

Earth observation image processing with the ORFEO ToolBox

Remote sensing real image processing

M. Grizonnet¹

¹FRENCH SPACE AGENCY , TOULOUSE, FRANCE

Part of the presentation is derived for a tutorial given by J. Inglada and E. Christophe at IGARSS: "Pragmatic Remote Sensing". This content is provided under a Creative Commons Attribution-ShareAlike 3.0 Unported License. 

What?

Remote sensing

- ▶ Reading images, Accessing metadata
 - ▶ Implementing state of the art algorithms → Reproducible research
- ⇒ to be able to **extract the most information**, we need to **use the best** of what is available: data, algorithms,...



What is Orfeo Toolbox (OTB)?

In the frame of CNES ORFEO Program - Very High Resolution images

Goal

Make the development of new algorithms and their validation easier

- ▶ C++ library: provide many algorithms (pre-processing, image analysis) with a common interface
- ▶ Open-source: free to use, to modify, you can make your own software based on OTB and sell it
- ▶ Multiplatform: Windows, Linux, Unix, Mac

End of the story : 2011 - Launch of Pleiades 1A



Crédits : Pléiades 1A - CNES 2012

A bit of History

Everything begins (2006)

- ▶ Started in 2006 by CNES (French Space Agency), funding **several full-time developers**
- ▶ Targeted at high resolution images (Pleiades) but with application to other sensors
- ▶ 4 year budget, over 1,000,000€ recently renewed for 3 additional years

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, Hawaii...

A bit of history (2)

Monteverdi (2009)

- ▶ Modular software with GUI → Access to some of OTB filters
- ▶ Use first for technical courses (*Capacity Building*)
- ▶ User feedback : great interest for this type of tool

Interoperability (2011) OSGeo community

- ▶ Framework **OTB-Applications**
- ▶ Plugin based architecture
- ▶ one code → multiple targets
- ▶ Example : Quantum GIS plugins (*via Sextante*)
- ▶ In parallel : continue to add new algorithms added in the C++ library

Why doing that?

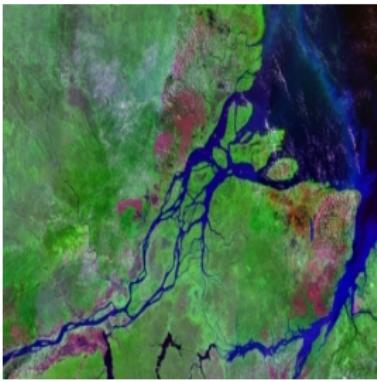
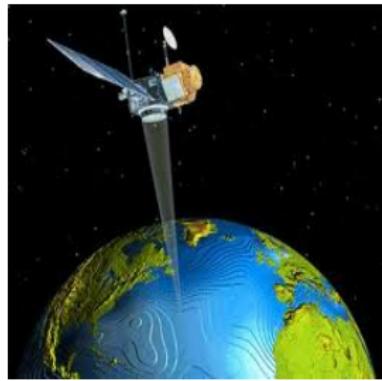
Is it successful so far?

- ▶ OTB user community **growing steadily** (programmers and application users)
- ▶ Presented at IGARSS and ISPRS in 2008, special session in IGARSS (2009,2010,2011)
- ▶ CNES is planning to extend the budget for several more years
- ▶ Value analysis is very positive (cf. Ohloh): **re-using is powerful**

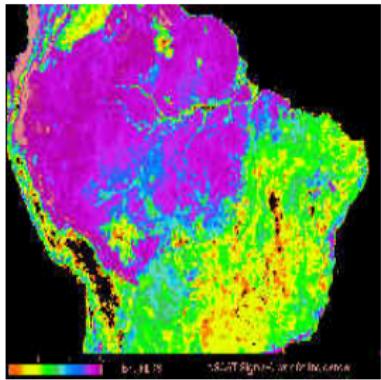
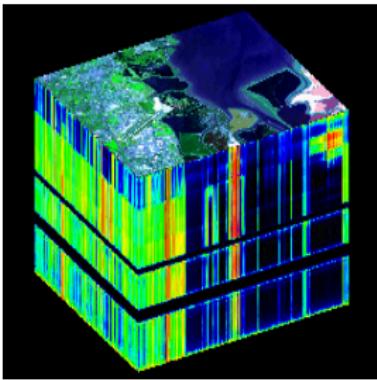
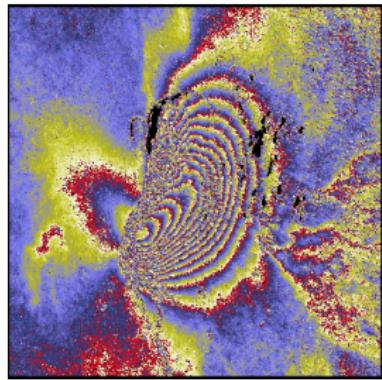
Why make a multi-million dollar software and give it for free?

