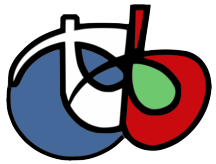
Orfeo Toolbox (OTB)

from satellite images to geographic information



"Orfeo Toolbox is not a black box"

Outline

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

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 (based on the CeCILL licence)
- Multiplatform: Windows, Linux, Unix, Mac

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 to be launched in 2010) but with application to other sensors
- 4 year budget, over 1,000,000€

Moving towards user friendly applications (2008)

- Strong interactions with the user community highlighted that applications for non-programmers are important
- Several applications for non programmers (with GUI) since early 2008
- Several courses (3/5-day courses) given in several French and Belgian institutions (Cesbio, RMA, ENST,...)

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
- There is planning to extend the budget for several more years
- Value analysis is very positive (cf. Ohloh): re-using is powerful

access to the online documentation from August 1st, 2009 until today

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

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)

$\Rightarrow$ all behind a common interface

Outline

Application

Currently

- Image viewer
- Image Segmentation
- Image Classification (by SVM)
- Land Cover
- Feature Extraction
- Road Extraction
- Orthorectification (with Pan Sharpening)
- Fine registration
- Image to database registration
- Object counting
- Urban area detection
- $\Rightarrow$ major changes coming

Components available

Currently

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

Huge documentation available

- Software Guide (+600 pages pdf), also the online version
- 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)

