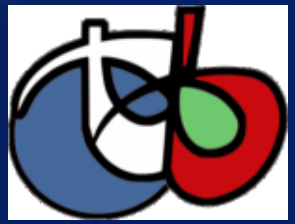
UFAZ – OTB Workshop 2021-02-22/23

Your first steps with Orfeo Toolbox :



How to write an OTB processing chain within a Jupyter Notebook ?

Emmanuelle SARRAZIN – emmanuelle.sarrazin@cnes.fr
Yannick TANGUY – yannick.tanguy@cnes.fr

cnes
CENTRE NATIONAL D'ÉTUDES SPATIALES

OTB Guided Tour overview

Overview of Orfeo ToolBox

Introduction to SENTINEL 2 images

Get familiar with Monteverdi & OTB applications

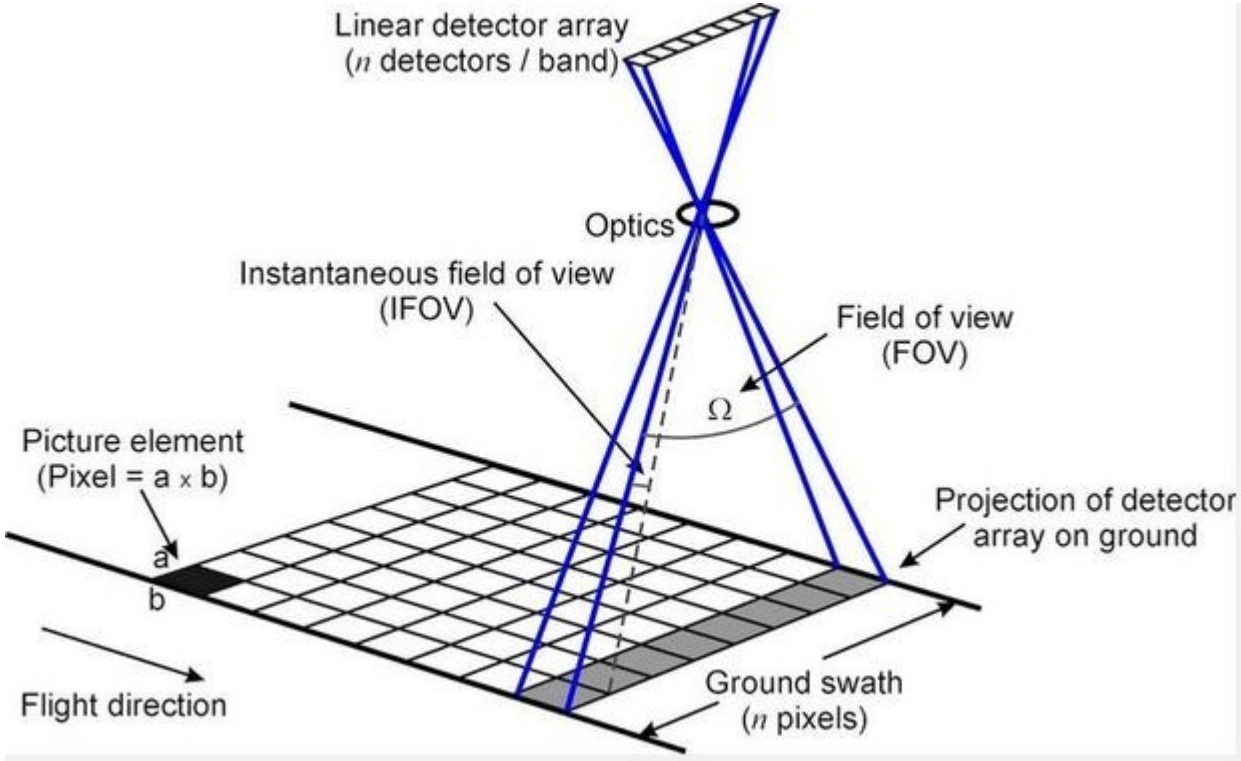
Processing chain with OTB in Python, within a Jupyter Notebook

A few words about satellite imagery

Digital camera sensor

10 M pixels

Satellite sensor (linear detector array)
(~ 30 000 pixels)



Introduction to SENTINEL 2 images

Part of COPERNICUS European program (Sentinel 1 – Radar, Sentinel 3 – Altimetry, Sentinel 4, Sentinel 5 – atmosphere, etc.)

Sentinel 2 : optical imagery, wide swath, multi-band imagery

- **Two satellites (Sentinel 2A, 2B)**
  - Low earth orbit (~ 788 km)
  - 5 days revisit
  - Swath (field of view) : 290 km
  - Systematic global coverage of land surfaces from 56° S to 84° N
  - 10 or 20 m ground resolution

Introduction to SENTINEL 2 images

- 13 bands from visible to infra-red

Sentinel-2 bands	Central wavelength (nm)	Bandwidth (nm)	Spatial resolution (m)
Band 1 – Coastal aerosol	442.7	21	60
Band 2 – Blue	492.4	66	10
Band 3 – Green	559.8	36	10
Band 4 – Red	664.6	31	10
Band 5 – Vegetation red edge	704.1	15	20
Band 6 – Vegetation red edge	740.5	15	20
Band 7 – Vegetation red edge	782.8	20	20
Band 8 – NIR	832.8	106	10
Band 8A – Narrow NIR	864.7	21	20
Band 9 – Water vapour	945.1	20	60
Band 10 – SWIR – Cirrus	1373.5	31	60
Band 11 – SWIR	1613.7	91	20
Band 12 – SWIR	2202.4	175	20