- ▶ CNES is not a software company
- ▶ One goal is to **encourage research**: it is critical for researchers to know what is in the box
- ▶ CNES makes satellites and wants to make sure the **images are used**
- ▶ if more people have the tools to use satellite images, it is good for CNES

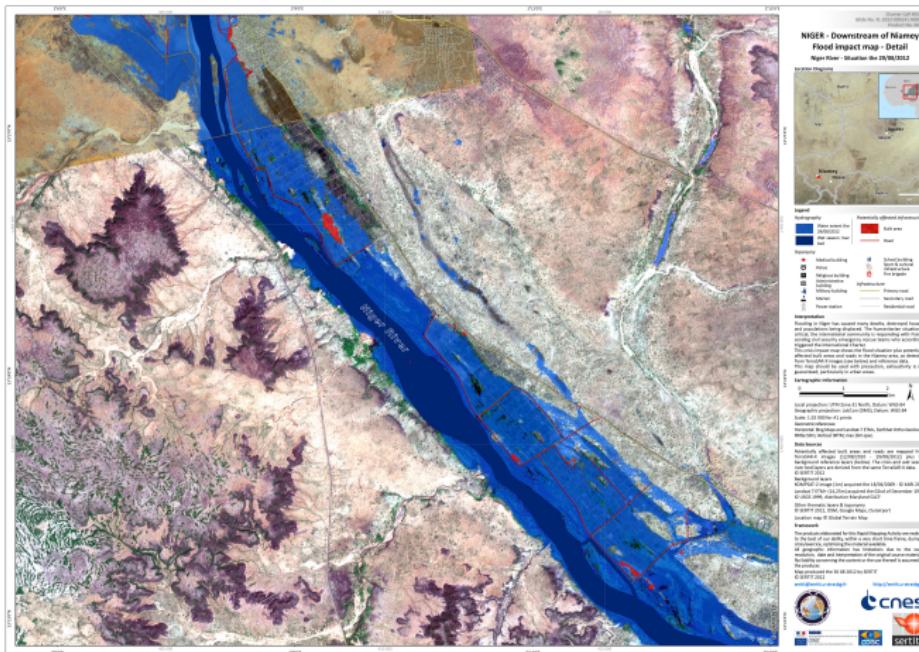
Remote sensing illustrations



Remote sensing illustrations



Remote sensing illustrations



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, ...)
- ▶ **OpenJPEG**: reading and writing of large data in jpeg2000 format
- ▶ **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)

⇒ all behind a common interface

Components available

Currently

- ▶ Most satellite image formats
- ▶ Geometric corrections
- ▶ Radiometric corrections
- ▶ Change detection
- ▶ Feature extraction
- ▶ Segmentation
- ▶ Classification

Huge documentation available

- ▶ Software Guide (+700 pages pdf), also the online version
- ▶ CookBook online version OTB recipes
- ▶ Doxygen: documentation for developers

A powerful architecture

Modular

- ▶ Easy to combine different blocks to do new processing

Scalable

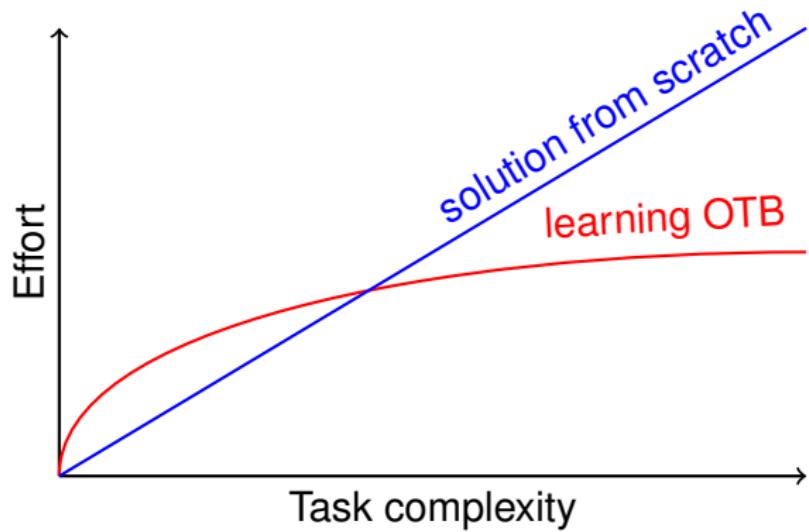
- ▶ Streaming (processing huge images on the flow) transparent for the user of the library
- ▶ Multithreading (using multicore CPUs) also

But a steep learning curve for the programmer

Advanced programming concepts

- ▶ Template metaprogramming (generic programming)
- ▶ Design patterns (Factory, Functors, Decorators, Smart Pointers, ...)

