Image Mining 2013 - OTB Training Session

OTB-Team

23 août 2013

Contents

0.1 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 Melbourne data product :

- `Pléiades Pan-sharpened ORTHO Compression REGULAR`
- `Pléiades TRISTEREO Bundle PRIMARY`
- `Pléiades Primary Product - Bundle`

They are covered by a cnes copyright.

Exercises based on these Pleiades data include a set of command-line to generate the needed extracts.

0.2 About the software

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

- **Orfeo ToolBox** 3.18 or later, including applications
- **Monteverdi** 2.0 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.

1 Exercises

1.1 Monteverdi2 and OTB applications

1.1.1 Description

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

Data If you need to generate the data used in this exercise from the original products, you can use the following command lines.

```
$ otbcli_ExtractROI -in ~/Demo_Product/ORTHO_PXS/IMG_PHR1A_PMS-N_001/IMG_PHR1A_PMS-N_201202250025599_ORT_PRG_FC_5855-001_R1C1.JP2
-out ~/Data/phr_pxs_melbourne.tif uint16 -startx 18432 -starty 4096 -sizex 4096 -sizey 4096
```

```
$ otbcli_ExtractROI -in ~/Demo_Product/ORTHO_XS/IMG_PHR1A_MS_002/IMG_PHR1A_MS_201202250025599_ORT_PRG_FC_5852-002_R1C1.JP2
        -out ~/Data/phr_xs_melbourne.tif uint16 -mode fit -mode.fit.ref ~/Data/phr_pxs_melbourne.tif
```

Pre-requisites

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

Achievements

- Visualize data in **Monteverdi2**,
- Basic processing in **Monteverdi2**,
- Basic processing with **OTB applications** in graphical mode,
- Basic processing with **OTB applications** in command-line mode.

1.1.2 Steps

Monteverdi2: data opening and visualisation In this part of the exercise, you will use the following data: `phr_pxs_melbourne.tif`

1. Run **Monteverdi2**: open a terminal and run the following command:

   ```
   $ monteverdi2
   ```

2. Open the image (use the Fichier/Importer image\ menu)
3. Navigate into the image:
   (a) Change the full resolution displayed area
   (b) Change the zoom displayed area,
   (c) Change the zoom level,
   (d) What are the information displayed about the current pixel under mouse pointer ?

4. Display dataset properties
   (a) What is the pixel resolution ?
   (b) What is image location ?
5. Dynamics setting :
   (a) Change the outlier rejection to 1%
   (b) Come back to min/max value

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.

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 ?

OTB applications throught Monteverdi2

1. Launch Monteverdi2
2. Find the list of available OTB Applications

Tips and Recommandations:

- **Monteverdi2** is under development vector data handling is not yet available, thus applications which rely on vector data are not ready to be used
- If any **OTB applications** seems to be available, check if *ITK_AUTOLOAD_PATH* variable is set with **OTB** binaries directory

OTB applications: Basic processing In this part of the exercise, you will use the following data:

`phr_xs_melbourne.tif`

1. Open the image in **Monteverdi2**.
2. Find the *BandMath* application launch it using **Monteverdi2** GUI. Open the image.
3. Using this application, compute the NDVI of the image:

$$NDVI = \frac{NIR - RED}{NIR + RED} \tag{1}$$

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

1. Using this application, build a mask of pixels whose Digital Number (DN) in the NIR channel is lower than 700. Visualize the input image and the mask in the same viewer.

Take carre about no data value which is set.

1. Using this application, build a mask of pixels whose DN is upper than 300 in all spectral bands. Visualize the input image and the mask in the same viewer.
2. Using the *Concatenate* application, 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.
3. Using the *Color Mapping* application, 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.
- MuParser library has some built-in functions and operators. More details can be found here
- The tooltip on expression area in bandMath application validate our formula

use SRTM information to display pixel elevation Elevation data for each image position can be found using DEM. Following steps are necessary to do that :

1. Set SRTM directory (use the *Edition Préférences* menu and fill *répertoire MNT* with SRTM directory)
2. Set the Geoid file (use the *Edition Préférences* menu and fill *Fichier Géoide* with geoid file)

Tips:

- SRTM data corresponding to any image can be automatically loaded or checked using **DownloadSRTMTiles** Application

```
$ otbcli_DownloadSRTMTiles -il
~/Demo_Product/PRIMARY_P_XS/IMG_PHR1A_MS_002/IMG_PHR1A_MS_201202250025599_SEN_PRG_FC_5847-002_R1C1.JP2
        -mode download -mode.download.outdir ~/Data/DEM/srtm_directory/
```

1.2 Segmentation

1.2.1 Description

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.