Introduction to SENTINEL 2 images

- **Data processing :**
  - L1C : geometric and radiometric corrections (Top of atmosphere reflectance), georeferenced products. Download : https://peps.cnes.fr (worldwide coverage)
  - L2A : atmospheric corrections, slope corrections (Top of canopy reflectance). Download : https://theia.cnes.fr (mainly France)
  - Each product is a tile of 100 km x 100 km. ~ 800 Mb per tile
  - More information here : https://sentinel.esa.int/web/sentinel/missions/sentinel-2/data-products

OTB Guided Tour overview

Overview of Orfeo ToolBox

Introduction to SENTINEL 2 images

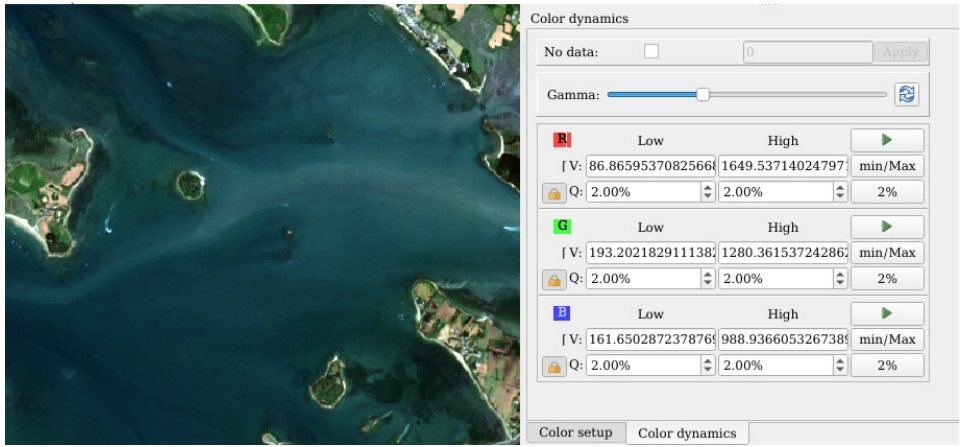
Get familiar with Monteverdi & OTB applications

Processing chain with OTB in Python, within a Jupyter Notebook

Get familiar with Monteverdi and OTB applications

- **Know how to display an image and set up rendering in Monteverdi,**
- **Know how to display an image stack in Monteverdi,**
- **Launch an OTB application**

- Open a terminal and launch Monteverdi
  - $> monteverdi &
- Sentinel2 extracts are in ~/otb-guided-tour/data
  - Open the image of the 2018-06-21
  - Open the image of the 2018-07-06

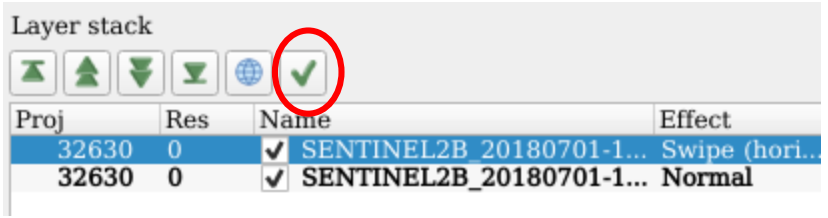


Get familiar with Monteverdi and OTB applications

- Try the different “Layer effects” on the bottom layer

- Scroll between layers (Ctrl + mouse-wheel)
- Switch between two layers (right click)

