Evolutions between OTB 8 and version 9

OTB User days 2024

*Le monde est tel que nous le façonnons

The world is how we shape it*

a Sopra Steria company

Presentation overview

1. A modular organisation
2. Removal of graphical interface dependencies (Qt)
3. Going Full cmake
4. Revamped Packaging method
5. Docker images
6. Deprecations (MacOS Intel, libKML, Ubuntu 18)

01

A Modular Organisation

The need of a modular organisation

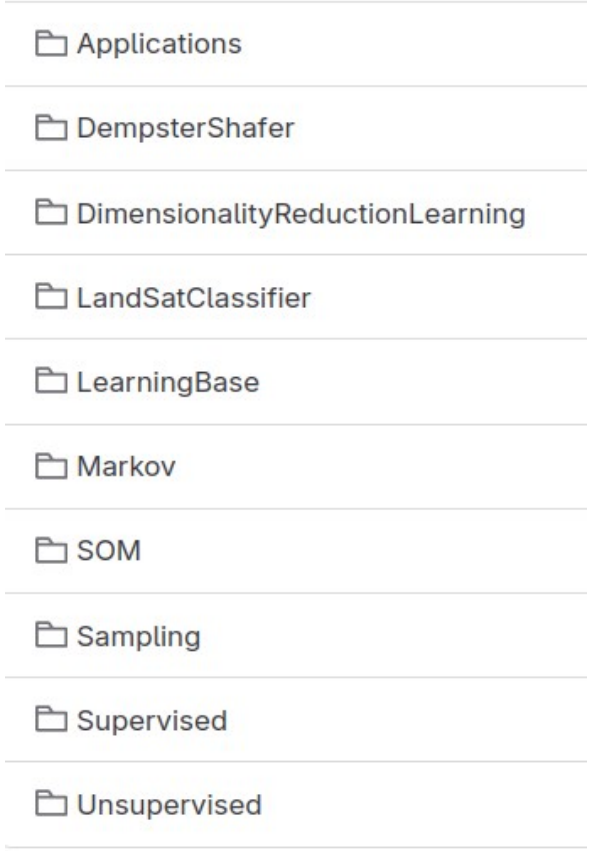
- A majority of users use **around 5 % or less** of all available applications (115)
- **Simplify installation process** / reduce installed size
- Provide an **easier interface** for remote modules and processing chains
- **Reduce compilation time** : OTB 8 takes ~ 45min on an Intel Core I5-8250U

Old vs New Modules organisation

- Adapters
- Applications
- Core
- Detection/CloudDetection
- Feature
- Filtering
- Fusion
- Hyperspectral
- IO
- Learning
- MPI
- Radiometry
- Registration
- Remote
- Segmentation
- ThirdParty
- Visualization
- Wrappers

- Core
- FeaturesExtraction
- Hyperspectral
- Learning
- Miscellaneous
- Remote
- SAR
- Segmentation
- StereoProcessing
- ThirdParty

In-module organisation



- One set of applications per module
- Sub modules clearly identified
- Every sub module contains
    - Sources
    - Include
    - tests

HMI Components removal

- Removal of Qt (**~150Mo gain** in install size) and **15 min compilation time gain**
- Removal of all graphics related libraries (GLUT, Glew…) 5 dependencies no longer need to be compiled
- **Replaced by the official QGIS plugin**

02

Going full Cmake

CMake

- State in OTB 8 :
    - A mix of Makeself/pkgconfig/cmake
    - A lot of cmake macros redefined for packaging (15 files)
- Advantages of a full cmake approach
- Cmake code modernization
- A lot easier for packaging / modules

CMake

- Clear build target definition per module :
    - **OTB_Build_Core**
    - **OTB_Build_ModuleName**
- **Automatic switch of dependencies** build when activating OTB_Build_module
- Per module build possible, also with packaging

Cmake improvements

50%

Compilation time gain

- 12 vs 17 dependencies to build
- Per target compilation

20%

Time gain on packaging process

Makeself was not efficient in packaging

40%

CI Time gain

An average full pipeline for release was 1h30

Now less than an hour

03

New Packaging method

Packaging

- State in OTB 8 :
    - Package format : .run for Linux and Macos, .zip for Windows (~200Mo each package) Installed size : 850 Mo
- One package vs modular packages
    - **One .tar.gz for each platform** containing all modules and dependencies (~115 Mo for one package) Installed size : 510 Mo
    - **OTB-Core + OTB-Dependencies** minimal install (~87 Mo split in two packages) : Installed size 440 Mo
- Possibility to install modules **on top** of the current Core+Dep install

Package Installation simplified

- **One click install** => Extract tar gz Here
- First source of otbenv.profile : automatic detection of python version and download of gdal bindings for this version of python => « from osgeo import gdal » works
- OTB python **bindings recompilation made easier**
- **Movable installation**

CI evolutions

- Default build platform :
  - Ubuntu 18.04 => Ubuntu 20.04
  - CentOS 7 => RedHat 8
  - Windows 10
  - ~~MacOS Intel~~
- Current Packages Compatiblity :
  - Ubuntu 20,22,24 / Debian 11 and 12 : Python 3.8 to 3.12
  - RedHat 8 : Python 3.8
  - Windows 10/11 : Python 3.10

Deprecations

- LibKML not updated since 2015
- MacOS intel platform, OTB now works for MacOS ARM platform via the docker image (soon in macports)
- Ubuntu 18 support ended in 2023, the main build is now done on Ubuntu 20.04
- CentOS 7 deprecated => RedHat 8
- Some OTB applications have been moved to « Miscellaneous » module, for deprecation

04

Docker images

Docker images

- Automatic generation and push to **DockerHub** of docker images for **release branches**
- Base image on Ubuntu 20.04 / Python 3.8 bindings
- Ubuntu 22 and Ubuntu 24 images to have a working python 3.10/python 3.12 ready images.
- « docker pull orfeotoolbox/otb:9.1.0 »
- MacOS users have to use this docker image to continue using OTB

Q&A