But a steep learning curve for the programmer

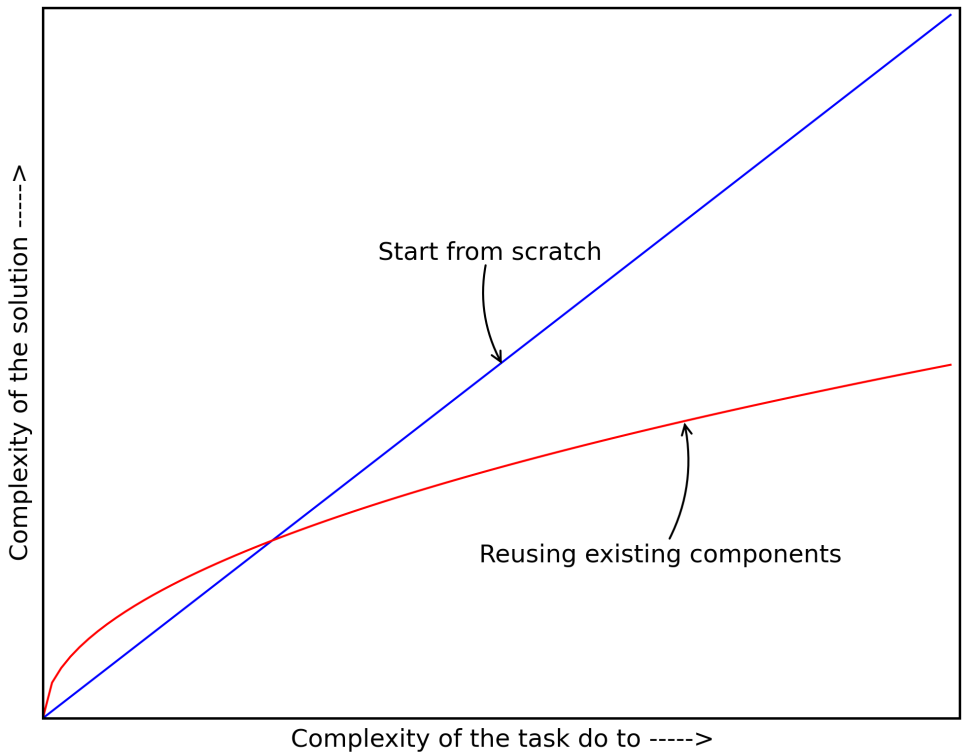
Advanced programming concepts

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

But a steep learning curve for the programmer

Advanced programming concepts

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



Open source community

Ask questions: easier when you're not alone

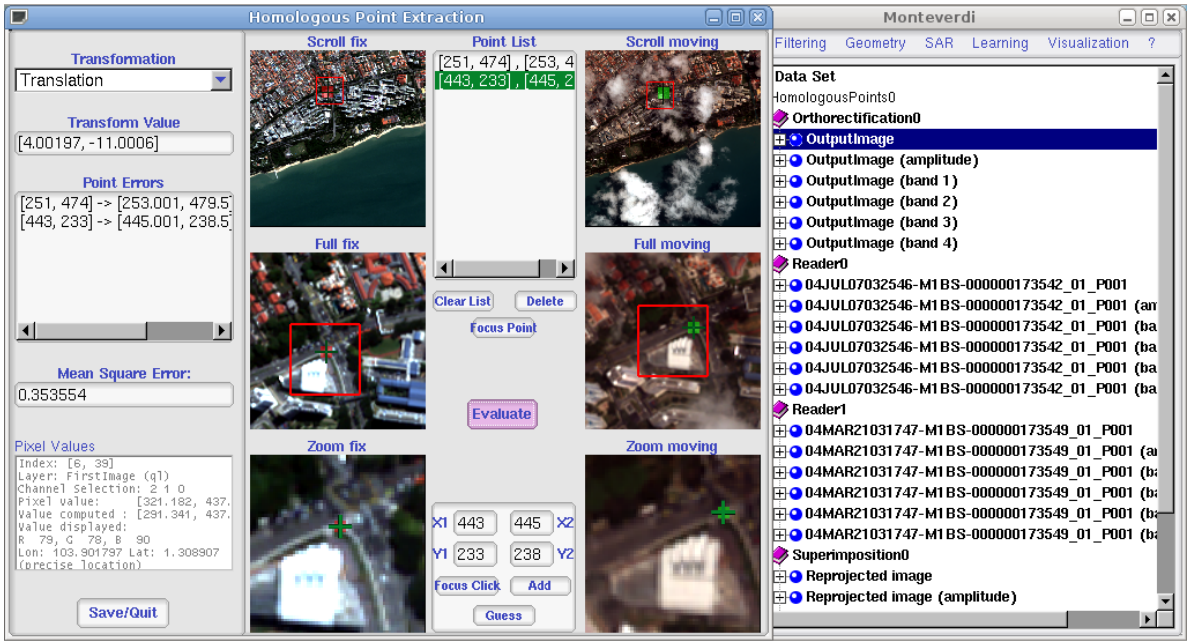
- Much easier if you have somebody around to help!
- Strong community around the OTB user mailing list: otb-users@googlegroups.com
- Replies usually come fast

Outline

Monteverdi

Capacity building

- Strong demand to provide tools for capacity building
- Decision to start an integrated application based on OTB
- Developpement started last month (September 2009)
- See demo later



GIS integration

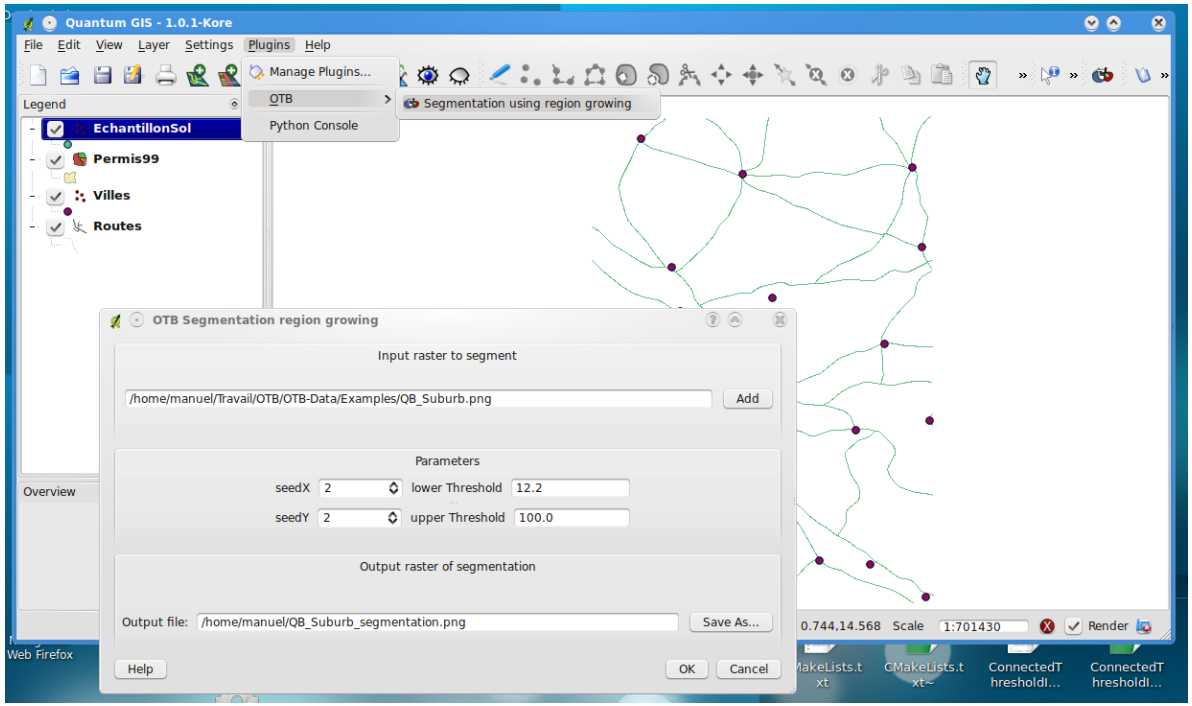
PostGIS

- Work in progress to integrate the connection to PostGIS database (IO)
- Querying: use the geographic capabilities in feature extraction algorithms (a building has a shadow in the north west for example)

