

# ORFEO Toolbox (OTB) Frequently Asked Questions (FAQ).

## Updated for OTB-5.10.0

OTB Development Team

February 13, 2017

### Contents

## 1 Introduction

### 1.1 What is OTB?

OTB, the ORFEO Toolbox is a library of image processing algorithms developed by CNES in the frame of the ORFEO Accompaniment Program. OTB is based on the medical image processing library ITK, <http://www.itk.org>, and offers particular functionalities for remote sensing image processing in general and for high spatial resolution images in particular.

OTB provides:

- image access: optimized read/write access for most of remote sensing image formats, meta-data access, simple visualization;
- sensor geometry: sensor models, cartographic projections;
- radiometry: atmospheric corrections, vegetation indices;
- filtering: blurring, denoising, enhancement;
- fusion: image pansharpening;
- feature extraction: interest points, alignments, lines;
- image segmentation: region growing, watershed, level sets;
- classification: K-means, SVM, Markov random fields;
- change detection.
- object based image analysis.
- geospatial analysis.

Many of these functionalities are provided by ITK and have been tested and documented for the use with remote sensing data.

You can get more information on OTB on the web at <http://www.orfeo-toolbox.org>.

## 1.2 What is ORFEO?

ORFEO stands for Optical and Radar Federated Earth Observation. In 2001 a cooperation program was set between France and Italy to develop ORFEO, an Earth observation dual system with metric resolution: Italy is in charge of COSMO-SkyMed the radar component development, and France of PLEIADES the optic component.

The PLEIADES optic component is composed of two "small satellites" (mass of one ton) offering a spatial resolution at nadir of 0.7 m and a field of view of 20 km. Their great agility enables a daily access all over the world, essentially for defense and civil security applications, and a coverage capacity necessary for the cartography kind of applications at scales better than those accessible to SPOT family satellites. Moreover, PLEIADES have stereoscopic acquisition capacity to meet the fine cartography needs, notably in urban regions, and to bring more information when used with aerial photography.

The ORFEO "targeted" acquisition capacities made it a system particularly adapted to defense or civil security missions, as well as critical geophysical phenomena survey such as volcanic eruptions, which require a priority use of the system resources.

With respect to the constraints of the Franco-Italian agreement, cooperation have been set up for the PLEIADES optical component with Sweden, Belgium, Spain and Austria.

### 1.2.1 Where can I get more information about ORFEO?

At the PLEIADES HR web site: <http://smc.cnes.fr/PLEIADES/>.

## 1.3 What is the ORFEO Accompaniment Program?

Beside the Pleiades (PHR) and Cosmo-SkyMed (CSK) systems developments forming ORFEO, the dual and bilateral system (France - Italy) for Earth Observation, the ORFEO Accompaniment Program was set up, to prepare, accompany and promote the use and the exploitation of the images derived from these sensors.

The creation of a preparatory program is needed because of :

- the new capabilities and performances of the ORFEO systems (optical and radar high resolution, access capability, data quality, possibility to acquire simultaneously in optic and radar),
- the implied need of new methodological developments : new processing methods, or adaptation of existing methods,
- the need to realize those new developments in very close cooperation with the final users, the integration of new products in their systems.

This program was initiated by CNES mid-2003 and will last until mid 2013. It consists in two parts, between which it is necessary to keep a strong interaction:

- A Methodological part,
- A Thematic part.

This Accompaniment Program uses simulated data (acquired during airborne campaigns) and satellite images quite similar to Pleiades (as QuickBird and Ikonos), used in a communal way on a set of special sites. The validation of specified products and services will be realized with Pleiades data

Apart from the initial cooperation with Italy, the ORFEO Accompaniment Program enlarged to Belgium, with integration of Belgian experts in the different WG as well as a participation to the methodological part.

### 1.3.1 Where can I get more information about the ORFEO Accompaniment Program?

Go to the following web site: [http://smsc.cnes.fr/PLEIADES/A\\_prog\\_accomp.htm](http://smsc.cnes.fr/PLEIADES/A_prog_accomp.htm).

