

Raster processing algorithms of the Orfeo Toolbox in QGIS

E. Christophe¹, M. Grizonnet², T. Dhar³, J. Michel²,
J. Inglada²

¹CENTRE FOR REMOTE IMAGING, SENSING AND PROCESSING,
NATIONAL UNIVERSITY OF SINGAPORE

²CENTRE NATIONAL D'ÉTUDES SPATIALES, TOULOUSE, FRANCE

³APOGEE IMAGING INTERNATIONAL, AUSTRALIA



Outline

History

What is Orfeo Toolbox?

Why doing that?

How?

Monteverdi

RoadMap

Example of ORFEO Users

Future of remote sensing applications

OTB and GIS

GIS capabilities in OTB

OTB - Quantum GIS plugins

What is Orfeo Toolbox (OTB)?

Initiated by CNES (French Space Agency)

- ▶ Following the feedback from SPOT satellite series
- ▶ In the frame of CNES ORFEO Program to prepare the launch of Pleiades (2011)

Goal

Make the development of new algorithms and their validation easier

Why?

- ▶ Traditionally (SPOT satellites): provide image simulations
- ▶ But here jump in resolution \Rightarrow develop new R&D methods
- ▶ To bridge the common gap between R&D study and tools available \Rightarrow the Orfeo Toolbox (OTB)



Progress

Everything begins (2006)

- ▶ Started in 2006 by CNES (French Space Agency), funding **several full-time developers**
- ▶ Targeted at high resolution images but with application to other sensors
- ▶ 4 year budget, over 1,000,000€ recently renewed for 1 additional year (500,000€)

Moving to user friendly applications (2008)

- ▶ Strong interactions with the end-user community highlighted that **applications for non-programmers** are important
- ▶ Several applications for non programmers (with GUI) since early 2008
- ▶ Several training courses (3/5-day courses) given in France, Belgium, Madagascar, UNESCO and Hawaii

But it's not just code

A few stats (active lines of codes: ohcount)

- ▶ Code: 75699 lines
- ▶ Test: 59410 lines
- ▶ Examples: 19254 lines
- ▶ Monteverdi: 39337 lines

Robust

- ▶ Nightly tests: 1500 tests
- ▶ Several platforms (linux, windows, OSX)
- ▶ Coverage > 85%



But it's not just code

Well documented

- ▶ Comprehensive software guide (pdf 700 pages)
- ▶ Doxygen
- ▶ Tutorials

Bindings

- ▶ Python
- ▶ Java



Software Guide example

Index

- III User's guide 55
 - ▷ Data Representation 57
 - ▷ Reading and Writing Images 93
 - ▷ Reading and Writing Vector Data 115
 - ▷ Basic Filtering 127
 - ▷ Image Registration 175
 - ▷ Disparity Map Estimation 235
 - ▷ Orthorectification and Map Projection 261
 - ▷ Radiometry 281
 - ▷ Image Fusion 299
 - ▷ Feature Extraction 307
 - ▷ Multi-scale Analysis 361
 - ▷ Image Segmentation 375
 - ▷ Classification 405
 - ▷ Object-based Image Analysis 473
 - ▷ Change Detection 481
 - ▷ Geospatial analysis 501
 - ▷ Image Visualization and output 503
 - ▷ Online data 513
- IV Developer's guide 519
- ▷ V Appendix 581

16.1.2 Otsu Segmentation

Another criterion for classifying pixels is to minimize the error of misclassification. The goal is to find a threshold that classifies the image into two clusters such that we minimize the area under the histogram for one cluster that lies on the other cluster's side of the threshold. This is equivalent to minimizing the within class variance or equivalently maximizing the between class variance.

The source code for this example can be found in the file `Examples/Segmentation/OtsuThresholdImageFilter.cxx`.

This example illustrates how to use the `itk::OtsuThresholdImageFilter`.

```
#include "itkOtsuThresholdImageFilter.h"
```

The next step is to decide which pixel types to use for the input and output images.

```
typedef unsigned char InputPixelType;
typedef unsigned char OutputPixelType;
```

The input and output image types are now defined using their respective pixel types and dimensions.

```
typedef otb::Image<InputPixelType, 2> InputImageType;
typedef otb::Image<OutputPixelType, 2> OutputImageType;
```

The filter type can be instantiated using the input and output image types defined above.



Does it work?

Is it successful so far?