GIS integration

Quantum GIS

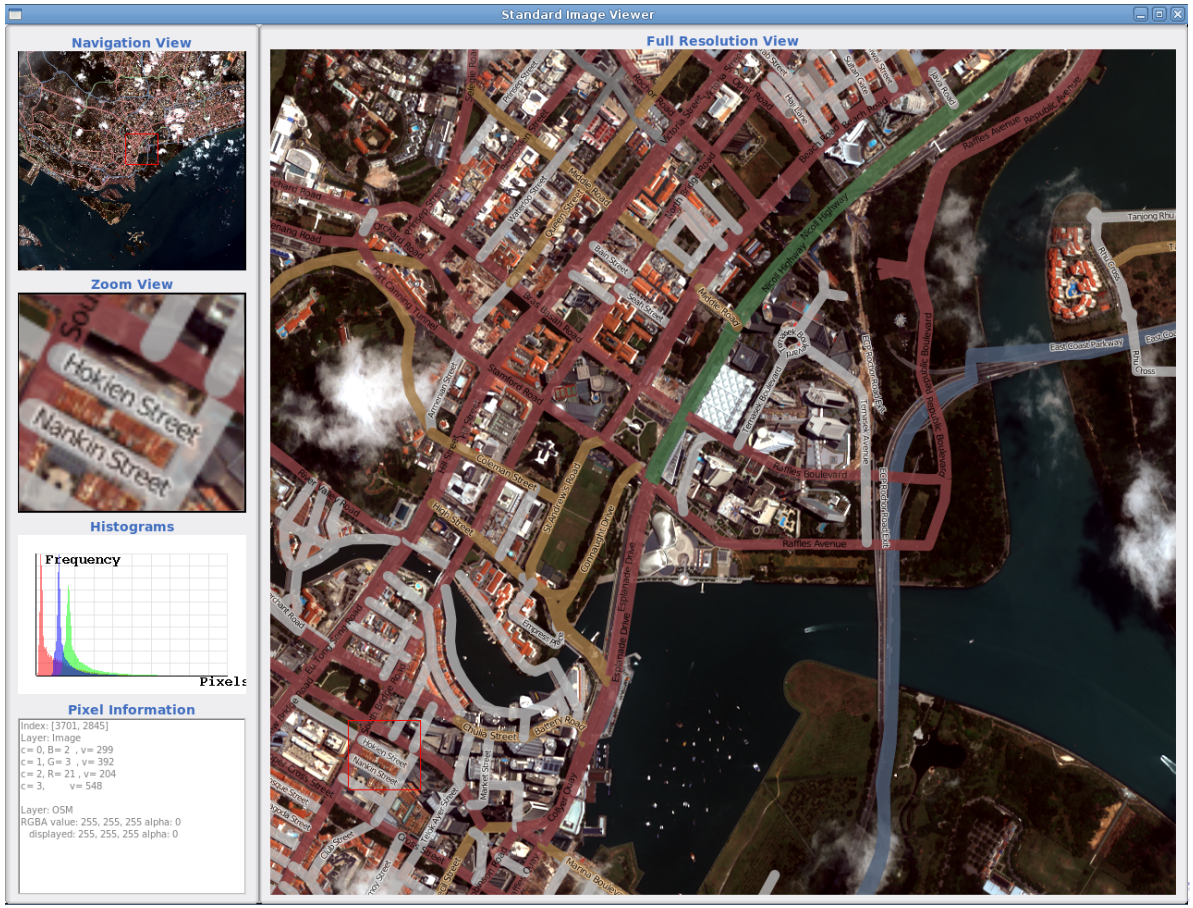
- Quantum GIS is a very user-friendly open source GIS application
- Work going on to access the OTB capabilities within Quantum GIS (plugins)



GIS integration

Mapping

- Stronger integration with mapping and data representation
- Integration of Mapnik, Open Street Map data



Bindings: access through other languages

Not everybody uses C++!

- Bindings provide an access to the library through other languages
- Python: mostly working but not fully tested
- Java: some work is still needed to be able to use OTB from java but most of the work is already done
- IDL/Envi: cooperation with ITT VIS to provide a method to access OTB through idl/envi (working but no automatic generation)
- Matlab: recent user contribution (R. Bellens from TU Delft)
- Other languages supported by Cable Swig might be possible (Tcl, Ruby?)

GPU and clusters

Now

- Can automatically use several cores on the same CPU (shared memory)
- This is transparent for the user thanks to the ITK architecture

GPU and clusters

- Some talks are going on between ITK and OTB to find the best solution: general, easy to use...
- Cuda is too dependent on the hardware but OpenCL is leading an effort to get a common architecture (Apple, AMD, Intel, Nvidia)
- Probably some novelties in the coming months

Questions?

A bit of code

```
#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;
  ReaderType::Pointer reader = ReaderType::New();

  typedef otb::ImageFileWriter<ImageType> WriterType;
  WriterType::Pointer writer = WriterType::New();

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

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

  return EXIT_SUCCESS;
}
```