### 1.4 Who is responsible for the OTB development?

The French Centre National d'Études Spatiales, CNES, initiated the ORFEO Toolbox and is responsible for the specification of the library. CNES funds the industrial development contracts and research contracts needed for the evolution of OTB.

## 2 License

### 2.1 Which is the OTB license?

OTB is distributed under a CeCILL free software license:

[http://www.cecill.info/licences/Licence\\_CeCILL\\_V2-en.html](http://www.cecill.info/licences/Licence_CeCILL_V2-en.html) which is recognized by the Free Software Foundation. It can be considered similar to the GNU GPL license.

### 2.2 Am I forced to distribute my code based on OTB?

No. The license gives you the option to distribute your application if you want to. You do not have to exercise this option in the license.

### 2.3 Am I forced to contribute my code based on OTB into the official repo?

No. The CeCILL license impose only to distribute the source of the application to your users.

### 2.4 If I wanted to distribute an application using OTB what license would I need to use?

The CeCILL license or the GNU GPL license.

### 2.5 I am a commercial user. Is there any restriction on the use of OTB?

OTB can be used internally ("in-house") without restriction, but only redistributed in other software that is under the CeCILL license. Moreover you need to distribute the source of your application to your users and only them.

## 3 Getting OTB

### 3.1 Who can download the OTB?

Anybody can download the OTB at no cost.

## 3.2 Where can I download the OTB?

Go to <http://www.orfeo-toolbox.org> and follow the "download OTB" link. You will have access to the OTB source code, to the Software User's Guide and to the Cookbook of the last release. Binary packages are also provided for the current version. OTB and Monteverdi are also integrated in OSGeo-Live since version 4.5. You can find more information about the project at <http://live.osgeo.org/>. Moreover you can find the last release of Monteverdi and OTB applications through the OSGeo4W installer.

## 3.3 How to get the latest bleeding-edge version?

You can get the current development version, as our repository is public, using Git (available at <http://git-scm.com>). Be aware that, even if the golden rule is *what is committed will compile*, this is not always the case. Changes are usually more than ten per day.

The first time, you can get the source code using:

```
git clone https://git@git.orfeo-toolbox.org/git/otb.git
```

Then you can build OTB as usual using this directory as the source (refer to build instructions). Later if you want to update your source, from the OTB source directory, just do:

```
git pull
```

A simple make in your OTB binary directory will be enough to update the library (recompiling only the necessary files).

# 4 Special issues about compiling OTB from source

All information about OTB compilation can be found into the related section. We present here only the special issues which can be encountered.

### 4.0.1 Debian Linux / Ubuntu

On some Debian and Ubuntu versions, the system GDAL library and its tiff internal symbol might conflict with the system Tiff library ([bugs.debian.org/558733](http://bugs.debian.org/558733)). This is most likely the case if you get odd segmentation fault whenever trying to open a tiff image. This symbol clash happens when using OTB. A workaround to the issue has been provided in GDAL sources, but is available in the 1.9.0 release.

The recommended procedure is to get this recent source and build GDAL from sources, with the following configure command:

```
./configure --prefix=INSTALL_DIR --with-libtiff=internal
            --with-geotiff=internal
            --with-rename-internal-libtiff-symbols=yes
            --with-rename-internal-libgeotiff-symbol=yes
```

### 4.0.2 Errors when compiling internal libkml

The internal version of libkml cannot be compiled when using an external build of ITK. See <http://bugs.orfeo-toolbox.org/view.php?id=879> for more details.

To workaroud the problem, either use an external build of libkml (it is provided on most systems), or use an internal build of ITK by setting to OFF the CMake variable `OTB_USE_EXTERNAL_ITK`.

#### 4.0.3 OTB compilation and Windows platform

To build OTB on Windows, we highly recommend using OSGeo4W which provides all the necessary dependencies.

Currently it is not possible to build OTB in Debug when using the dependencies provided by OSGeo4W. If you want to build OTB in Debug for Windows, you will need to build and install manually each dependency needed by OTB. You should use the same compiler for all the dependencies, as much as possible.

Therefore, we highly recommend you to use OSGeo4W shell environment to build OTB. You can use the 32 or 64 bit installer, since OSGeo4W provides all the necessary dependencies in the two cases. Please follow carefully the procedure provided in the Software Guide.