- ▶ OTB user community **growing steadily** (programmers and application users)
- ▶ Presented at IGARSS and ISPRS in 2008, special session in IGARSS in 2009, presented at Foss4g in 2009, tutorial at IGARSS in 2010; 4 papers using OTB at Foss4g 2010
- ▶ There is planning to extend the budget for several more years
- ▶ Value analysis is very positive (cf. Ohloh): **re-using is powerful**



Why doing that?

Why make a multi-million dollar software and give it for free?

- ▶ The French space agency (CNES) is not a software company, its goal is to promote space technologies and encourage the development of new applications.
- ▶ CNES makes satellites and wants to make sure the **images are used**
- ▶ One goal is to **encourage research**: it is critical for researchers to know what is in the box



How?

How to reach this goal?

Using the best work of others: do not reinvent the wheel

Many open-source libraries of good quality

- ▶ ITK: software architecture (streaming, multithreading), many image processing algorithms
- ▶ Gdal/Ogr: reading data format (geotiff, raw, png, jpeg, shapefile, ...)
- ▶ Ossim: sensor models (Spot, RPC, SAR, ...) and map projections
- ▶ 6S: radiometric corrections
- ▶ and many other: libLAS (lidar data), Edison (Mean Shift clustering), libSiftFast (SIFT), Boost (graph), libSVM (Support Vector Machines), Mapnik (vector data representation)

⇒ all behind a common interface

How?

How to reach this goal?

Using the best work of others: do not reinvent the wheel

Many open-source libraries of good quality

- ▶ ITK: software architecture (streaming, multithreading), many image processing algorithms
- ▶ Gdal/Ogr: reading data format (geotiff, raw, png, jpeg, shapefile, ...)
- ▶ Ossim: sensor models (Spot, RPC, SAR, ...) and map projections
- ▶ 6S: radiometric corrections
- ▶ and many other: libLAS (lidar data), Edison (Mean Shift clustering), libSiftFast (SIFT), Boost (graph), libSVM (Support Vector Machines), Mapnik (vector data representation)

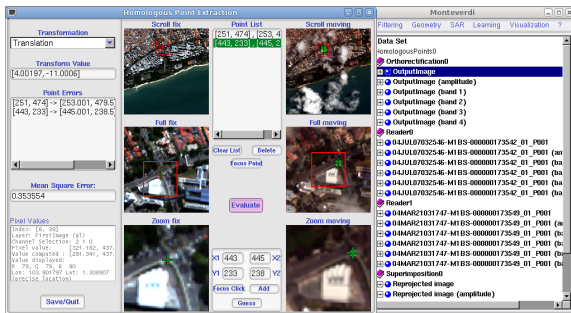
⇒ all behind a common interface



Monteverdi

Capacity building

- ▶ Strong demand to provide tools for capacity building
- ▶ Decision to start an integrated application based on OTB
- ▶ Development started last year (September 2009)



Outline

History

What is Orfeo Toolbox?

Why doing that?

How?

Monteverdi

RoadMap

Example of ORFEO Users

Future of remote sensing applications

OTB and GIS

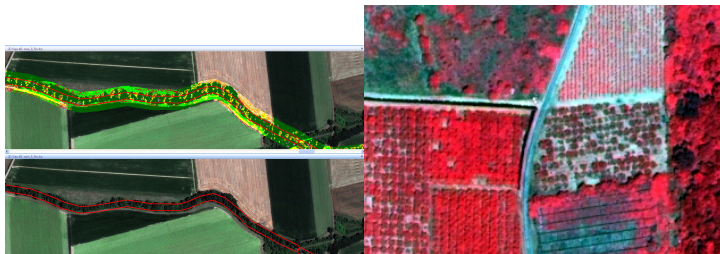
GIS capabilities in OTB

OTB - Quantum GIS plugins

Context

Academic and commercial research

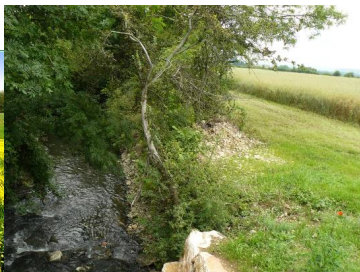
- ▶ Hydrological network extraction (*Mines d'Alès*)
- ▶ Artificial drainage linears identification (*SOGREAH*)
- ▶ Global agricultural monitoring systems (*CEMAGREF*)



Context

National & European environmental policy

- ▶ In the frame of water & nitrate European directives
- ▶ Mapping natural area for biodiversity assessment and protection
- ▶ Potential of VHSR imagery



Finally

Complete package

- ▶ From best effort to production
- ▶ Different category of users without remote sensing knowledge
- ▶ **Ecology minister regional delegations** ⇒ Package : OTB + QGIS + Pleiades imagery



GIS integration

- ▶ Need for an “operational & validated methodology” applied at regional scale
- ▶ More interaction between remote sensing and GIS application
- ▶ Drive the development of OTB/QGIS plugins



Outline

History

What is Orfeo Toolbox?

