Orfeo ToolBox
www.orfeo-toolbox.org

OPEN-SOURCE REMOTE SENSING

Orfeo ToolBox in a nutshell
• It is an image processing library dedicated to remote sensing
• It is an open-source software under CeCill-v2 license (french equivalent to GPL)
• It is funded and developed by CNES in the frame of the ORFEO accompaniment program
• It is written in C++ on top of ITK (medical image processing)
• It interfaces seamlessly with other image processing and remote sensing open-source softwares, like GDAL or OSSIM
• Available on multiple platforms (Windows, Linux, Mac OS X)
• It is scalable thanks to parallel and on the flow processing

Thanks to its modular architecture, Orfeo ToolBox allows fast prototyping and development of processing chains.

By programming

Know a little about programming and want to access the full range of Orfeo ToolBox functions to build custom tools? Dive into our Software Guide and start writing your own C++ code with OTB.

A simple example of C++ OTB code

```cpp
#include "itkRescaleIntensityImageFilter.h"
#include "itkCannyEdgeDetectionImageFilter.h"
#include "otbStreamingImageFileWriter.h"
#include "otbImageFileReader.h"

void main(void)
{
  typedef double InputPixelType;
  typedef unsigned char OutputPixelType;
  typedef ImageFileReader<ImageType> ReaderType;
  typedef RescaleIntensityImageFilter<ImageType, OutputImageType> RescalerType;
  typedef CannyEdgeDetectionImageFilter<ImageType> FilterType;
  typedef StreamingImageFileWriter<OutputImageType> WriterType;

  ReaderType::Pointer reader = ReaderType::New();
  RescalerType::Pointer rescaler = RescalerType::New();
  FilterType::Pointer filter = FilterType::New();
  WriterType::Pointer writer = WriterType::New();

  reader->SetFileName(argv[1]);
  reader->Update();
  rescaler->SetInput(reader->GetOutput());
  rescaler->Update();
  filter->SetInput(rescaler->GetOutput());
  filter->Update();
  writer->SetInput(filter->GetOutput());
  writer->Update();

  return EXIT_SUCCESS;
}
```

How can I help?

We do not only need developers! To get involved, you can:
• Help us to validate algorithms and send feedback
• Provide OTB case studies (user stories)
• Help us to track bugs on bugs.orfeo-toolbox.org
• Provide new ideas and feature requests
• Spread the word!

What is Orfeo ToolBox?
What you can do with Orfeo ToolBox
• Read, write, convert, extract parts of your remote sensing data,
• Perform basic pre-processing like orthorectification, radiometric calibration or pansharpening,
• Perform common image processing tasks (thresholding, dimensionality reduction, Fourier or wavelets transform …)
• Extract features (radiometric indices, textures, shapes …),
• Segment images and vectorize segmentation results,
• Classify images in a supervised or unsupervised way,
• Perform object-based image analysis,
• Export results in Google Earth, Qgis and pretty print for publishing.

Getting help and support
• As an open-source software, OTB has its own users and developers community
• The development team provides support through mailing-list (otb-users@googlegroups.com) or IRC channel
• Features request can also be sent this way
• Want to keep in touch with the OTB? Bookmark our blog (blog.orfeo-toolbox.org)

What does Orfeo ToolBox look like?

Calling applications from command-line

```
$ otbcli_ImageSVMClassifier -in QB_1_ortho.tif -imstat cImageStatisticsQB1.xml -svm clsvmModelQB1.svm -out classification.png uchar
```

Calling applications from Qt interface

```
Monteverdi viewer

Monteverdi feature extraction module

Monteverdi segmentation module
```

Calling applications from python

```
# Running python
# Import the ctb 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("svm", "clsvmModelQB1.svm")
ImageSVMClassifier.SetParameterString("out", "classification.png")
# The following line executes the application
ImageSVMClassifier.ExecuteAndWriteOutput()
```