Typically, when using the dependencies provided by OSGeo4W, compile OTB in Release or RelWithDebInfo mode.

## 5 Using OTB

### 5.1 Where to start ?

OTB presents a large set of features and it is not always easy to start using it. After the installation, you can proceed to the tutorials (in the Software Guide). This should give you a quick overview of the possibilities of OTB and will teach you how to build your own programs. You can also found some information in the OTB Cookbook in which we provide some recipes about remote sensing with OTB.

### 5.2 What is the image size limitation of OTB ?

The maximum physical space a user can allocate depends on her platform. Therefore, image allocation in OTB is restricted by image dimension, size, pixel type and number of bands.

Fortunately, thanks to the streaming mechanism implemented within OTB's pipeline (actually ITK's), this limitation can be bypassed. The use of the `otb::ImageFileWriter` at the end of the pipeline, will seamlessly break the large, problematic data into small pieces whose allocation is possible. These pieces are processed one after the other, so that there is not allocation problem anymore. We are often working with images of  $25000 \times 25000$  pixels.

For the streaming to work, all the filters in the pipeline must be streaming capable (this is the case for most of the filters in OTB). The output image format also need to be streamable (not PNG or JPEG, but TIFF or ENVI formats, for instance).

The class `otb::ImageFileWriter` manage the steaming process following two strategies: by tile or by strip. Different size configuration for these two strategies are available into the interface. The default mode use the information about how the file is streamed on the disk and will try to minimize the memory consumption along the pipeline. More information can be found into the documentation of the class.

## 6 Getting help

### 6.1 Is there any mailing list?

Yes. There is a discussion group at <http://groups.google.com/group/otb-users/> where you can get help on the set up and the use of OTB.

## 6.2 Which is the main source of documentation?

The main source of documentation is the OTB Software Guide which can be downloaded at <http://www.orfeo-toolbox.org/packages/OTBSoftwareGuide.pdf>. It contains tenths of commented examples and a tutorial which should be a good starting point for any new OTB user. The code source for these examples is distributed with the toolbox. Another information source is the on-line API documentation which is available at <http://www.orfeo-toolbox.org/doxygen>.

You can also find some information about how to use Monteverdi and the OTB-Applications into the Cookbook at <http://www.orfeo-toolbox.org/CookBook/>.

## 7 Contributing to OTB

### 7.1 I want to contribute to OTB, where to begin?

There are many ways to join us in the OTB adventure. The more people contribute, the better the library is for everybody!

First, you can send an email to the user mailing list ([otb-users@googlegroups.com](mailto:otb-users@googlegroups.com)) to let us know what functionality you would like to introduce in OTB. If the functionality seems important for the OTB users, we will then discuss on how to retrieve your code, make the necessary adaptations, check with you that the results are correct and finally include it in the next release.

You can also run the nightly tests so we have a wider range of platforms to detect bugs earlier. Look at section ??.

You can also find more information about how to contribute at <https://www.orfeo-toolbox.org/community>

### 7.2 What are the benefits of contributing to OTB?

Besides the satisfaction of contributing to an open source project, we will include the references to relevant papers in the software guide. Having algorithms published in the form of reproducible research helps science move faster and encourages people who needs your algorithms to use them.

You will also benefit from the strengths of OTB: multi-platform, streaming and threading, etc.

### 7.3 What functionality can I contribute?

All functionalities which are useful for remote sensing data are of interest. As OTB is a library, it should be generic algorithms: change, detection, fusion, object detection, segmentation, interpolation, etc.

More specific applications can be contributed using the framework directly in the Applications directory of OTB.

## 8 Running the tests

### 8.1 What are the tests?

OTB is an ever changing library, it is quite active and have scores of changes per day from different people. It would be a headache to make sure that the brand new improvement that you introduced didn't break anything, if we didn't have automated tests. You also have to take into account differences in OS, compilers, options, versions of external libraries, etc. By running the tests and submitting it to the dashboard, you will help us detect problems and fix them early.