Why doing that?

How?

Monteverdi

RoadMap

Example of ORFEO Users

Future of remote sensing applications

OTB and GIS

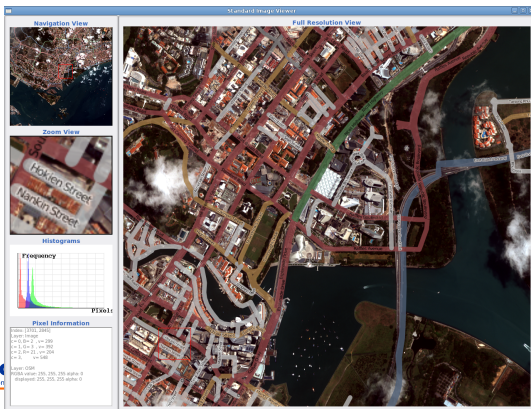
GIS capabilities in OTB

OTB - Quantum GIS plugins

GIS in OTB

Several softwares

- ▶ Integration of Mapnik with reprojection to display vector data
- ▶ PostGIS, SpatiaLite integration
- ▶ Use of Open Street Map data for image to database registration



Use new data source

Number of sources

- ▶ Tiles: OSM, Near Map, etc.
- ▶ Export to mapfile (in development)
- ▶ Use IGN aerial photos (now freely available)

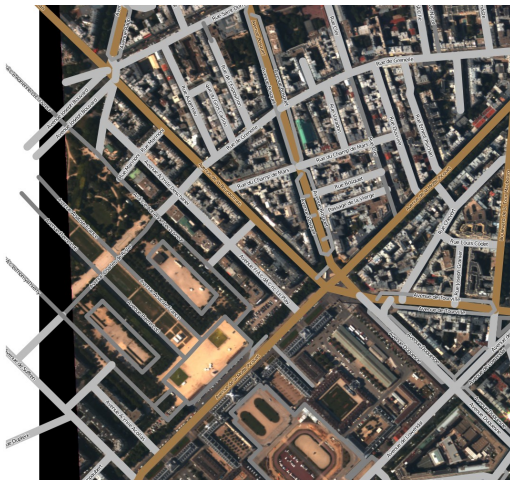
IGN aerial imagery



Orthorectified IGN aerial imagery



Orthorectified IGN aerial imagery



Done yesterday (thanks to J. Malik)

With all tools included in OTB

- ▶ Data read by Gdal
- ▶ GCP entered in Monteverdi
- ▶ with tiles from OSM
- ▶ RPC model by Ossim
- ▶ resampling by ITK
- ▶ OSM shapefile read with OGR
- ▶ rendered with Mapnik
- ▶ soon in QGIS



How to build a QGIS plugin: recipes!

Example from OTB software guide

The source code for this example can be found in the file `Examples/BasicFilters/HillShadingExample.cxx`.

Visualization of digital elevation models (DEM) is often more intuitive by simulating a lighting source and generating the corresponding shadows. This principle is called hill shading.

Using a simple functor [otb::HillShadingFunctor](#) and the dem image generated using the [otb::DEMtoImageGenerator](#) (refer to [7.1](#)) you can easily obtain a representation of the DEM. Better yet, using the [otb::ScalarToRainbowRGBPixelFunctor](#), combined with the [otb::ReliefColormapFunctor](#) you can easily generate the classic elevation maps.

This example will focus on the shading itself.

After generating the dem image as in the DEMtoImageGenerator example, you can declare the hill shading mechanism. The hill shading is implemented as a functor doing some operations in its neighborhood. A convenient filter called [otb::HillShadingFilter](#) is defined around this mechanism.

```
typedef otb::HillShadingFilter<ImageType, ImageType> HillShadingFilterType;
HillShadingFilterType: Pointer hillShading = HillShadingFilterType: New();
hillShading->SetRadius(1);
hillShading->SetInput(demToImage->GetOutput());
```

Figure [21.6](#) shows the hill shading result from SRTM data.

