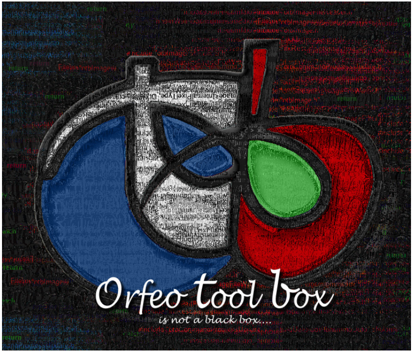
Orfeo ToolBox

Open source processing of remote sensing images (updated for 5.6)

OTB Team CNES



CENTRE NATIONAL D'ÉTUDES SPATIALES

Things to know about OTB. . .

Orfeo ToolBox is:

- An **image processing library** for remote sensing
- **Free and open source software** under CeCILL-v2 license (equivalent to GPL)
- **Funded and developed by CNES** (French Space Agency) in the frame the development of the Pléiades satellite (and beyond)
- Used at CNES, ESA (European Space Agency), mission exploitation platforms, remote sensing labs, teaching. . .
- Written in **C++** on top of ITK (medical image processing)
- Built on the shoulders of giants (GDAL, OSSIM, OpenCV. . .)
- **Big Data** capable, thanks to built-in streaming and multithreading

orfeo-toolbox.org

Why open source?

Maximum reach

OTB is dedicated to every user of satellite images. Its wide dissemination contributes to the missions success (Pléiades, Sentinels. . .)

Quality and efficiency

OTB covers a vast panel of applications and thematic fields. Openness should:

- Facilitate appropriation and validation for users
- Encourage contributions and bug reports
- Available on multiple platforms
- "The Cathedral & the Bazaar"[1]: the more widely available the source code is for public testing experimentation, the more rapidly all forms of bugs will be discovered

Reproducible research

OTB capitalizes a part of the CNES R&D in IP, open source contributes to transparent, reproducible and trans-disciplinary research.

[1] http://www.catb.org/esr/writings/cathedral-bazaar/

Outline

Functions and algorithms

Key characteristics

How to use OTB?

What's new in OTB?

Conclusion

Incomplete list of OTB functions

Pre-processing

- Radiometric calibration, orthorectification, resampling (raster and vector), pan-sharpening, stereo rectification...
- Sensor supported: Sentinels, Pléiades, SPOT6, SPOT5, Digital Globe satellites
- Geometric models (thanks to OSSIM), support for DEM (SRTM or GeoTIFF)

Images and vector manipulation

- Formats supported by GDAL (raster and vector), conversion raster/vector
- Region of interest extraction, of spectral bands, concatenation or splitting...
- Band math, color mapping, contrast enhancement
- Linear filtering, Mathematical morphology

(Incomplete) List of OTB functions

Feature extraction

- Edge detection, scale-invariant feature transform, lines, corners
- Radiometric indices, textures (Haralick, SFS, PanTex)
- Local statistics (Flusser moments, Histogram of Oriented Gradient)
- Keypoints matching (SIFT, SURF. . .)

Change detection

- Classic methods with image metrics comparison
- Multivariate Alteration Detector

Dimensionality reduction, hyperspectral processing

- PCA, NAPCA, ICA, MAF. . .
- Dimension estimation, endmembers extraction, Vertex Component Analysis(VCA)

