize the results.

1. Try the default parameters first
2. Try to change the parameters and see how it influences the results.

3. Compare the best results from the three algorithms. Keep the best segmentation result you had for Exercise 3.

Tips and Recommendations:

- There are two implementations of the Mean-Shift filter. Edison is the original implementation from the Mean-Shift paper authors.

2.3.2.4. Going big: the vector mode

In this part of the exercise, you will use the following data:

`segmentation_large_xt_phr.tif`

1. Run the **Segmentation** application in *raster* mode again, using the best parameters you had in previous section, on the large image. Look at computer resources. What happens ?
2. Run the **Segmentation** application again, this time in *vector* mode, and **disable the stitching option**. Look at computer resources. What happens ?
3. Open the result of the input image and the segmentation file in **QGis**. Tune **QGis** to allow for proper visualization (see Tips and Recommendation). What do you see ?
4. Run the **Segmentation** application again, this time in *vector* mode, and **enable the stitching mode**. Write the results to a different file and load it into the **QGis** project as well. What is the effect of the **stitch** option ?

Tips and Recommendations:

- Computer resources can be monitored by running `top` in another terminal
- Hit `Ctrl C` to interrupt the processing
- Use the *sqlite* file format to store vector outputs (`.sqlite` file extension)
- In **QGis**, one can import both raster and vector layers
- In **QGis**, one can tune raster layers rendering the following way:
  - Right-click on the layer, select *Properties*
  - Go to the *style* tab
  - Select *Use standard deviation*
  - In *Contrast enhancement*, select *Stretch to MinMax*
- In **QGis**, one can tune vector layers rendering the following way:
  - Right-click on the layer, select *Properties*
  - In the *style* tab, select *Change*

    - As *Symbol layer type*, select *Outline: Simple line*
    - You might change the color as well
- In **QGis**, you can save your project to a file and avoid having to reset those parameters

2.3.2.5. Homework

1. In *vector* mode, study the effect of the *tilesize*, *simplify* and *minsize* option.
2. Using the **Segmentation** application (and maybe other OTB applications), how can we segment everything but vegetation ?
3. Using the **Segmentation** application (and maybe other OTB applications), how can we deal with segmentation of high reflectance structures ?

2.4. Feature extraction

2.4.1. Description

2.4.1.1. Abstract

This workshop will introduce you to the **Feature Extraction** modules of **Monteverdi**. You will learn how to use these modules and visually evaluate the usefulness of different features for image information extraction.

2.4.1.2. Pre-requisites

- Basic knowledge of **Monteverdi** usage

2.4.1.3. Achievements

Being able to use the **Feature Extraction** modules of **Monteverdi**.

2.4.2. Steps

2.4.2.1. Getting around the *Feature Extraction* modules

Feature extraction is a generic term which refers to the procedure of processing images in order to produce either new images or a set of objects in order to represent the information contained by the processed image in a higher level of abstraction.

Other terms than *features* can be used as for instance *primitives*, *indices* or *descriptors*. Although minor differences in meaning exist among these terms, we will use the as synonims here.

In **Monteverdi** there are 5 different modules for feature extraction and they are all grouped under the *Filtering/$→$/Feature extraction* menu.

All *Feature extraction* modules follow the same rationale: they take an input image (with any number of channels) and produce an output image where each band is one of the computed features.

In this part of the exercise, we will use the following data:

`phr_xs_melbourne.tif`

1. Open the image in **Monteverdi**.
2. Open the **Smoothing** module from the *Filtering/$→$/Feature extraction* menu, and load the image and the segmentation inside the module.
3. What are the 3 image windows which appear at the top of the graphical user interface?
4. The *Action* tab is selected by default. What can you do on this part of the module?
5. Select the *Output* tab and describe what can you do on it.

*2.4.2.2. Smoothing **module***

You can either keep the *Smoothing* module open from the previous exercice or close it and open another one in order to start with a fresh one.

1. One of the available features is *Original data*. What is it and what may its purpose be?
2. Select the *Mean* feature and choose to compute it only on the *intensity* channel. What is the meaning of the *Radius along X* and *Radius along Y* parameters and what is their effect on the computed feature?
3. Select the *Meanshift smooth* feature. And try to guess the meaning of the parameters and their influence on the results.
4. Select the *Meanshift clusters* feature. As you can see, the same parameters as for the *Meanshift smooth are available*. Which is the difference in terms of the computed feature?

*2.4.2.3. Edge extraction **module***

Open the *Edge extraction* module and give it the input image. In order to keep thing simple, you will only work with one channel. Also, the *Harris detector* – which an interest point detector – and the *Touzi* filter – which is to be used with SAR images – will be ignored.

By reading text displayed on the *Feature Information* window and the list of available parameters and their effect on the computed features, answer to the following questions.

1. What does the *Meanshift boundaries* feature provide?
2. How can a *Variance* filter detect edges?
3. What is the effect of the *sigma* value on the *Recursive gradient* filter?
4. In the *Sobel* edge detection, the result is an edge density computed after the application of a Sobel edge enhancing filtering (high pas filter). You can set 2 different thresholds. What is their effect on the result?