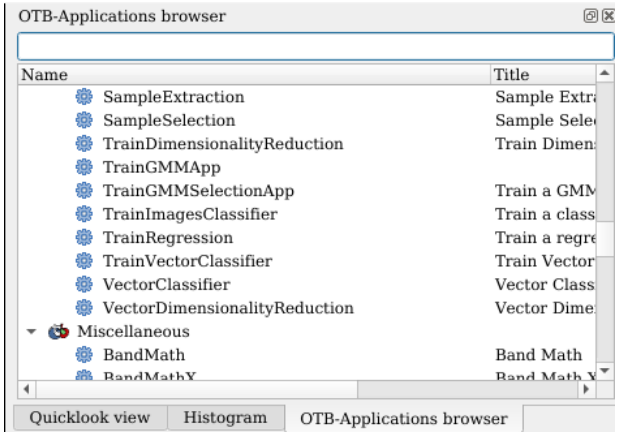
- Apply the display settings

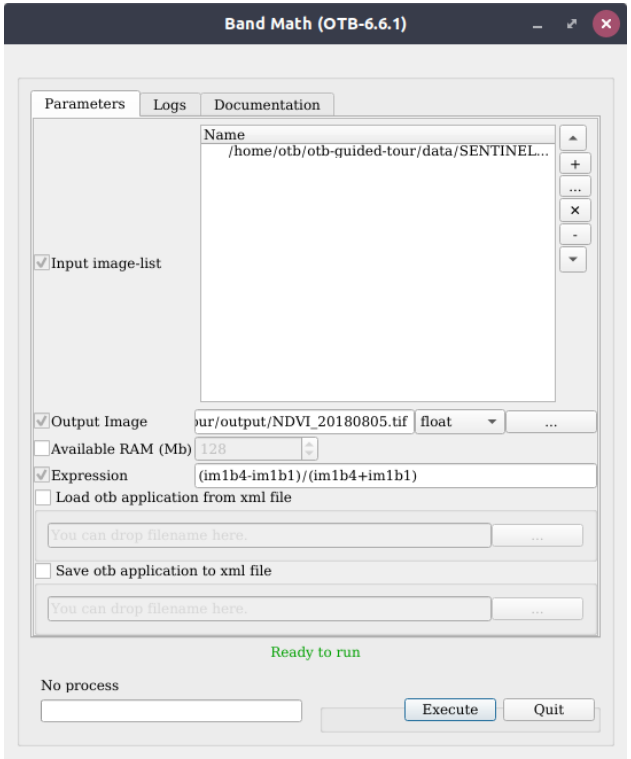


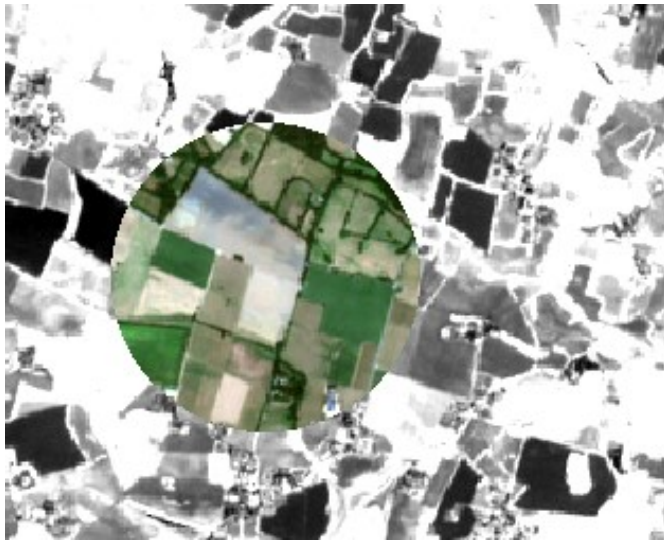
- Change band selection to make a false-color composite
  - Red : band 4 (Near InfraRed)
  - Green : band 3 (Red)
  - Blue : band 2 (Green)