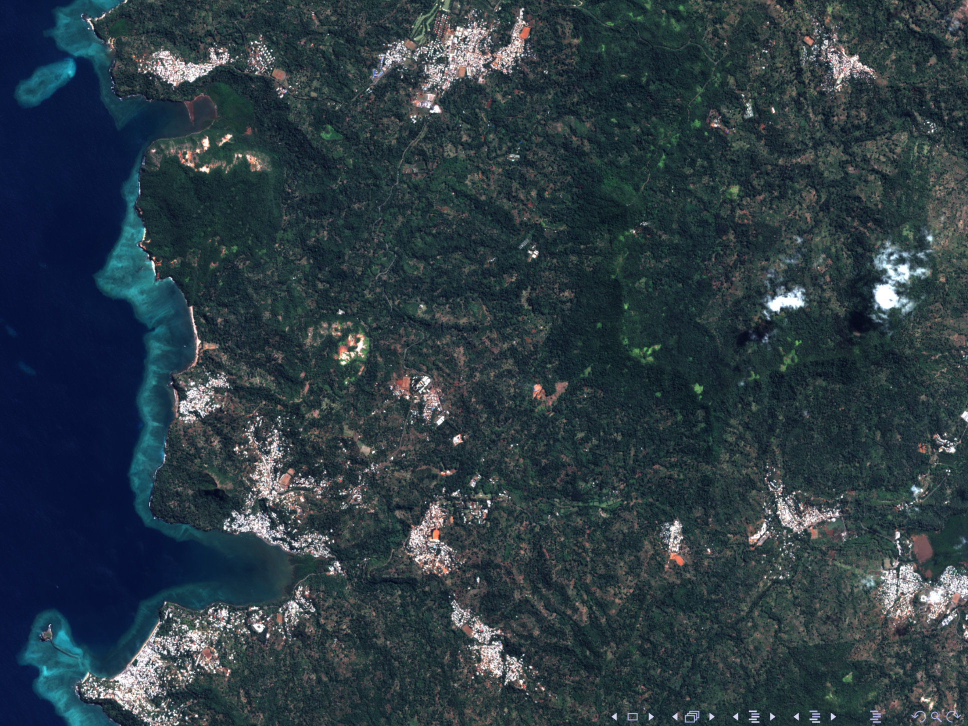
Incomplete list of OTB functions

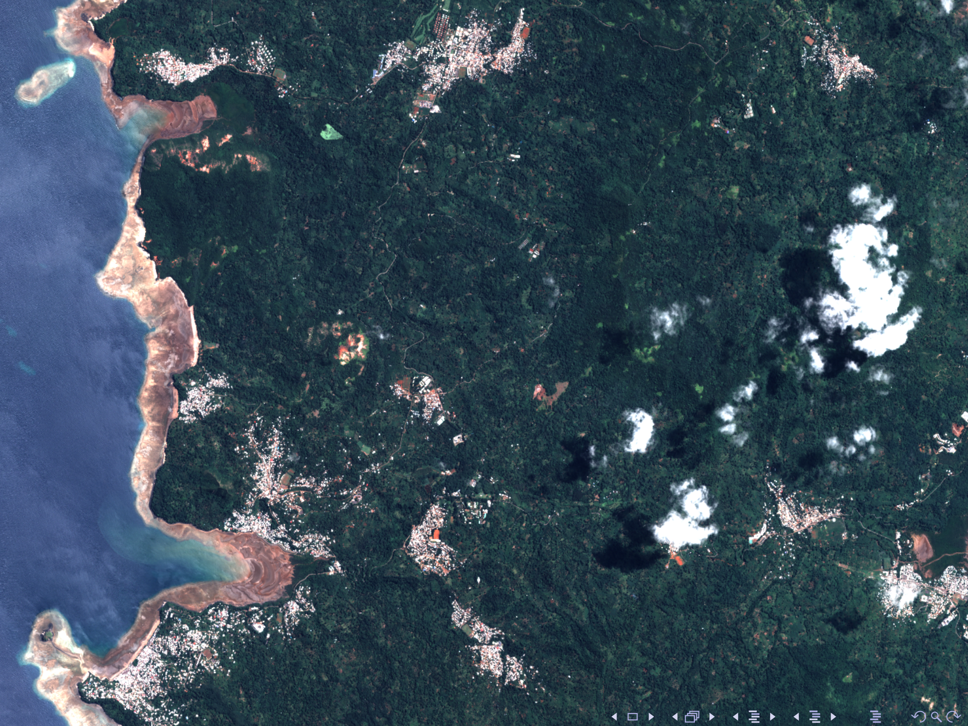
Segmentation

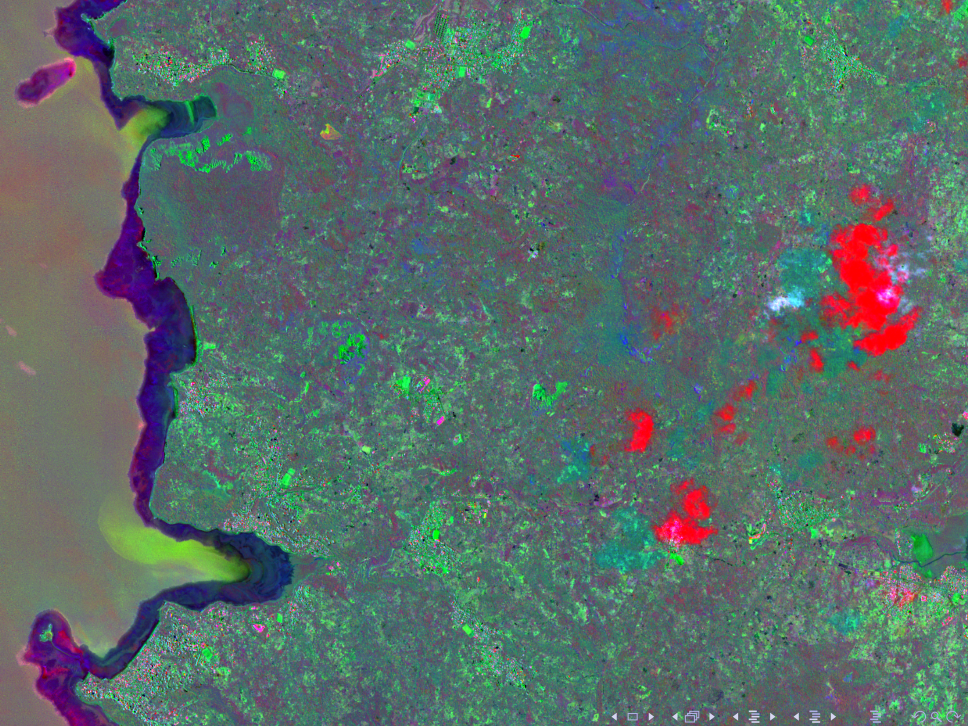
- Segmentation algorithms: Connected Components, MeanShift,Watershed. . .
- Methods to apply those algorithms on large dataset
- Vector or raster representation which allow Object Based Image Analysis