Steep learning curve

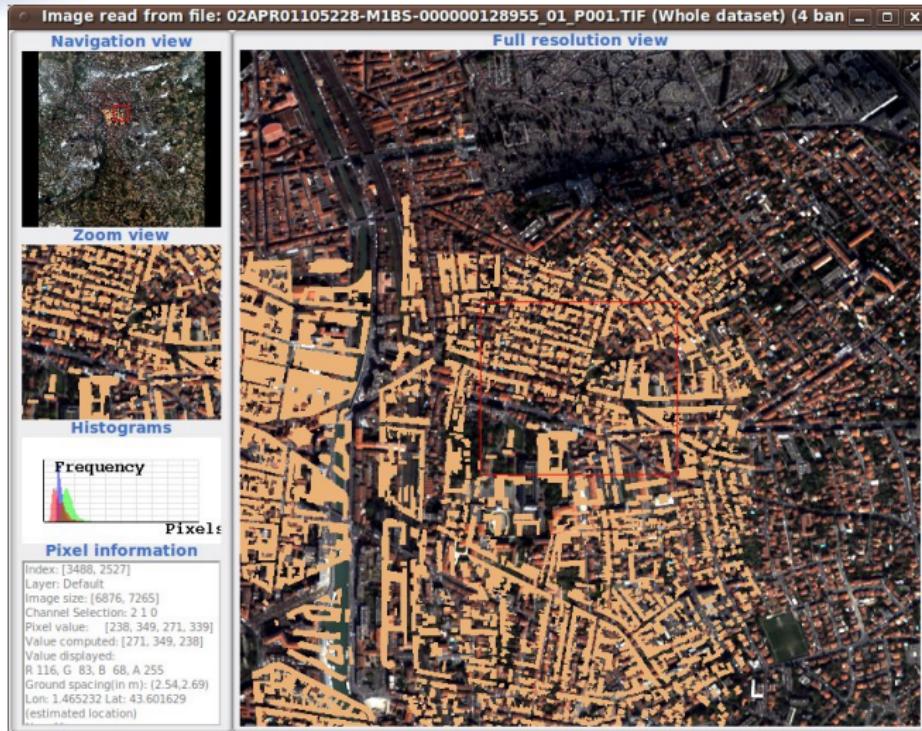


Ask questions

As for everything: easier when you're not alone

- ▶ Much easier if you have somebody around to help!
- ▶ We didn't know anything not so long ago...
- ▶ Not surprising that most software companies now focus their offer on support: help is important

Making it easier for the users: Monteverdi



Code!

```
#include "otbImage.h"
#include "otbImageFileReader.h"
#include "otbStreamingImage.FileWriter.h"
#include "itkCannyEdgeDetectionImageFilter.h"
#include "itkRescaleIntensityImageFilter.h"

int main(int argc, char * argv[])
{
    typedef double PixelType;
    typedef otb::Image<PixelType> ImageType;

    typedef unsigned char OutputPixelType;
    typedef otb::Image<OutputPixelType> OutputImageType;

    typedef otb::ImageFileReader<ImageType> ReaderType;
    ReaderType::Pointer reader = ReaderType::New();

    reader->SetFileName(argv[1]);

    typedef itk::CannyEdgeDetectionImageFilter
    <ImageType, ImageType> FilterType;
    FilterType::Pointer filter = FilterType::New();

    filter->SetInput(reader->GetOutput());
```

I want some more (code)

```
typedef itk::RescaleIntensityImageFilter<ImageType, OutputImageType> RescalerType;
RescalerType::Pointer rescaler = RescalerType::New();

rescaler->SetOutputMinimum(0);
rescaler->SetOutputMaximum(255);

rescaler->SetInput(filter->GetOutput ());

typedef otb::StreamingImageFileWriter<OutputImageType> WriterType;
WriterType::Pointer writer = WriterType::New();

writer->SetFileName(argv[2]);

writer->SetInput(rescaler->GetOutput ());

writer->Update();

return EXIT_SUCCESS;
}
```