Get familiar with Monteverdi and OTB applications

- Display list of applications
- Launch an application
- Compute NDVI







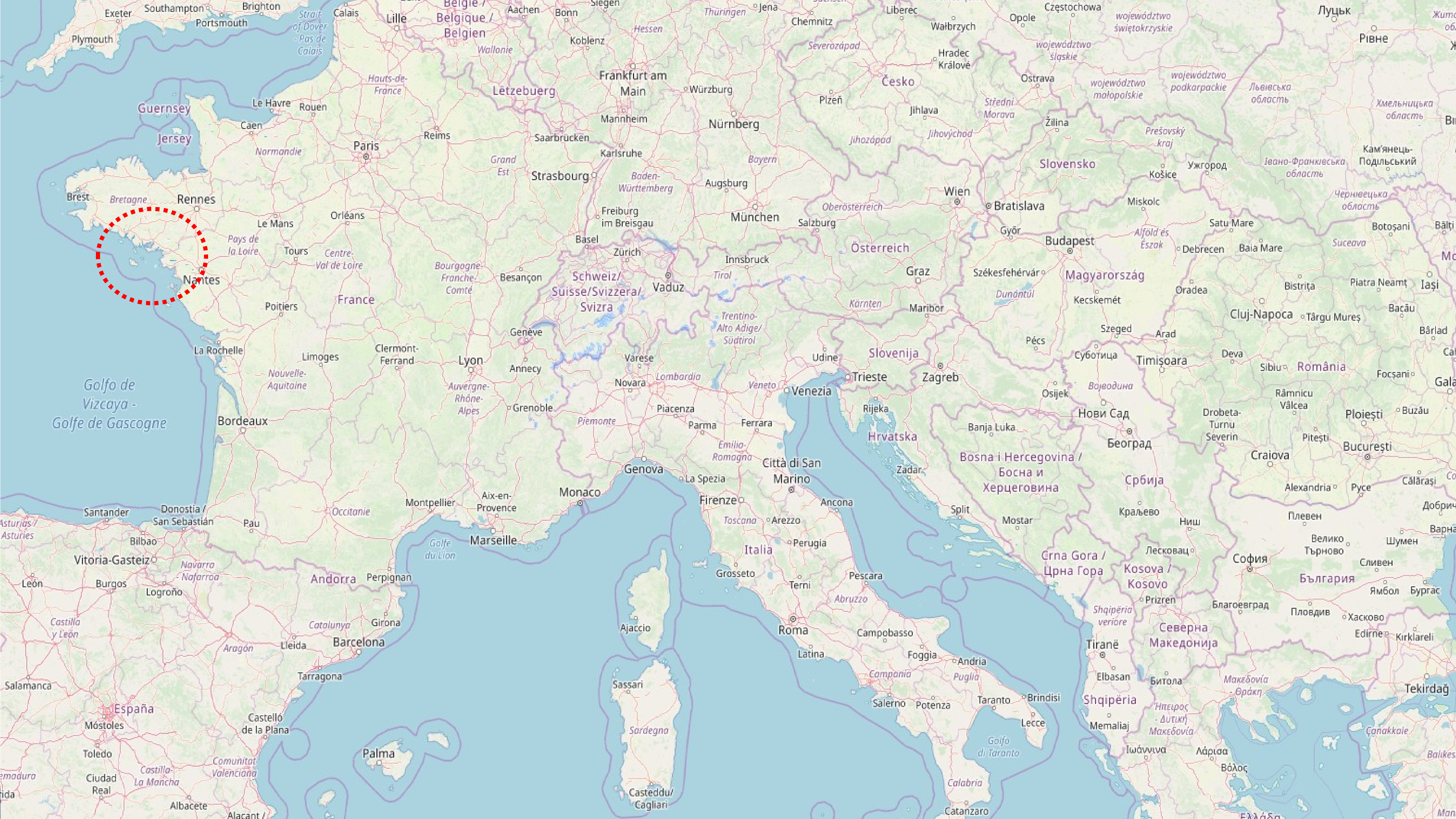
OTB Guided Tour overview

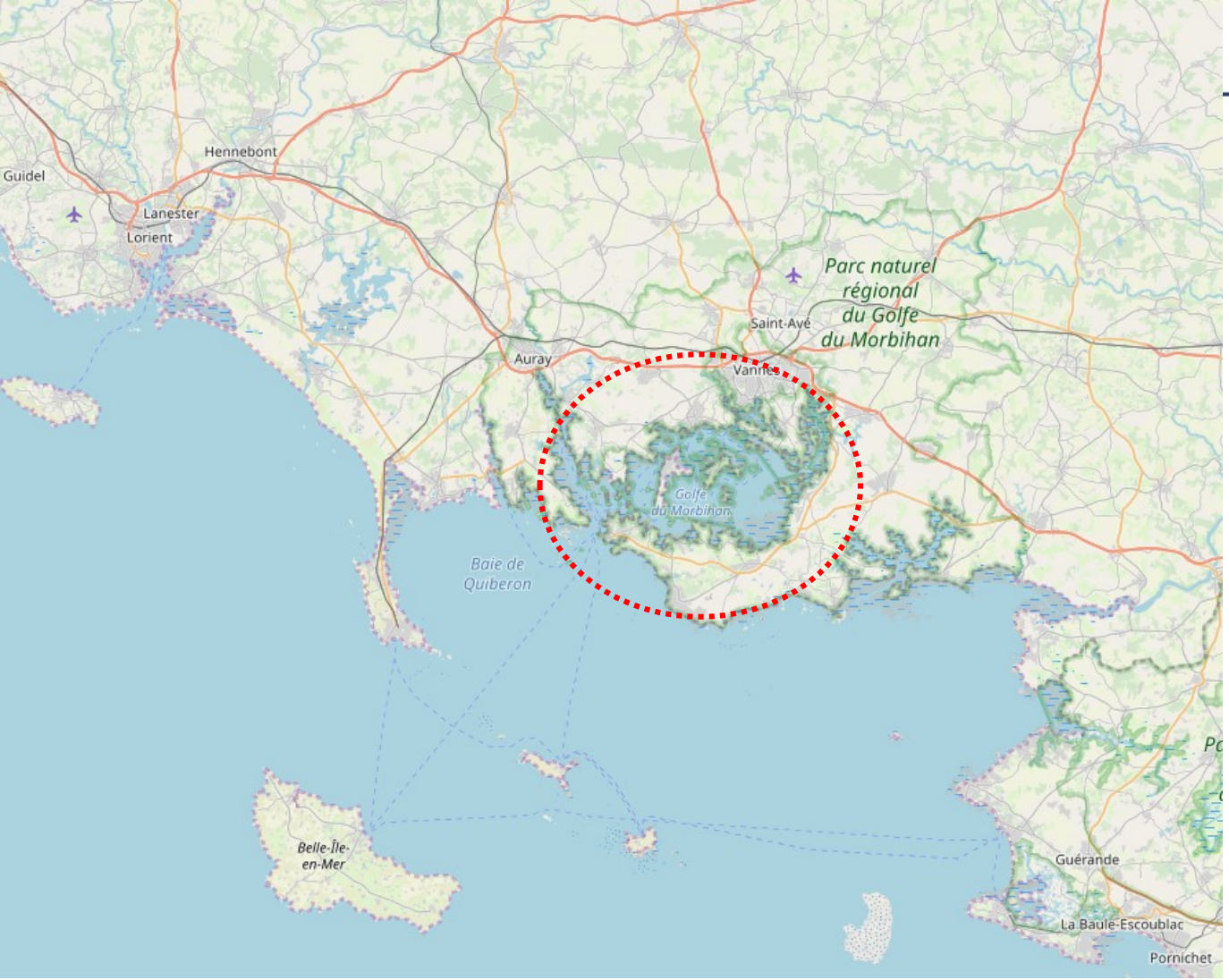
Overview of Orfeo ToolBox

Introduction to SENTINEL 2 images

Get familiar with Monteverdi & OTB applications

Processing chain with OTB in Python, within a Jupyter Notebook


Exeter
Southampton
Brighton
Plymouth
Portsmouth
Strait of Dover - Pas de Calais
Calais
Lille
Belgïe / Belgique / Belgien
Wallonie
Aachen
Bonn
Siegen
Thüringen
Jena
Chemnitz
Liberec
Częstochowa
Opole
Wałbrzych
Hessen
Koblenz
Severozápad
Hradec Králové
województwo świętokrzyskie
województwo śląskie
Луцьк
Рівне
Guernsey
Jersey
Le Havre
Rouen
Caen
Hauts-de-France
Lëtzebuerg
Frankfurt am Main
Würzburg
Česko
Plzeň
Ostrava
województwo małopolskie
województwo podkarpackie
Львівська область
Хмельницька область
Normandie
Paris
Reims
Saarbrücken
Mannheim
Nürnberg
Jihlava
Střední Morava
Žilina
Karlsruhe
Jihočeský
Jihozápad
Prešovský kraj
Bayern
Grand Est
Strasbourg
Baden-Württemberg
Augsburg
Slovensko
Košice
Ужгород
Івано-Франківська область
Кам'янець-Подільський
Brest
Bretagne
Rennes
Orléans
Le Mans
Freiburg im Breisgau
München
Oberösterreich
Salzburg
Wien
Bratislava
Miskolc
Чернівецька область
Pays de la Loire
Tours
Centre-Val de Loire
Basel
Zürich
Innsbruck
Győr
Budapest
Alföld és Észak
Satu Mare
Debrecen
Baia Mare
Botoșani
Suceava
Nantes
Bourgogne-Franche-Comté
Besançon
Schweiz/ Suisse/Svizzera/ Svizra
Vaduz
Tirol
Österreich
Graz
Székesfehérvár
Magyarország
Oradea
Bistrița
Piatra Neamț
Iași
Poitiers
France
Kärnten
Maribor
Dunántúl
Kecskemét
Cluj-Napoca
Târgu Mureș
Bacău
Genève
Trentino-Alto Adige/ Südtirol
Szeged
Arad
Bârlad
La Rochelle
Limoges
Clermont-Ferrand
Lyon
Annecy
Varese
Udine
Slovenija
Zagreb
Pécs
Суботица
Timișoara
Deva
Sibiu
România
Focșani
Trieste