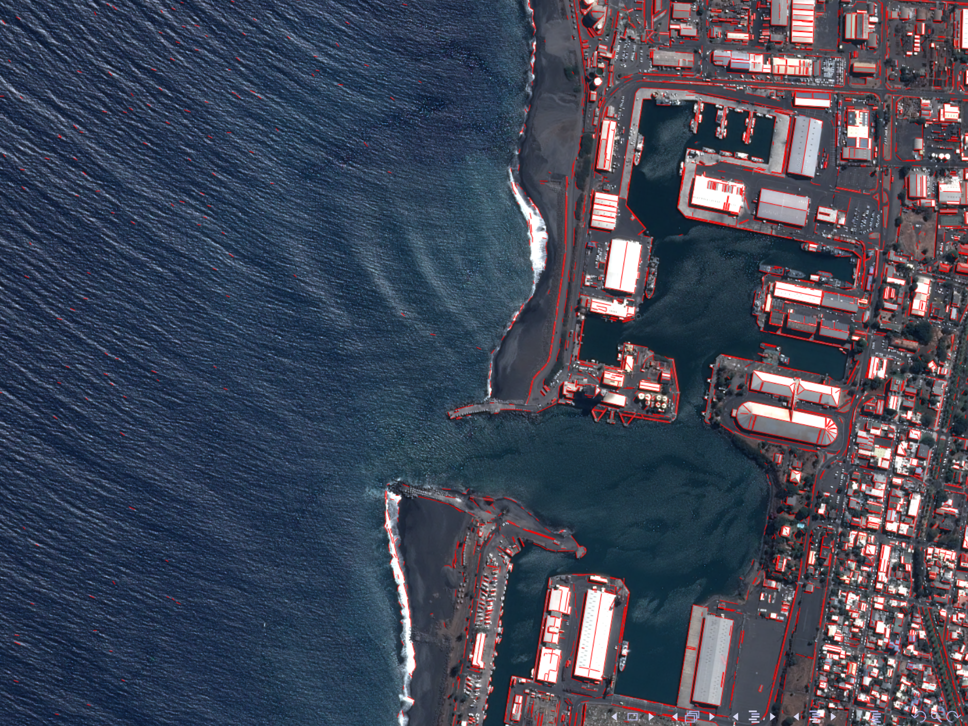
Classification

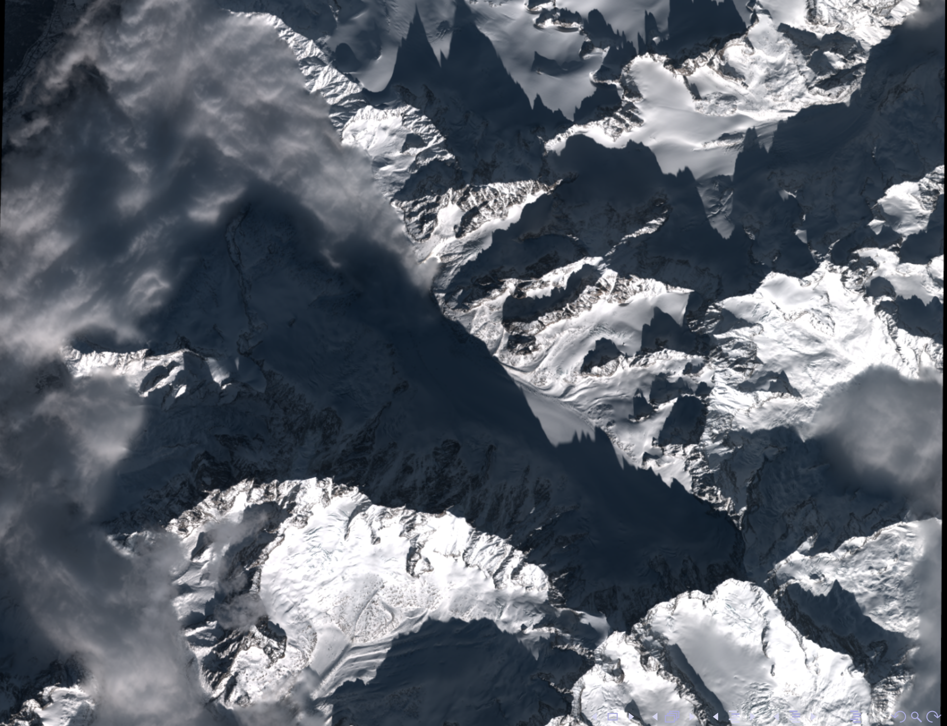
- 9 supervised methods available (including SVM and Random Forests)
- Fusion and regularization of classifications
- K-Means clustering or Kohonen maps
- Object classification (from a segmentation)

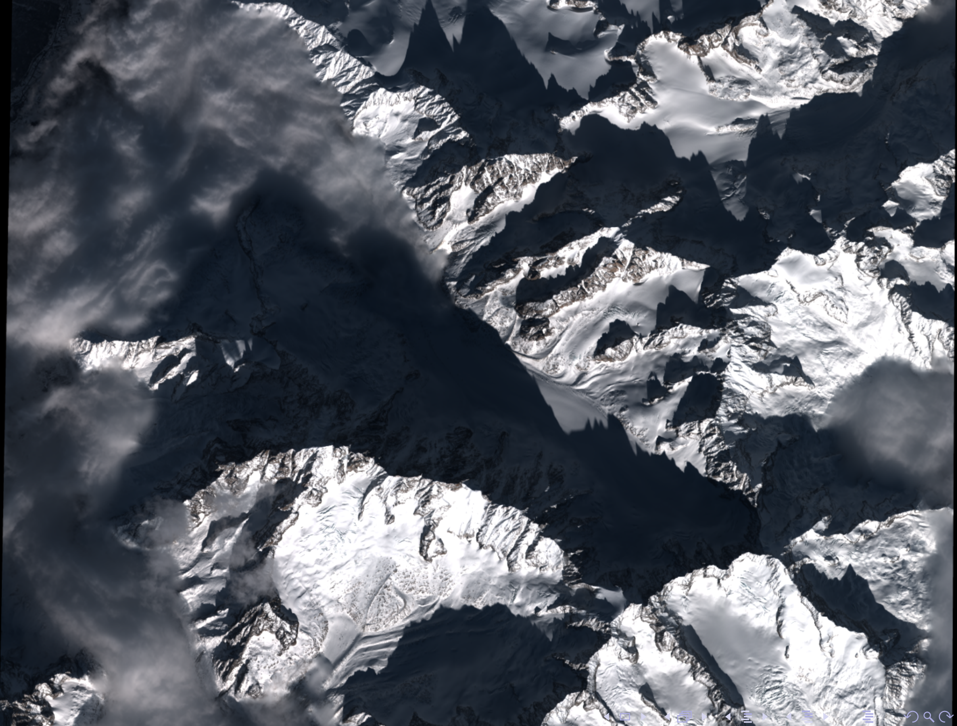


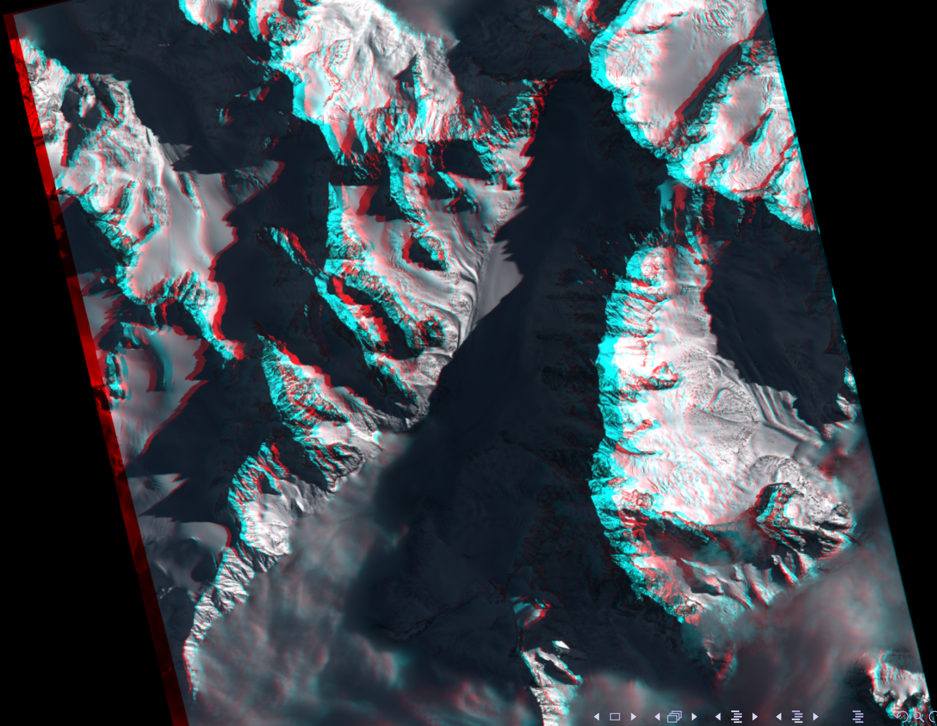












Outline

Functions and algorithms

Key characteristics

How to use OTB?