*2.4.2.4. Radiometric index extraction **module***

By radiometric index we understand a combination of spectral bands which enhances a particular type of material. There are indices for vegetation, soils, water, artificial surfaces (built-up), and so on. You can find detailed descriptions and bibliographic references about this subjects by visiting the Radiometric indices page on the Orfeo Toolbox Wiki and following the links therein.

1. Open the *Radiometric index extraction* module and feed it with the 4-band Pléiades image.

Select the *NDVI* index under the *Vegetation* list of indices. Generate the feature by using different selections on the *Channels Selection* list. Why is the result always the same regardless of the chosen channels?

2. Which are the spectral bands most commonly used for vegetation indices?
3. Find a value of the parameter *s* for the MSAVI index which gives results close to the ones obtained by the RVI. What is the interest of indices with parameters as this one with respect to indices like RVI or NDVI for which there are no parameters?
4. Which water indices can't be computed on a Pléiades image?

*2.4.2.5. Texture extraction **module***

- SFSThe Structural Feature Set approach computes textures based on line direction analysis through the central pixel.

  Directions are computed at a constant step angle. A direction is defined as: $$ \mathit{d_{i} = \sqrt{(m^{e1}-m{e2})^{2}+(n^{e1}-n{e2})^{2}}} $$ From $\mathit{d_{i}}$, histograms are defined: $$ \mathit{H(c) : \{c \in I \mid \lbrack d_{1}(c), \ldots , d_{i}(c), \ldots , d_{D}(c)\rbrack \}} $$ Thus, 6 textures are defined : $$ \mathit{length = \max_{i \in \lbrack1; D\rbrack}(d_{i}(c)} $$ $$ \mathit{width = \min_{i \in \lbrack1; D\rbrack}(d_{i}(c)} $$ $$ \mathit{PSI = \frac{1}{D}\sum_{1=1}^{D}d_{i}(c)} $$ $$ \mathit{\omega-mean = \frac{1}{D}\sum_{1=1}^{D}\frac{\alpha.(k_{i}-1)}{st_{i}}d_{i}(c)} $$ $$ \mathit{ratio = \arctan{\frac{\sum_{j=1}^{n}{sort_{min}^{j}(H(c))}}{\sum_{j=1}^{n}{sort_{max}^{j}(H(c))}}}} $$ $$ \mathit{SD = \frac{1}{D-1}\sqrt{\sum_{1=1}^{D}(d_{i}(c)-PSI)^{2}}} $$

  1. Compute the $width$ and the $length$ features using the default parameters and explain what are the differences you observe on the results.
  2. Which is the effect of the spectral threshold on the computation of the features? Same question for the spatial threshold.

- HaralickThe Haralick textures are a set of indices computed from the grey-level co-occurrence matrices of an image. These are matrices computed on a grey-level image and for each pixel, a neighborhood is defined by a rectangular window which is shifted by an offset.

  Mathematically, a co-occurrence matrix C is defined over an $n x m$ image I, parameterized by an offset $(\Delta x,\Delta y)$, as:

  The image values are quantized using small number of bins so that the equality between pixels are likely to occur.

  There are 2 different implementations of the Haralick textures in Monteverdi. We will use the 1st one.

  3. Given the description above, what do you think is the meaning of the *radius*, *offset*, *min\/max* and *quant. levels* parameters?

4. Compute the *Entropy* of the *intensity* channel for different radiuses (2, 3, etc.). Which is the effect of this parameter?
5. Compute the *Energy*, *Entropy*, *Correlation* and *Inertia* textures with the same parameters. Compare them and see if some of them are redundant.

2.5. Learning and classification from pixels

2.5.1. Description

2.5.1.1. Abstract

This exercise will get you familiar with the OTB pixel based classification applications. You will learn how to train a SVM classification model from Pleiades images and a set of training regions. You will then learn how to apply this model to images and produce shiny classification maps.

2.5.1.2. Pre-requisites

- Basic knowledge on OTB applications and QGis usage
- Basic knowledge on image supervised classification
- Basic knowledge on GIS vector file formats

2.5.1.3. Achievements

- Usage of the OTB Classification applications
- Classification of large images
- Import of results in a GIS software

2.5.2. Steps

In this part of the exercise, you will use the following data:

`melbourne_ms_toa_ortho_extract_small.tif`

2.5.2.1. Produce and analyze learning samples

- Use Qgis to produce polygons for 5 classes (vegetation, roads, soil, buildings and water)
- Export this vector layer in shapefile
- What is the label corresponding to the class **water** in the shapefile? An example set of learning samples is provided for the exercise in *training.shp*

<u>Tips and Recommendations:</u>

- Note the field name of the shapefile which contains the label. You will need to provide this field in the training application

2.5.2.2. Estimate image statistics

In order to make these features comparable between each images, the first step is to estimate the input images statistics. These statistics will be used to center and reduce the intensities (mean of 0 and standard deviation of 1) of training samples from the vector data produced by the user.