Golfo de Vizcaya - Golfe de Gascogne
Nouvelle-Aquitaine
Auvergne-Rhône-Alpes
Novara
Lombardia
Veneto
Venezia
Osijek
Војводина
Rijeka
Rămnicu Vâlcea
Grenoble
Piacenza
Нови Сад
Drobeta-Turnu Severin
Ploiești
Buzău
Bordeaux
Piemonte
Parma
Ferrara
Banja Luka
Hrvatska
Emilia-Romagna
Београд
Pitești
București
Città di San Marino
Genova
Bosna i Hercegovina / Босна и Херцеговина
Craiova
Zadar
La Spezia
Србија
Alexandria
Русе
Călărași
Monaco
Aix-en-Provence
Montpellier
Firenze
Ancona
Santander
Donostia / San Sebastián
Occitanie
Split
Крагујевац
Плевен
Добрич
Pau
Toscana
Arezzo
Mostar
Ниш
Bilbao
Marseille
Golfe du Lion
Perugia
Велико Търново
Шумен
Варна
Vitoria-Gasteiz
Italia
Лесковац
Crna Gora / Црна Гора
София
Сливен
Navarra - Nafarroa
Andorra
Perpignan
Grosseto
Pescara
Kosova / Kosovo
България
León
Burgos
Logroño
Terni
Prizren
Благоевград
Ямбол
Бургас
Abruzzo
Shqipëria veriore
Castilla y León
Catalunya
Girona
Ajaccio
Roma
Campobasso
Северна Македонија
Пловдив
Хасково
Edirne
Kırklareli
Aragón
Lleida
Barcelona
Tiranë
Latina
Foggia
Andria
Sassari
Campania
Puglia
Elbasan
Битола
Μακεδονία - Θράκη
Salamanca
Tarragona
Taranto
Brindisi
Tekirdağ
España
Salerno
Potenza
Shqipëria
Ήπειρος - Δυτική Μακεδονία
Móstoles
Castelló de la Plana
Lecce
Memaliaj
Sardegna
Çanakkale
Toledo
Palma
Golfo di Taranto
Ιωάννινα
Λάρισα
Comunitat Valenciana
Castilla-La Mancha
Βόλος
Ciudad Real
Calabria
Albacete
Casteddu/ Cagliari
Catanzaro



"Legend says that there are as many islands in the Gulf as there are days of the year.
However, this is untrue and the gulf has about 40, depending on the tide." (Wikipedia)

Let's count the Islands with Orfeo ToolBox and a few Sentinel 2 images !

Write a processing chain with OTB in Python

Goals of this session :

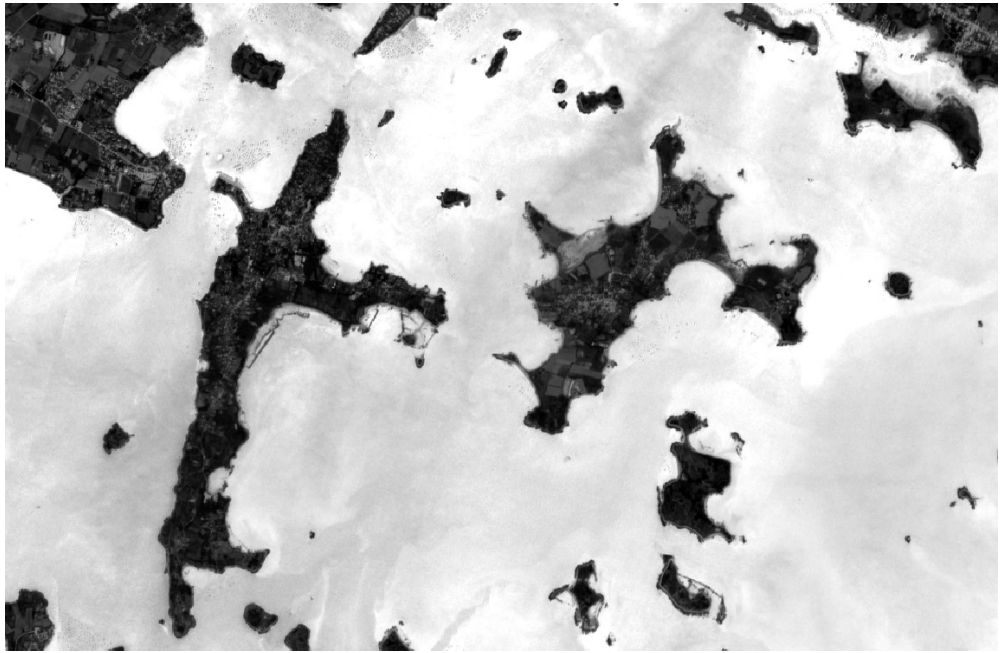
- Focus on OTB applications mechanisms
- Image analysis
- From remote-sensing to GIS tools with OTB : GDAL, Rasterio, shapely, etc.