What's new in OTB?

Conclusion

Build on top of other open source image processing software

Motivations

- Interfaces seamlessly with other image processing and remote sensing open-source software
- Increase the number of functions
- Combine tools to create hybrid data pipeline

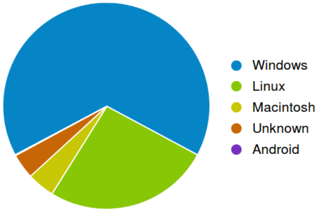
OTB backbone

- ITK: data processing pipeline
- GDAL: read and write raster and vector data
- OSSIM: sensor modelling and metadata support
- OpenCV and LibSVM: machine learning algorithms
- MuParser and MuParserX: powerful parsing of mathematical expression (band math)

Compatible (and available) on multiple platforms

Goal

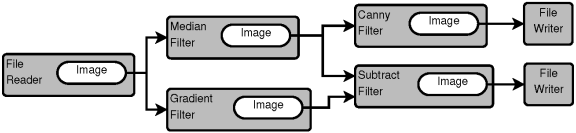
- Compile with recent versions of:
  - GCC
  - Clang
  - MinGW
  - Visual Studio. . .
- Binary packages available:
  - UbuntuGIS repository (GIS and IP software for Ubuntu)
  - Experimental Debian packages
  - Available in OSGeo4W (OSGeo tools on Windows)
  - Binary installers, Port and Brew formula for Mac OS X. . .



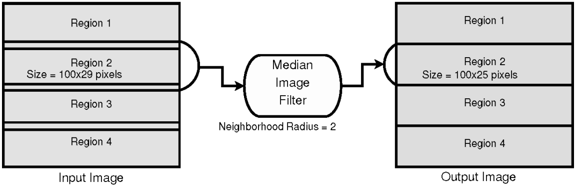
Number of OTB downloads on Sourceforge per Operating System

Flexibility, scalability: *Pipeline*, *Streaming* and *multithreading*

Pipeline data model

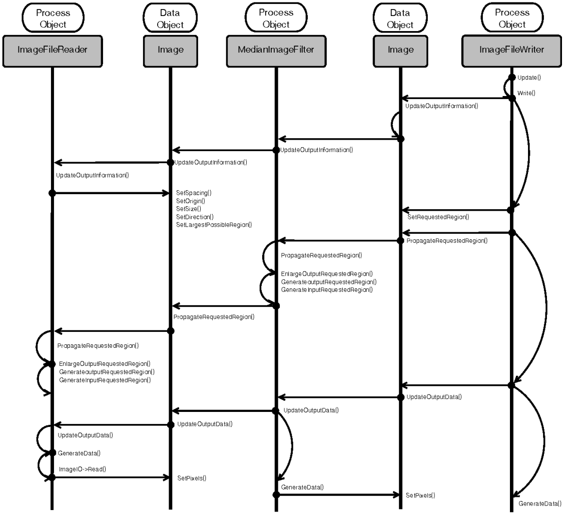


Streaming



source: `http://www.aosabook.org/en/itk.html`

Behind the scene


Process Object
Data Object
Process Object
Data Object
Process Object
ImageFileReader
Image
MedianImageFilter
Image
ImageFileWriter
Update()
Write()
UpdateOutputInformation()
UpdateOutputInformation()
UpdateOutputInformation()
UpdateOutputInformation()
SetSpacing()
SetOrigin()
SetSize()
SetDirection()
SetLargestPossibleRegion()
SetRequestedRegion()
PropagateRequestedRegion()
PropagateRequestedRegion()
EnlargeOutputRequestedRegion()
GenerateoutputRequestedRegion()
GenerateInputRequestedRegion()
PropagateRequestedRegion()
PropagateRequestedRegion()
EnlargeOutputRequestedRegion()
GenerateoutputRequestedRegion()
GenerateInputRequestedRegion()
UpdateOutputData()
UpdateOutputData()
UpdateOutputData()
UpdateOutputData()
GenerateData()
ImageIO->Read()
SetPixels()
GenerateData()
SetPixels()
GenerateData()

source: `http://www.aosabook.org/en/itk.html`

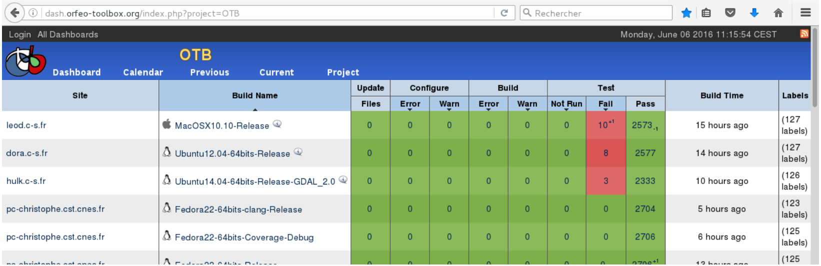
State of the art

- Try to keep track of up-to-date information about the latest developments, exchanging ideas, identifying future trends, and making networking
- Reference implementation of algorithms based on publications
- e.g.: morphological profile, MeanShift segmentation, Haralick textures, SURF keypoints. . .
- Reference implementation contributes by authors with their publications. e.g.: Large Scale MeanShift, object detection . . .

How is OTB developed?

- Distributed version control: Git (migration from Mercurial in July 2015)
- C++ and CMake (CTest, CDash)
- Test driven development (TDD)
- Agile (scrum)
- Continuous integration and packaging

Every day, almost 3000 tests are compiled, launched on 16 different configurations.



Monday, June 06 2016 11:15:54 CEST

Site	Build Name	Update	Configure		Build		Test			Build Time	Labels
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass		
leod.c-s.fr	MacOSX10.10-Release	0	0	0	0	0	0	10^{+1}	2573_{-1}	15 hours ago	(127 labels)
dora.c-s.fr	Ubuntu12.04-64bits-Release	0	0	0	0	0	0	8	2577	14 hours ago	(127 labels)
hulk.c-s.fr	Ubuntu14.04-64bits-Release-GDAL_2.0	0	0	0	0	0	0	3	2333	10 hours ago	(126 labels)
pc-christophe.cst.cnes.fr	Fedora22-64bits-clang-Release	0	0	0	0	0	0	0	2704	5 hours ago	(123 labels)
pc-christophe.cst.cnes.fr	Fedora22-64bits-Coverage-Debug	0	0	0	0	0	0	0	2706	6 hours ago	(125 labels)
pc-christophe.cst.cnes.fr	Fedora22-64bits-Release	0	0	0	0	0	0	0	2706^{+1}	13 hours ago	(125

Outline

Functions and algorithms

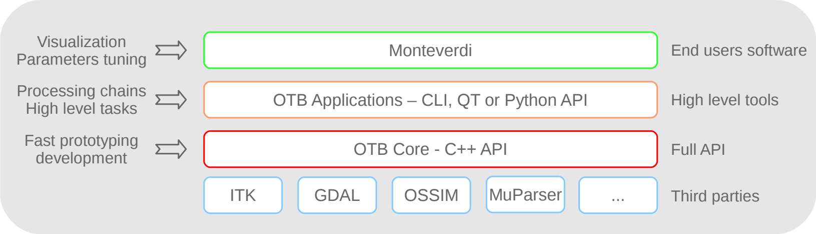
Key characteristics

How to use OTB?

