Earth observation image processing with the ORFEO ToolBox
Remote sensing real image processing

M. Grizonnet\(^1\), J. Michel\(^1\)

\(^1\)French 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.
Why?

Common problems

- 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

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
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)

- 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)
- 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
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

Module architecture

- Standard input/output
- Easy to customize for a specific purpose
- Full streaming or caching the data
- Graphical pipeline
- Classic: viewer, threshold...
- Advanced VHR image analysis: classification, feature extraction, fusion, change detection...
Making it easier for the users: Monteverdi
Bindings: access through other languages

Not everybody uses C++!

- Bindings provide an access to the library through other languages
- **Python**: available
- **Java**: available with also JVM (Clojure, Scala, etc.
- **IDL/Envi**: cooperation with ITT VIS to provide a method to access OTB through idl/envi (working but no automatic generation)
- **Matlab**: ser contribution (R. Bellens from TU Delft)
- Other languages supported by Cable Swig might be possible (Tcl, Ruby...)
No C++ and no templates please!

- Get benefit of the OTB pipeline architecture from other software (auto magic I/O, streaming, threading, filters...)
- Plugin based architecture

Wrappers

- CLI interface
- GUI interface (QT)
- Python API
- QGIS plugins
```cpp
#include "otbImage.h"
#include "otbImageFileReader.h"
#include "otbStreamingImageFileWriter.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());
}
```
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;
}
#!/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", "clsvmModelQB1.svm")
ImageSVMClassifier.SetParameterString("out", "classification.png")
ImageSVMClassifier.SetParameterOutputImagePixelType("out", 1)

# The following line execute the application
ImageSVMClassifier.ExecuteAndWriteOutput()
IPOL workshop 2012

OTB access - Applications, Monteverdi...
Image access

Pléiades data support

- Improve support for Jpeg2000 via OpenJPEG\(^1\)
- Metadata access (RPC model) via OSSIM\(^2\)
- Metadata access (from DN to reflectance)
- Direct access to other resolution (thanks to Jpeg2000)
- Efficient visualization and navigation in Monteverdi

\(^1\)https://code.google.com/p/openjpeg/
\(^2\)Awesome image Processing
Uncompress Pleiades tiles PHR JPEG2000

Figure: Extraction ROI - Pléiades 50cm
Classification and OBIA

- OBIA : Object Based Image Analysis
- OBIA : lots of methods published
- Labeling objects not pixels
- Improve efficiency of structural approaches

Classification

- Supervised classification based on objects attributes
- SVM for example (but others can be plugged)
Results

Original image (Massif des Maures - QuickBird pan-sharpening 0.7m)
Results

Segmented image (mean-shift here) - Crucial point!
Results

Object based classification (SVM in this case)
OBIA

- Challenge: Large scale analysis with export to SIG
- Performance, ergonomy (vector/raster), streaming, multi-threading
- Different access: C++ classes, applications (supervision needs a good user interface)
Massive segmentation or nothing

Principe

- Large scale segmentation
- Segmentation quality measurement! → Hoover scores
- Efficient and correct segmentation → “Cachez cette tuile que je ne saurait voir”
- Objects manipulation? Efficient conversion between raster/vector?

Principles

- Apply segmentation (many algorithms available)
- Vectorized results on each tile (memory management)
- Concatenate vector data and remove tile effects (fusion)
Tiles sucks!
Available in 3.14

- Threaded mean-shift algorithm
- Generic filters for tiled based segmentation
- Method for polygons fusion near tiles region
- Fully compatible with OGR
Stereo reconstruction with OTB

- Stereo-rectification of optical images pair based on sensor model
- Complete spectrum of stereo correspondence algorithms has been published
- Block matching, sub-pixel estimation of disparity, median filter...
- Use SRTM as initial disparity (DEM)
- What’s new? Use/Combine with feature extraction pipeline (common interface)

Figure: DEM result from Pleiades stereo pair over the Pyramid of Cheops!
And now?

- ORFEO program - Thematic in-flight commissioning activities until mid-2013
- OTB will continue after the ORFEO program (VHR, SAR, MX, HX...)
Introduction

What?  What’s new in OTB 3.12 (February)  What’s new in OTB 3.14  Extra

OTB → On the road to ITKv4

ITK

► 10 years project
► 5M de $ (Leverage)
► From the American Recovery and Reinvestment Act of 2009
► ITKv4 : The next generation 1.5 years project -> Prepare next 10 years

Functionalities

► GPGPU -> massively parallel processing power
► Modularity
► Review, refactoring, new filters...
► OTB : Reuse is better than recycling
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/
Questions?

ORFEO ToolBox is not a black box