- Python scripts

Write a processing chain with OTB in Python

- **Open a web browser (firefox, chrome, etc.)**
- **Connect to : https://colab.research.google.com/**
- **Login with your google account (gmail, etc.)**
- **Open from a notebook from github :**
  - https://github.com/otbformation/virtual-workshop
  - Select the first notebook (OTB Guided Tour)

1st step : compute radiometric indice



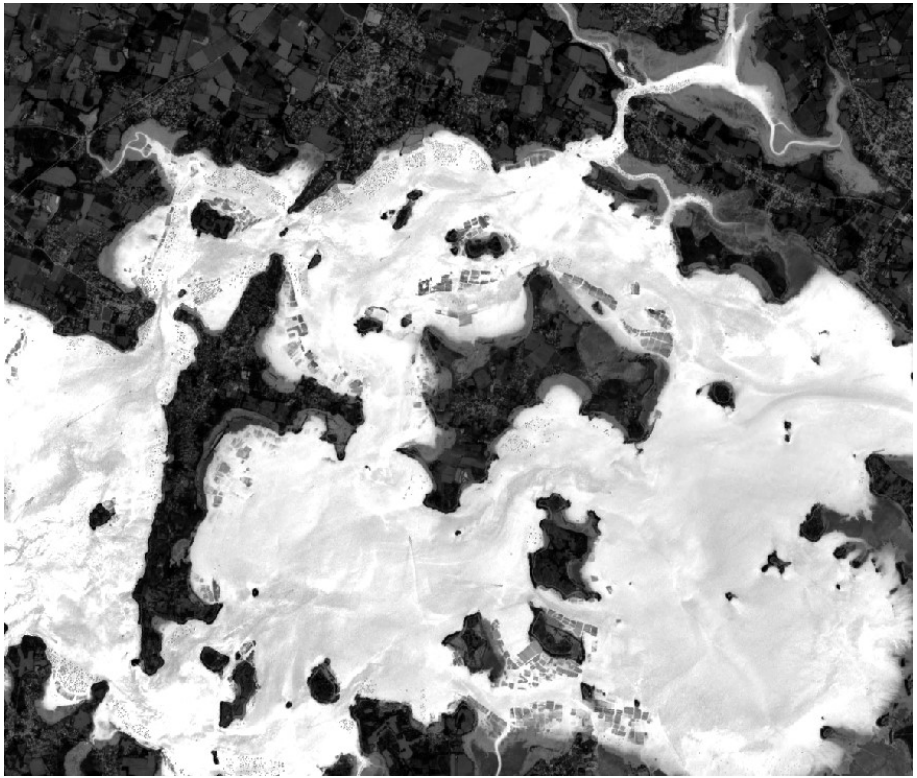
1st step : compute radiometric indice

Normalized Difference Water Index « 2 »

(green – nir) / (green + nir)

Very convenient to identify water bodies
Lands < 0
Lakes, sea > 0,3

Note : NDWI (based on NIR and MIR) is more adapted to monitor water in vegetation



OTB Python API in a nutshell ☺ (1/2)

Use OTB within Python :

```
import otbApplication as otb

app = otb.Registry.CreateApplication(« BandMath")
```

Set parameters :

```
app.SetParameterString("in", argv[1])

app.SetParameterValue(<key>,<value>)

app.SetParameterStringList(<key>,[<value1>, <value2>, ..])
```