Calling applications from Python

```
#!/usr/bin/python

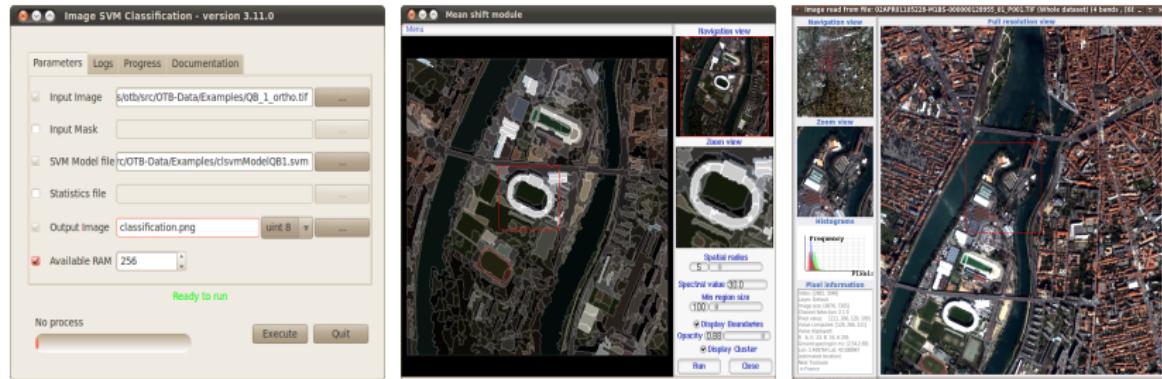
# Import the otb applications package
import otbApplication

# The following line creates an instance of the ImageSVMClassifier
# application
ImageSVMClassifier =
    otbApplication.Registry.CreateApplication("ImageSVMClassifier")

# The following lines set all the application parameters:
ImageSVMClassifier.SetParameterString("in", "QB_1_ortho.tif")
ImageSVMClassifier.SetParameterString("imstat",
    "clImageStatisticsQB1.xml")
ImageSVMClassifier.SetParameterString("svm", "clsSvmModelQB1.svm")
ImageSVMClassifier.SetParameterString("out", "classification.png")
ImageSVMClassifier.SetParameterOutputImagePixelType("out", 1)

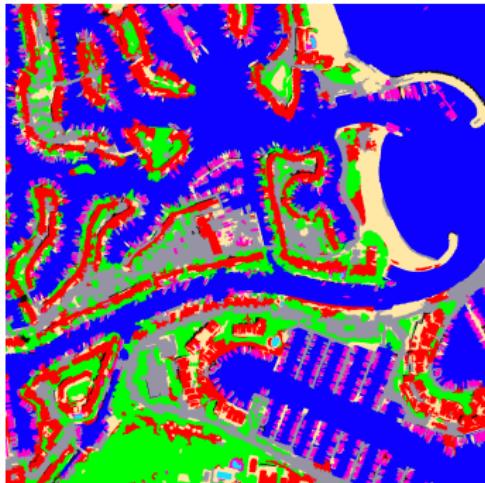
# The following line execute the application
ImageSVMClassifier.ExecuteAndWriteOutput()
```

OTB access - Applications, Monteverdi . . .



And now?

- ▶ OTB will continue after the ORFEO program (VHR, SAR, MX,HX...)
- ▶ Open software → Open Data now?



Size does matter...

- ▶ Remote sensing data become more accessible...
- ▶ But...need (among other things) dedicated tool to manage it
- ▶ SMOS : 11 Tera bytes per year
- ▶ **Sentinel-2** (10m/60M resolution) :Systematic global coverage of land surfaces : from 56°South to 84°North!
- ▶ → Lewis Carroll's Sylvie and Bruno Concluded: a fictional map that had "the scale of a mile to the mile."

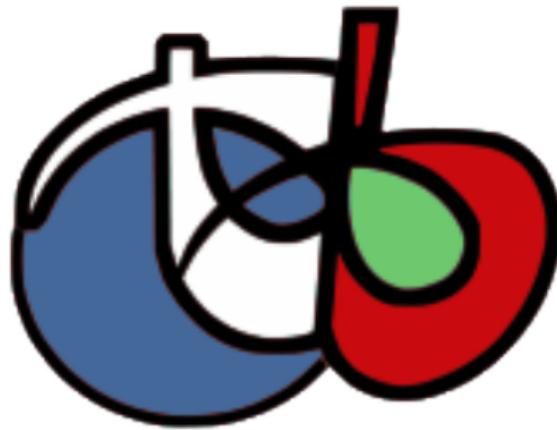


Capitole du libre 2012

Where can you find informations?

- ▶ <http://www.orfeo-toolbox.org/>
- ▶ <http://groups.google.fr/group/otb-users>
- ▶ <http://groups.google.fr/group/otb-developers>
- ▶ <http://blog.orfeo-toolbox.org/>
- ▶ <http://wiki.orfeo-toolbox.org/>
- ▶ <http://blog.jordiinglada.net/>

Questions?



ORFEO ToolBox is not a black box

(OTB slideshow)