What's new in OTB?

Conclusion

How to use OTB?



Visualization Parameters tuning ⟹ Monteverdi — End users software

Processing chains High level tasks ⟹ OTB Applications – CLI, QT or Python API — High level tools

Fast prototyping development ⟹ OTB Core - C++ API — Full API

ITK | GDAL | OSSIM | MuParser | ... — Third parties

Write your own code

Flexible, access to full API, requires C++ knowledge

Use the applications

High level functions (e.g. segmentation), callable from CLI, Qt, Python, can be extended

Use Monteverdi

Visualization, data management, Access to all applications

Show me the code!

```
#include "otbImage.h"
#include "otbImageFileReader.h"
#include "otbImageFileWriter.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 otb::ImageFileWriter<OutputImageType> WriterType;
  WriterType::Pointer writer = WriterType::New();

  writer->SetFileName(argv[2]);

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

  writer->Update();
}
```

The applications: write it once, use everywhere

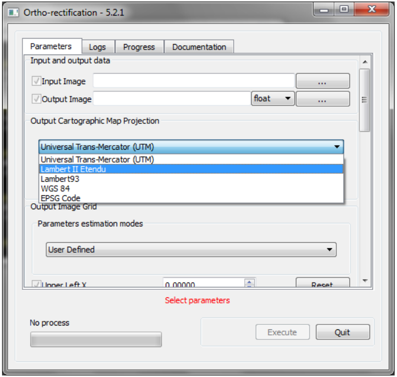
- 87 applications are shipped with OTB
- 1 application = 1 dynamic library (plugin)
- Applications are auto-descriptive and auto-documented
- Applications can be extended outside of OTB
- Several plugins players:
  - Command-line
  - Qt auto-generated
  - Python
- Applications are meant for integration in external systems

DisparityMapToElevationMap
ComputePolylineFeatureFromImage
Segmentation
BandMath
DSFuzzyModelEstimation
BundleToPerfectSensor
SplitImage
Rescale
GenerateRPCSensorModel
GeneratePlyFile
OSMDownloader
ConvertSensorToGeoPoint
KmzExport
Quicklook
BlockMatching
Smoothing
FusionOfClassifications
DimensionalityReduction
FineRegistration
KMeansClassification
RadiometricIndices
OrthoRectification
HaralickTextureExtraction
GridBasedImageResampling
ConcatenateImages
CompareImages
DownloadSRTMTiles
Pansharpening
LSMSVectorization
PixelValue
HomologousPointsExtraction
ImageEnvelope
ColorMapping
Convert
MultiResolutionPyramid
ImageClassifier
ExtractROI
SOMClassification
Quicklook
PixelValue
Rescale
Convert
ExtractROI
ComputeConfusionMatrix
Rasterization
ReadImageInfo
GrayScaleMorphologicalOperation
TileFusion
ClassificationMapRegularization
ConcatenateVectorData
ConnectedComponentSegmentation
LSMSSmallRegionsMerging
LSMSSegmentation
TestApplication
OpticalCalibration
BinaryMorphologicalOperation
HyperspectralUnmixing
MeanShiftSmoothing
KmzExport
LineSegmentDetection
EdgeExtraction
BandMath
RefineSensorModel
StereoFramework
LocalStatisticExtraction
Superimpose
ConvertCartoToGeoPoint
ComputeImagesStatistics

Applications: command-line invocation

```
$ otbcli_OrthoRectification

ERROR: Waiting for at least one parameter...
This is the OrthoRectification application, version 5.2.1
This application allows to ortho-rectify optical images from supported sensors.

Complete documentation: http://www.orfeo-toolbox.org/Applications/OrthoRectification.html

Parameters:
        -progress                <boolean>        Report progress
MISSING -io.in                   <string>         Input Image  (mandatory)
MISSING -io.out                  <string> [pixel] Output Image  [pixel=uint8/uint16/int16/uint32/int32/float/double] (default v
        -map                     <string>         Output Cartographic Map Projection [utm/lambert2/lambert93/wgs/epsg] (mandato
        -map.utm.zone            <int32>          Zone number  (mandatory, default value is 31)
        -map.utm.northhem        <boolean>        Northern Hemisphere  (optional, off by default)
        -map.epsg.code           <int32>          EPSG Code  (mandatory, default value is 4326)
        -outputs.mode            <string>         Parameters estimation modes [auto/autosize/autospacing/outputroi/orthofit] (m
MISSING -outputs.ulx             <float>          Upper Left X  (mandatory)
MISSING -outputs.uly             <float>          Upper Left Y  (mandatory)
MISSING -outputs.sizex           <int32>          Size X  (mandatory)
MISSING -outputs.sizey           <int32>          Size Y  (mandatory)
MISSING -outputs.spacingx        <float>          Pixel Size X  (mandatory)
MISSING -outputs.spacingy        <float>          Pixel Size Y  (mandatory)
        -outputs.lrx             <float>          Lower right X  (optional, off by default)
        -outputs.lry             <float>          Lower right Y  (optional, off by default)
        -outputs.ortho           <string>         Model ortho-image  (optional, off by default)
        -outputs.isotropic       <boolean>        Force isotropic spacing by default  (optional, on by default)
        -outputs.default         <float>          Default pixel value  (optional, on by default, default value is 0)
        -elev.dem                <string>         DEM directory  (optional, off by default)
        -elev.geoid              <string>         Geoid File  (optional, off by default)
        -elev.default            <float>          Default elevation  (mandatory, default value is 0)
        -interpolator            <string>         Interpolation [bco/nn/linear] (mandatory, default value is bco)
```

Applications: Graphical interface



Ortho-rectification - 5.2.1

Input and output data

Input Image

Output Image: float

Output Cartographic Map Projection: Universal Trans-Mercator (UTM)

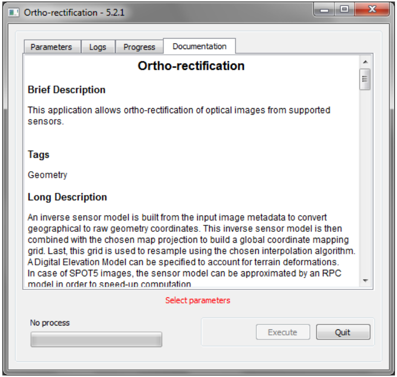
Output Image Grid

Parameters estimation modes: User Defined

Upper Left X: 0.00000

Select parameters

No process



Ortho-rectification - 5.2.1

Ortho-rectification

Brief Description

This application allows ortho-rectification of optical images from supported sensors.