Launch the application

`App.ExecuteAndWriteOutput()` -> wire the pipeline and write output

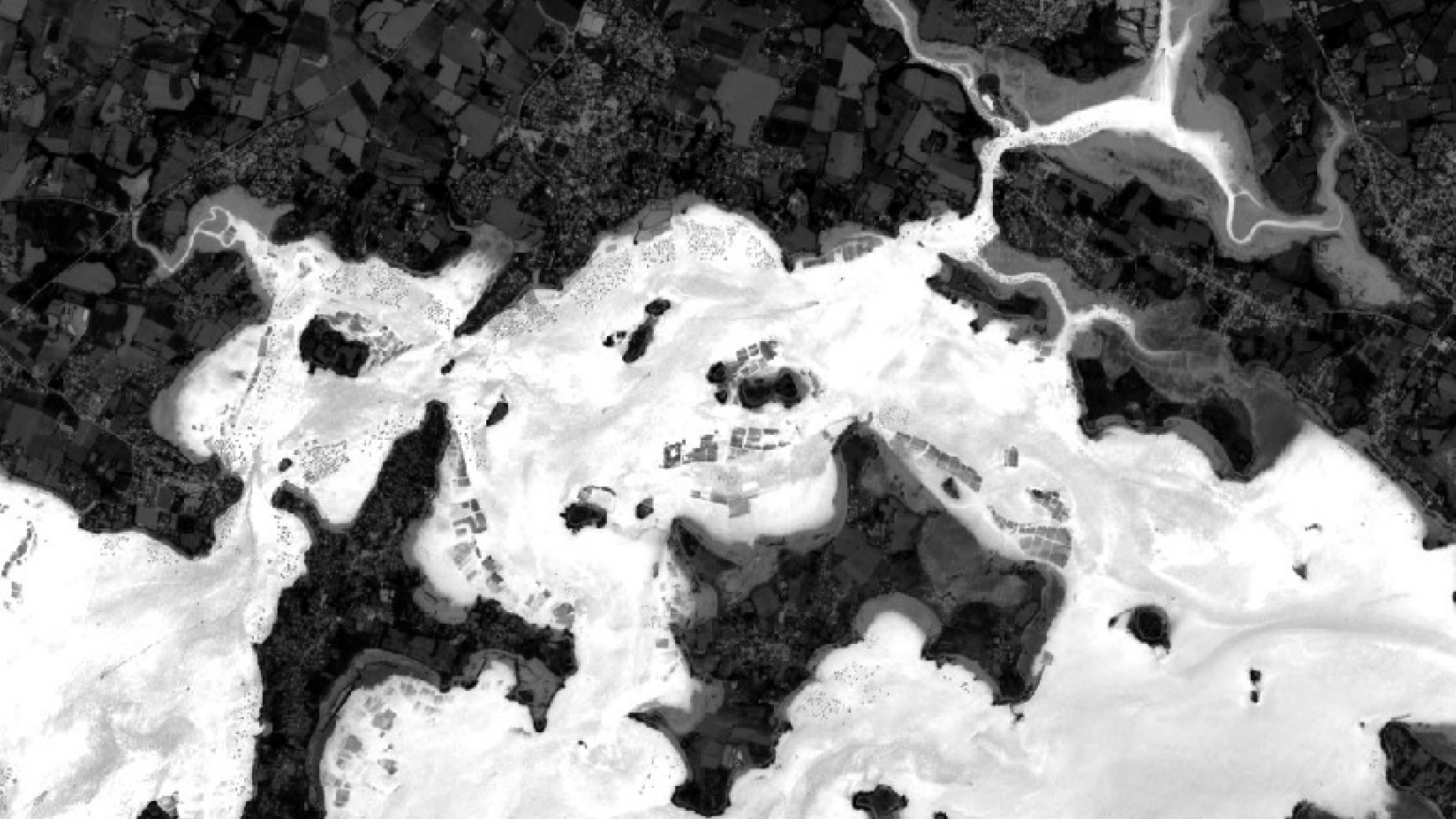
2nd step : compute a mask from the various NDWI images

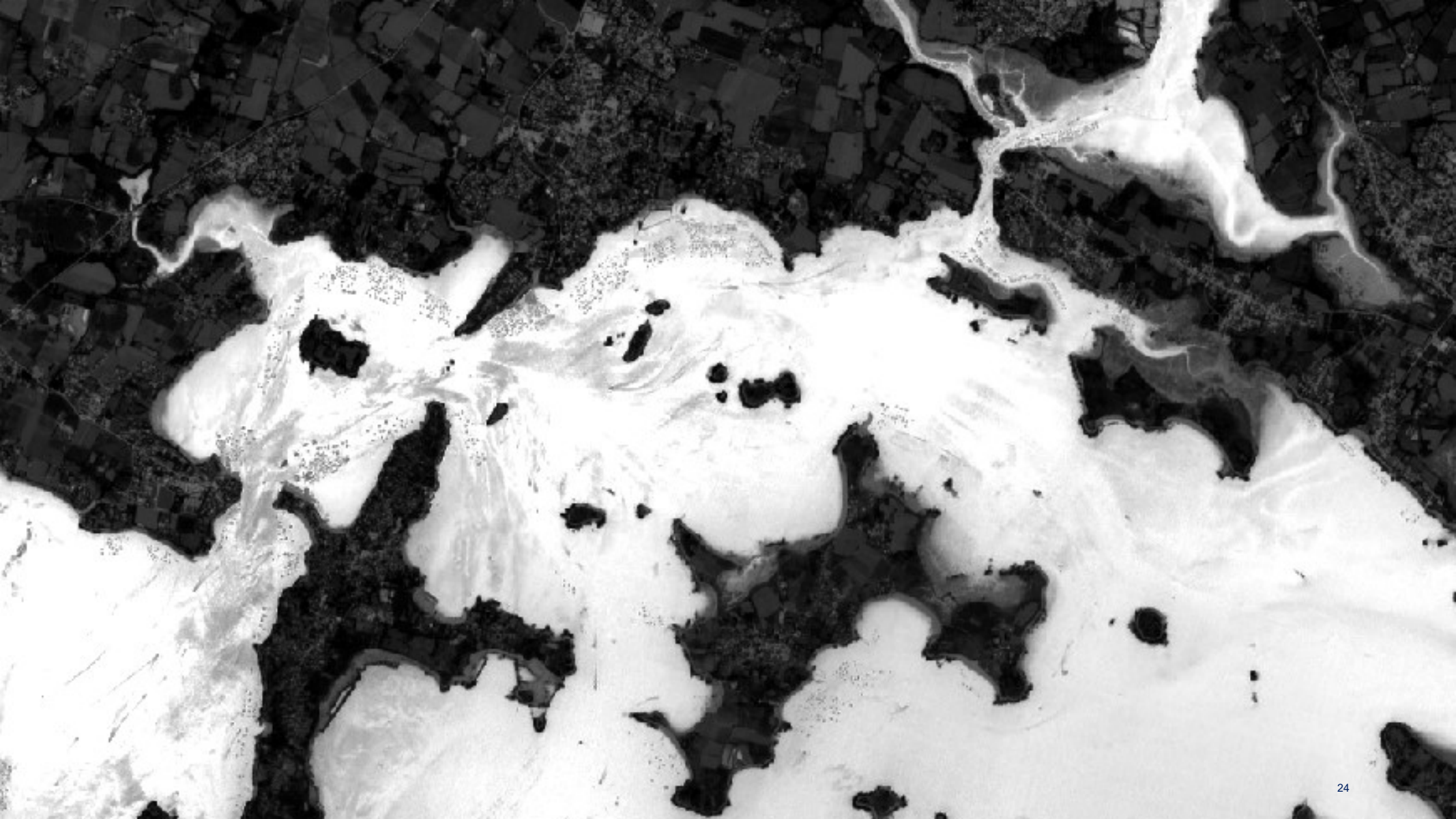
Have a look at the images ?