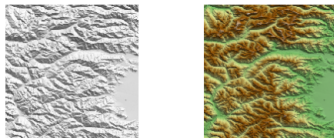


Figure 21.6: Hill shading obtained from SRTM data (left) and combined with the color representation (right)

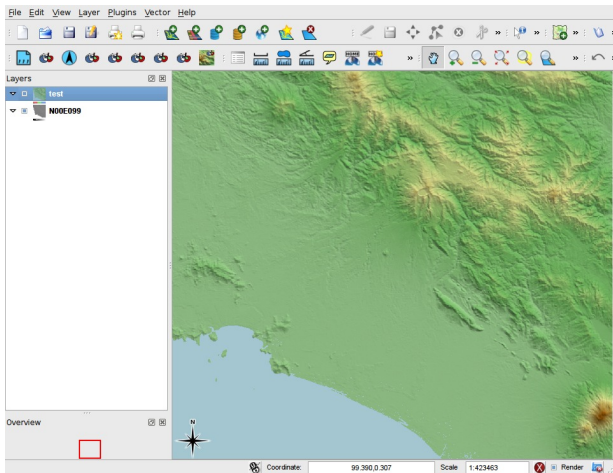
QGIS integration

Quantum GIS plugins (C++)

- ▶ Create otbgui (qt designer)
- ▶ Create HillShadeFilter (based on the OTB example)
- ▶ Add the plugin to the otb generic plugin constructor
- ▶ Drawback: need to develop a new plugin for each feature



Plugin screenshot



Several plugins since last year

- ▶ Classification: KMeans, SVM, SOM
- ▶ Segmentation
- ▶ Disparity map
- ▶ Hill shading
- ▶ more coming...

Even more since last night

Reprojection

- ▶ Birds of a feather QGIS session yesterday night
- ▶ One user ask for raster reprojection feature

Extreme coding

- ▶ First prototype committed last night
- ▶ Still some GUI adjustment to make

Monteverdi: pipeline front end

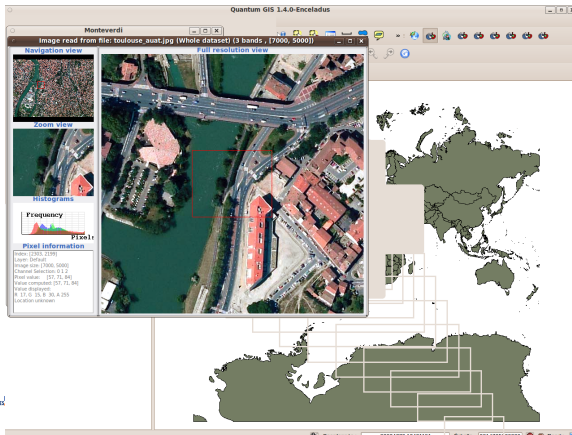
Pipeline

- ▶ End to end processing: raster reprojection (GCP), land cover, etc
- ▶ Gui interface (based on FLTK)
- ▶ OTB pipeline: able to process large images, multiprocessor
- ▶ Module system: easy to customize the application

Monteverdi in QGIS

Brute force integration

- ▶ Launch Monteverdi in a specific plugin
- ▶ Automatic I/O between QGIS layers and Monteverdi dataset
- ▶ Drawback: not so nice integration with QGIS



Streamed OTB results in QGIS viewer

Technical points

- ▶ Objective: Provide directly the result of the processing without going through disk writing
- ▶ Based on the WMS provider
- ▶ Suggestion to use the plugin layer on QGIS developers mailing list



Where can you find informations?

- ▶ <http://www.orfeo-toolbox.org/>
- ▶ <http://groups.google.fr/group/otb-users>
- ▶ http://wiki.orfeo-toolbox.org/index.php/Quantum_GIS_plugins



Questions?



A bit of code

▶ Example

```
#include "otbImage.h"
#include "otbImageFileReader.h"
#include "otbImageFileWriter.h"

int main(int argc, char * argv[])
{
    typedef otb::Image<unsigned char, 2> ImageType

    typedef otb::ImageFileReader<ImageType> ReaderType;
    typedef otb::ImageFileWriter<ImageType> WriterType;

    WriterType::Pointer writer = WriterType::New();
    ReaderType::Pointer reader = ReaderType::New();

    reader->SetFileName(argv[1]);
    writer->SetFileName(argv[2]);

    writer->SetInput(reader->GetOutput());
    writer->Update();

    return EXIT_SUCCESS;
}
```