Tags

Geometry

Long Description

An inverse sensor model is built from the input image metadata to convert geographical to raw geometry coordinates. This inverse sensor model is then combined with the chosen map projection to build a global coordinate mapping grid. Last, this grid is used to resample using the chosen interpolation algorithm. A Digital Elevation Model can be specified to account for terrain deformations.
In case of SPOT5 images, the sensor model can be approximated by an RPC model in order to speed-up computation

Select parameters

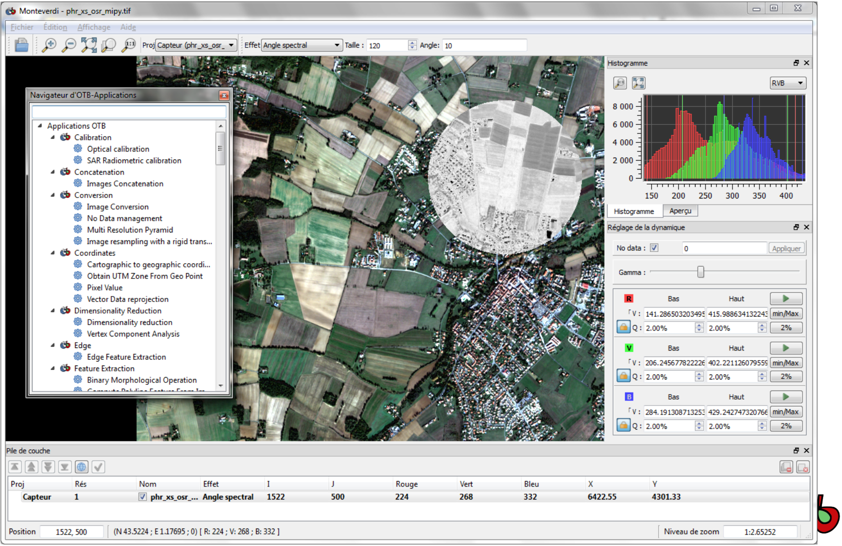
No process

Applications: Python interface

```
#!/usr/bin/python

# Import the otb applications package
import otbApplication

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

# The following lines set all the application parameters:
OrthoRectification.IO.IN = "QB_TOULOUSE_MUL_Extract_500_500.tif"
OrthoRectification.IO.OUT = "QB_Toulouse_ortho.tif"

app.MAP = ’epsg’
app.MAP.EPSG.CODE = 32768

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

CENTRE NATIONAL D’ÉTUDES SPATIALES

Monteverdi (acces to OTB applications)



Monteverdi - phr_xs_osr_mipy.tif

Proj: Capteur (phr_xs_osr_
Effet: Angle spectral
Taille : 120
Angle: 10

Navigateur d'OTB-Applications

- Applications OTB
  - Calibration
    - Optical calibration
    - SAR Radiometric calibration
  - Concatenation
    - Images Concatenation
  - Conversion
    - Image Conversion
    - No Data management
    - Multi Resolution Pyramid
    - Image resampling with a rigid trans...
  - Coordinates
    - Cartographic to geographic coordi...
    - Obtain UTM Zone From Geo Point
    - Pixel Value
    - Vector Data reprojection
  - Dimensionality Reduction
    - Dimensionality reduction
    - Vertex Component Analysis
  - Edge
    - Edge Feature Extraction
  - Feature Extraction
    - Binary Morphological Operation

Histogramme

Réglage de la dynamique

No data : ☑ 0

	Bas	Haut	
R			
V :	141.28650320349	415.98863413224	min/Max
Q :	2.00%	2.00%	2%
V			
V :	206.24567782222	402.22112607955	min/Max
Q :	2.00%	2.00%	2%
B			
V :	284.19130871325	429.24274732076	min/Max
Q :	2.00%	2.00%	2%

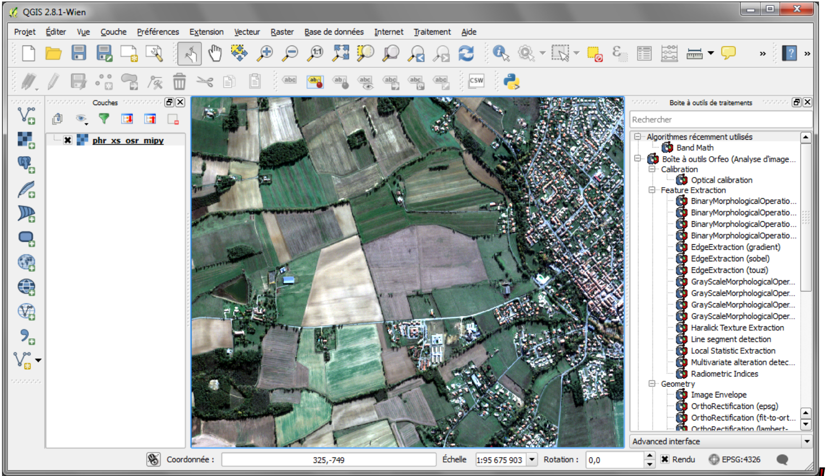
Pile de couche

Proj	Rés	Nom	Effet	I	J	Rouge	Vert	Bleu	X	Y
Capteur	1	☑ phr_xs_osr_...	Angle spectral	1522	500	224	268	332	6422.55	4301.33

Position 1522, 500 (N 43.5224 ; E 1.17695 ; 0) [R: 224 ; V: 268 ; B: 332]

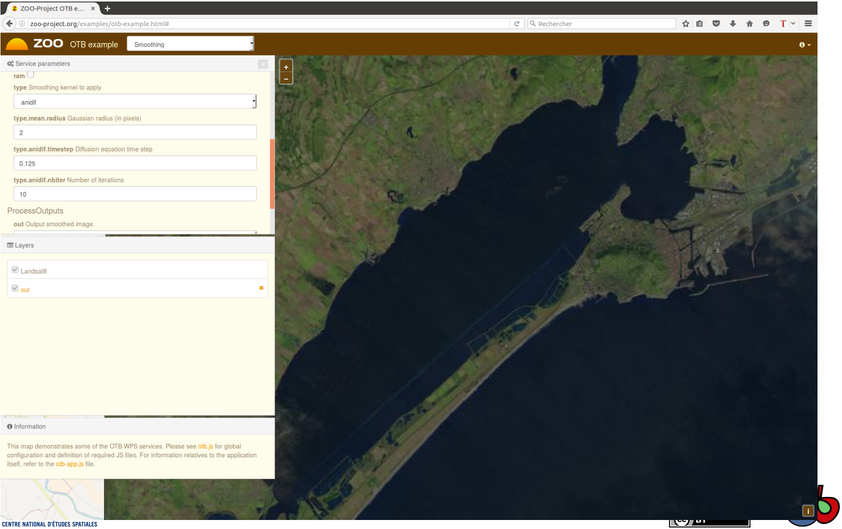
Niveau de zoom 1:2.65252

QGIS


QGIS 2.8.1-Wien
Projet
Éditer
Vue
Couche
Préférences
Extension
Vecteur
Raster
Base de données
Internet
Traitement
Aide
CSW
Couches
phr_xs_osr_mipy
Boite à outils de traitements
Rechercher
Algorithmes récemment utilisés
Band Math
Boîte à outils Orfeo (Analyse d'image...
Calibration
Optical calibration
Feature Extraction
BinaryMorphologicalOperatio...
BinaryMorphologicalOperatio...
BinaryMorphologicalOperatio...
BinaryMorphologicalOperatio...
EdgeExtraction (gradient)
EdgeExtraction (sobel)
EdgeExtraction (touzi)
GrayScaleMorphologicalOper...
GrayScaleMorphologicalOper...
GrayScaleMorphologicalOper...
GrayScaleMorphologicalOper...
Haralick Texture Extraction
Line segment detection
Local Statistic Extraction
Multivariate alteration detec...
Radiometric Indices
Geometry
Image Envelope
OrthoRectification (epsg)
OrthoRectification (fit-to-ort...
OrthoRectification (lambert-
Advanced interface
Rendu

Coordonnée : 325,-749
Échelle 1:95 675 903
Rotation : 0,0
EPSG:4326

OTB applications as ZOO WPS service



ram □

type Smoothing kernel to apply: anidif

type.mean.radius Gaussian radius (in pixels): 2

type.anidif.timestep Diffusion equation time step: 0.125

type.anidif.nbiter Number of iterations: 10

ProcessOutputs

out Output smoothed image.

This map demonstrates some of the OTB WPS services. Please see otb.js for global configuration and definition of required JS files. For information relatives to the application itself, refer to the otb-app.js file.

Outline

Functions and algorithms

Key characteristics

How to use OTB?

What's new in OTB?

Conclusion

5.0 (May 2015)

Make OTB more modular

- Better code layout, coherent modules (124 modules and 16 groups) with source, test and applications.
- Dependency management
- External contributions: `https://www.orfeo-toolbox.org/external-projects/`

SuperBuild

- No more third party software in OTB!
- The Superbuild downloads, configures, builds and installs dependencies
- Offline mode for compiling OTB without network access (e.g. airplane)

Open governance: Project Steering Committee

PSC beginning

- Until 2015: OTB is open-source software
- In march 2015: OTB become free software, with CNES as the first PSC