Data If you need to generate the data used in this exercise from the original products, you can use the following command lines.

```
$ otbcli_ExtractROI in ~/Demo_Product/ORTHO_PXS/IMG_PHR1A_PMS-N_001/IMG_PHR1A_PMS-N_201202250025599_ORT_PRG_FC_5855-001_R1C1.JP2
-out ~/Data/segmentation_small_xt_phr.tif uint16 -startx 11848 -starty 11426 -sizex 1024 -sizey 1024

$ otbcli_ExtractROI -in ~/Demo_Product/ORTHO_PXS/IMG_PHR1A_PMS-N_001/IMG_PHR1A_PMS-N_201202250025599_ORT_PRG_FC_5855-001_R1C1.JP2
-out ~/Data/segmentation_large_xt_phr.tif uint16 -startx 10240 -starty 10240 -sizex 4096 -sizey 4096
```

Pre-requisites

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

Achievements

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

1.2.2 Steps

Getting familiar with the Segmentation application

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

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 through in *raster* mode, using the connected components filter and a thresholding condition on the spectral distance
2. View the resulting segmentation in **Monteverdi2**. What do you see ?
3. Use the **ColorMapping** application to enhance the rendering of the result:

   (a) Try the *optimal* method

   (b) 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

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.
   (a) Try the default parameters first
   (b) 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.
   (a) Try the default parameters first
   (b) 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.

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

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 ?

1.3 Learning and classification from pixels

1.3.1 Description

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.

Data If you need to generate the data `melbourne_ms_toa_ortho_extract_small.tif` used in this exercise from the original products, you can use the following command lines.

To orthorectify the MS product:

```
$ otbcli_OrthoRectification
-io.in ~/Demo_Product/PRIMARY_P_XS/IMG_PHR1A_MS_002/IMG_PHR1A_MS_201202250025599_SEN_PRG_FC_5847-002_R1C1.JP2
-io.out ~/Data/Results/ortho_phr_ms_small.tif uint16 -outputs.mode auto -outputs.ulx 313892
-outputs.uly 5816639 -outputs.spacingx 2 -outputs.spacingy -2 -opt.ram 1024 -map utm
-map.utm.zone 55 -outputs.sizex  2048 -outputs.sizey 2048  -map.utm.northhem 0
```

To convert pixel values in top of atmosphere milli-reflectance:

```
 $ otbcli_OpticalCalibration  -in ~/Data/Results/ortho_phr_ms_small.tif
  -out ~/Data/Results/melbourne_ms_toa_ortho_extract_small.tif uint16 -milli 1
```

You can also generate also the `melbourne_ms_toa_ortho_extract_large.tif` with the same set of commands. You just need to change the size of the orthorectify region to 4096 instead of 2048.

Pre-requisites

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

Achievements

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

1.3.2 Steps

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

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*

Tips and Recommendations:

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

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?

Estimate classification model using the Support Vector Machine algorithm The **TrainImagesClassifier** 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?

Apply classification model

- Use the **ImageClassifier** 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`

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.

Try another classification method Since OTB 3.18 version a bridge to Open-CV have been done.So in addition to LibSVM, you can now use 8 other algorithms for your images classification tasks. All these classifiers can be reached directly from the **TrainImagesClassifier** application

- What are the classification method available
- Let's try another classifier for example the random forest classifier (or another if you want !!!)
- create a new model using **TrainImagesClassifier** application, then apply it using **ImageClassifier**
- which one is better ?

Fusion of Classification **FusionOfClassifications** application make it possible to fuse multiple classification maps into a single one. Two mode are available , a basic one : Majority Voting and Dempster-Shafer based fusion of classifications. This method will take advantage of the per-class strengh and weaknesses of each input classification (estimated from confusion matrices) to produce a robust output map combining the best of each input.

- Try to fuse two or more classification maps using majority voting. Take care about unspecified value label.
- Use Dempster Shafer based fusion mode

Estimate Classification accuracy using validation data **ComputeConfusionMatrix** allow to export classification results using raster validation image or vector data training sample

- Use QGis to produce validation sample (validation sample shouldn't contain any training sample in order to do accurate validation)
- Launch **ComputeConfusionMatrix** application, to estimate accuracy of produced map (fused or not).

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 To go further

plenty of exercises related to preprocessing, object based classification, stereo reconstruction,sar processing are available here in org format. Don't hesitate to try it and send us our feedbacks.