For each class, at minimum there is a test which tries to instantiate it and another one which uses the class. The output of each test (image, text file, binary file) is controlled against a baseline to make sure that the result hasn't changed.

All OTB tests source code are available in the directory `Testing` and are also good examples on how to use the different classes.

## 8.2 How to run the tests?

There is more than 2500 tests for OTB and it takes from 20 minutes to 3 hours to run all the test, mainly depending on your compilation options (Release mode does make a difference) and of course your hardware.

To run the tests, you first have to make sure that you set the option `BUILD_TESTING` to `ON` before building the library. If you want to modify it, just rerun `ccmake`, change the option, then make.

For some of the tests, you also need the test data and the baselines (see ??).

Once OTB is built with the tests, you just have to go to the binary directory where you built OTB and run `ctest -N` to have a list of all the tests. Just using `ctest` will run all the tests. To select a subset, you can do `ctest -R Kml` to run all tests related to kml files or `ctest -I 1,10` to run tests from 1 to 10.

## 8.3 How to get the test data?

Data used for the tests are also versioned using Git (see ??).

You can get the base doing:

```
git clone https://git@git.orfeo-toolbox.org/git/otb-data.git
```

This is about 1 GB of data, so it will take a while, but you have to do it only once, as after, a simple

```
git pull
```

will update you to the latest version of the repository.

You can also easily synchronize the directory you retrieve between different computers on your network, so you don't have to get it several times from the main server. Check out Git capabilities.

## 8.4 How to submit the results?

Once you know how to run the tests, you can also help us to detect the bugs or configuration problems specific to your configuration. As mentioned before, the possible combinations between OS, compiler, options, external libraries version is too big to be tested completely, but the more the better.

You just have to launch `ctest` with the `-D Experimental` switch. Hence:

```
ctest -D Experimental -A CMakeCache.txt
```

And you will be able to see the result at

<http://dash.orfeo-toolbox.org/Dashboard/index.php?project=OTB>.

If you are interested in setting up a nightly test (automatically launched every night), please contact us and we will give you the details.

## 9 OTB's Roadmap

### 9.1 Which will be the next version of OTB?

OTB's version numbers have 3 digits. The first one is for major versions, the second one is for minor versions and the last one is for bugfixes.

The first version was 1.0.0 in July 2006. The current one is 5.10.0.

#### 9.1.1 What is a major version?

A major version of the library implies the addition of high-level functionalities as for instance image registration, object recognition, etc.

#### 9.1.2 What is a minor version?

A minor version is released when low-level functionalities are available within one major functionality, as for instance a new change detector, a new feature extractor, etc.

#### 9.1.3 What is a bugfix version?

A bugfix version is released when significant bugs are identified and fixed.

### 9.2 When will the next version of OTB be available?

We plan to release two major new OTB version once a year. You can find some information into the roadmap section of the [wiki](#)

### 9.3 What features will the OTB include and when?

There is no detailed plan about the availability of OTB new features, since OTB's content depends on ongoing research work and on feedback from thematic users of the ORFEO Accompaniment Program.

Nevertheless, the main milestones for the OTB development are the following:

- Version 1 (2006):
  - core of the system,
  - IO,
  - basic filtering, segmentation and classification,
  - basic feature extraction,
  - basic change detection.
- Version 2 (2007):
  - geometric corrections,
  - radiometric corrections,
  - registration.
- Version 3 (2009):

- multi-scale and multi-resolution analysis,
  - object detection and recognition,
  - supervised learning.
- Version 3.X (2010-14):
  - data fusion,
  - spatial reasoning,
  - hyperspectral images analysis,
  - large scale segmentation,
  - stereo reconstruction
  - ...
- Version 4.X (2014 and later):
  - support of ITK 4.X (internal or external)
  - Clean up: migration of FLTK related code from OTB to Monteverdi, remove support for pqxx, gettext and LibLAS.
- Version 5.X (2015 and later):
  - Modularization
  - Support Spuperbuild mechanism to build OTB and all dependencies from source
  - No more internal versions of most third parties
  - Monteverdi as a module in OTB

You can find more information in the [RELEASE\\_NOTES.txt](#).