A club of developers, not managers

- High level project steering, roadmaps, communication and planning
- Vote RFCs: all members' votes have the same value (±1, ±0)
- Seats do not expire. Exits are by resignation or vote of expulsion
- The PSC is not a legal entity and has no funding

Numbers

- 5 members from 4 different organizations
- 2 releases under a PSC (5.2, 5.4)
- 3 online meetings (with public logs)

5.2 (December 2015)

OTB

- New SAR processing applications (polarimetry, radiometry, speckle)
- Support for Sentinel-1 products
- Better Python bindings
- Better GDAL 2.0 compatibility and support Sentinel-2 images
- Official package in DebianGIS (special thanks to Rashad and Debian maintainer)
- ...

Monteverdi 3.0

- Display an image mosaic or multi-temporal dataset
- Efficient visualization tools (local contrast, gradient. . .)
- Access to OTB applications

5.4 (May 2016)

OTB

- Switched to a fixed release schedule
- Merged Ice (visualization lib) into OTB
- External build of external modules
- New SAR decomposition methods: Barnes, Huynen, Pauli

Monteverdi 3.2

- Screen-shot feature
- Generate GDAL overviews
- Support for GDAL sub-datasets
- Added to the SuperBuild

5.6 (August 2016)

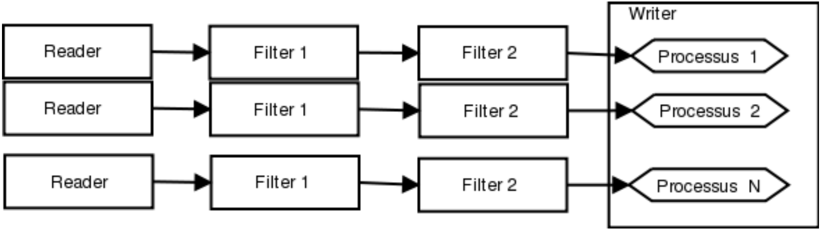
OTB

- MPI Image File Writer
- Samples extractor and selection for supervised classification
- Improve classification on vector
- Support for Sentinel-1 geometry (SAR sensor)

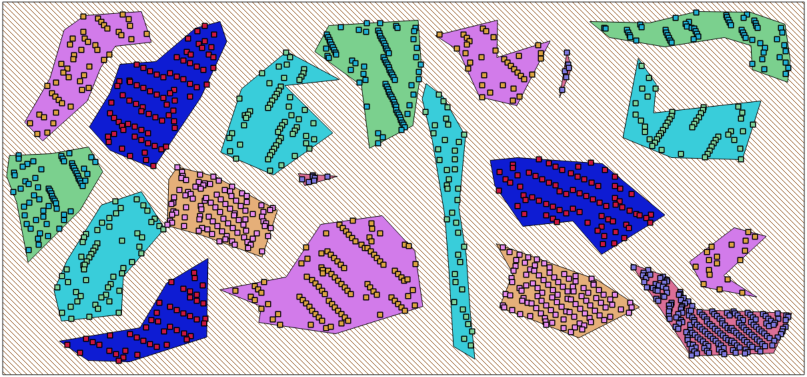
Monteverdi 3.4

- Improve OTB-applications display & search bar
- Open Sentinel-1 image (SLC product)

Parallel OTB pipeline with MPI



Polygons sampling strategy



Outline

Functions and algorithms

Key characteristics

How to use OTB?

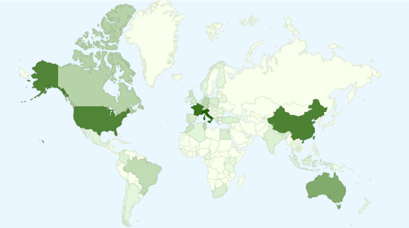
What's new in OTB?

Conclusion

How many users?

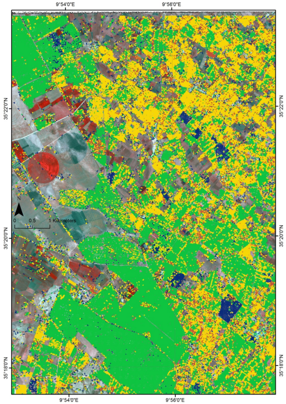
Hard to tell. . .

- ≈ 600 members on the otb-users list
- Between 100 and 150 mails by months
- ≈ 100 members on the developers list
- ≈ 118 user accounts on the bug tracker
- ≈ 50 contributors in the documentation
- ≈ 3400 downloads for OTB 5.0 on SourceForge(released June 1, 2015).



Success stories

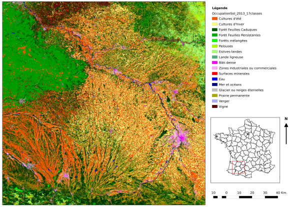
- OTB has been useful for (some) ORFEO users!
- Several training courses (3/5-day courses) given in France, Belgium, Madagascar, UNESCO, Hawaii,Finland. . .
- OTB has successfully processed 619 Pléiades images on RTU web site
- OTB provides many useful RS functions in **one single tool**
- OTB is/was the only open-source supporting PHR images (thanks to OpenJPEG)
- OTB equals or beats state-of-the-art tools (open source and maybe $$) on some points:
  - band calculator
  - tile-wise segmentation of full imagery
  - full scene classification with a range of machine learning algorithms
  - bridges between RS and GIS . . .
- Beyond Orfeo, OTB is already used in several projects and software
- OSGeo incubation



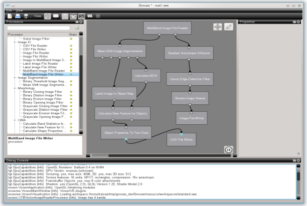
Thematic map from OTB segmentation, B. Mougenot - IRD

Projects and software using OTB

- OTB applications are available through QGIS processing framework
- OTB applications are available in Zoo Project (WPS service)
- OTB is a component of Sentinel-2 and Venus ground segment (CNES and ESA)
- Terr'Image: Educational software for satellite image analysis
- Use to prototype THEIA products from the Scientific Expertise Centres
- ESA Sentinel-2 for Agriculture
- Gnorasi Software (National Technical University of Athens)
- Geosud project(IRSTEA)
- TCM research program (ETS Quebec)



Prototype of THEIA Land cover product(CESBIO)


Gnorasi * - test1.wws
File
View
View
Processors
Processor
State
Sobel Image Filter
Image IO
CSV File Reader
CSV File Writer
Image File Reader
Image File Writer
Image to MultiBand Image C...
Label Image File Reader
Label Image File Writer
MultiBand Image File Reader
MultiBand Image File Writer
Image Segmentation
Binary Threshold Image Seg...
Mean Shift Image Segmenta...
Morphology
Binary Closing Image Filter
Binary Dilation Image Filter
Binary Erotion Image Filter
Binary Opening Image Filter
Grayscale Closing Image Filter
Grayscale Dilation Image Filter
Grayscale Erosion Image Fil...
Grayscale Opening Image F...
OBIA
Calculate Band Statistics fo...
Calculate New Feature for O...
Calculate Shape Properties ...
MultiBand Image File Writer
processor.
MultiBand Image File Reader
Mean Shift Image Segmentation
Gradient Anisotropic Diffusion
Calculate NDVI
Canny Edge Detection Filter
Label Image to Object Map
Simple Image Viewer
Calculate New Feature for Objects
Image File Writer
Object Properties To Text Data
CSV File Writer
Properties
Debug Console