- Use the **ComputeImagesStatistics** to compute statistics on the image
- What is the mean of the red band?
- The extract provided has been converted from DN to milli-reflectance. For what reasons, is it advised to do so when performing multiple images classification?

2.5.2.3. Estimate classification model using the Support Vector Machine algorithm

The **TrainSVMImagesClassifier** application performs SVM classifier training from multiple pairs of input images and training vector data. Samples are composed of pixel values in each band optionally centered and reduced using XML statistics file produced by the **ComputeImagesStatistics** application. We will use this application with only one image in this exercise.

- Use the **TrainSVMImagesClassifier** to produce SVM model
- Which kernel is used by default in the application?
- What is the measured accuracy?

2.5.2.4. Apply classification model

- Use the **ImageSVMClassifier** to apply the classification model to the input image
- What is the output of the application?
- Bonus : Use the same model to apply the classification to the other extract `melbourne_ms_toa_ortho_extract_large.tif`

2.5.2.5. Produce printable classification map

We are now going to produce a printable classification map using the **ColorMapping** application This tool will replace each label with an 8-bits RGB color specified in a mapping file. The mapping file should look like this :

```
$ # Lines beginning with a # are ignored
1 255 0 0
```

- Produce your custom look-up table (LUT)
- Use this LUT to produce a printable classification map (in PNG format)
- Overlay this map on the input image in QGIS. Comment on the classification results.

2.5.2.6. Homework

- Produce classification model with different kind of SVM kernels. Comment different accuracies obtained?
- Going big: Apply this classification on the pan-sharpened image over Melbourne

2.6. Learning and classification from objects

2.6.1. Description

2.6.1.1. Abstract

This workshop will introduce you to the **Object Labeling** module of **Monteverdi**. You will learn how to use the module and see the influence of different features on classification results. You will also experiment with a simple active learning implementation on objects.

2.6.1.2. Pre-requisites

- Basic knowledge of Object Based Image Analysis
- Basic knowledge on learning and classification

2.6.1.3. Achievements

Being able to use the **Object Labeling** module of **Monteverdi**.

2.6.2. Steps

2.6.2.1. The preliminary segmentation

In this part of the exercise, we will use the following data:

`phr_pxs_melbourne_xt_small.tif`

`phr_pxs_melbourne_xt_small_segmentation.tif`

1. Use the **ColorMapping** application to enhance the visualization of the segmented image (you can use the *optimal* and *image* modes as learned in the segmentation exercise).
2. Analyze the color-mapped segmentation results. For which kind of objects is the object based classification likely to work well ? For which kind of objects is it likely to perform badly?

2.6.2.2. Object Labeling *module - basics*

1. Open both the image and the segmentation image in **Monteverdi**.
2. Open the **Object Labeling** module from the *learning* menu, and load the image and the segmentation inside the module.
3. What is the purpose of each tab on the left side of the module?
4. In the *Objects* tab, create a new class. You can change its color and its name.
5. Right-click on an object of interest in the image. What happens?
6. Right-click a second time inside the selected object. What happens?
7. Add a few more objects to the current class.
8. Create a new class and add some objects to it.

9. Go to the *Features* tab, uncheck all features but the mean radiometric values.
10. Go to the *Learning* tab and click on classify. What happens?
11. Click on the *Save/Quit* button. What kind of outputs is produced by the module?

Tips and Recommendations:

- Choose two simple classes for this part of the exercise (for instance a *Water* class and a *Land* class)
- Use the navigation map to change the displayed area
- You can change the opacity of the classification layer as well as of the selected objects layer so as to better analyze the results.
- You can also clear the classification layer.

2.6.2.3. *Object Labeling* module - advanced

In this part of the exercise, we will use these additional files: `samples.xml` and `parameters.xml`

1. Load again the image and the segmentation inside the module.
2. Load the samples file using *File/Load Samples*. What are the different object classes loaded ? How many samples per classes are used ?
3. Uncheck all features except from radiometric means:
    - Band1::Mean
    - Band2::Mean
    - Band3::Mean
    - Band4::Mean
4. Perform the classification. What are the objects in the image that are badly classified because of missing classes ?
5. What are the objects in the image that are poorly classified because they are badly segmented or too complex ?
6. Try to enhance the classification by adding missing classes.
7. Try to enhance the classification by adding new features.

Tips and Recommendations:

- The **Object Labeling** module is quite memory consuming. Depending on the available memory on your system, you might want to restart **Monteverdi**.

2.6.2.4. *Object Labeling* module - active learning

1. In the *Objects* tab, click on the *Sample* button in the lower-left area. This will show you difficult samples by using the *margin sampling* technique.

2. What kind of segments are considered by the algorithm as hard to classify ?
3. Try to create a *Trash* class to handle noise segments.
4. Perform a few more iteration of active learning. What do you observe ?