```
tgt.GpuCapabilities (Info)  OpenGL Renderer: Gallium 0.4 on NV84
tgt.GpuCapabilities (Info)  GPU Vendor: nouveau (unknown)
tgt.GpuCapabilities (Info)  Texturing: yes, max size: 4096, 3D: yes, max 3D size: 512
tgt.GpuCapabilities (Info)  Texture features: 16 units, NPOT, rectangles, compression, 16x anisotropic
tgt.GpuCapabilities (Info)  Framebuffer Objects: yes, max 8 color attachments
tgt.GpuCapabilities (Info)  Shaders: yes (OpenGL 2.0), GLSL Version 1.20, Shader Model 2.0
voreen.VoreenApplication (Info)  OpenGL initializing modules
voreenve.VoreenMainWindow (Info)  VoreenVE plugins:
voreenve.VoreenVisualization (Info)  Loading workspace /home/kalxas/tmp/gnorasi_dev/Gnorasi/resource/workspaces/standard.vws
voreen.OTBVectorImageReaderProcessor (Info)  Image has 4 bands.
```

The Gnorasi software

Support/Help/Contribute

General resources

Site web	orfeo-toolbox.org
Wiki	wiki.orfeo-toolbox.org
Blog	blog.orfeo-toolbox.org

Documentation and help

Guides	Software Guide and CookBook (remote sensing recipes)
Doxygen	doxygen
Users mailing list	otb-users@googlegroups.com
Developers mailing list	otb-developers@googlegroups.com

Follow-up

Look at the code?	git.orfeo-toolbox.org
Find a bug?	bugs.orfeo-toolbox.org
Agile?	scrum.orfeo-toolbox.org
Weather?	dash.orfeo-toolbox.org

CENTRE NATIONAL D'ÉTUDES SPATIALES

Thank you! Any questions?