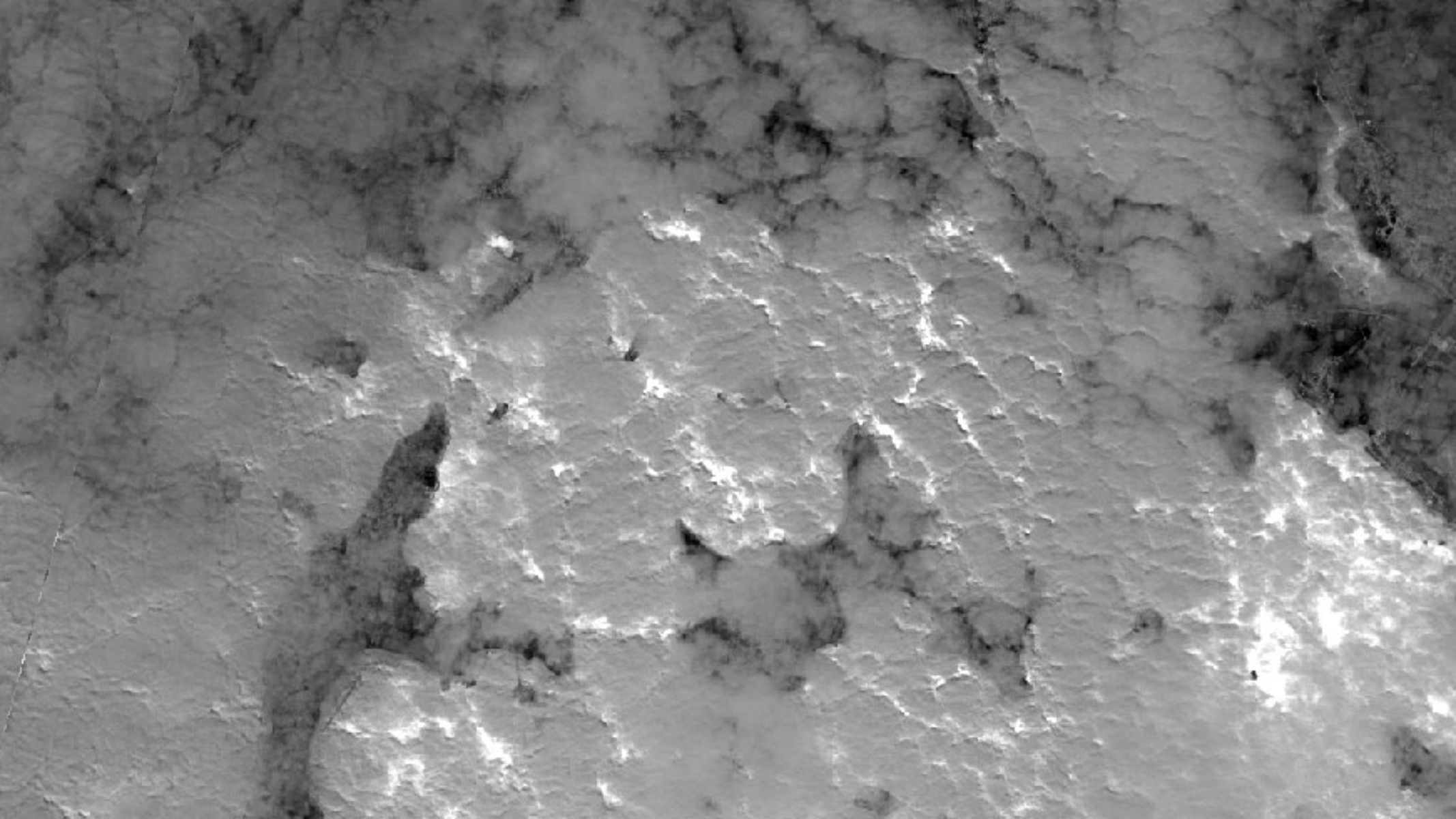
What do you observe ?

Are there some artifacts to deal with ?

What kind of formula could we use to have a mask that maximize water extent (high tides) ?







3rd step : finalize the watermask

From our NDWI2 synthesis, we want to compute a watermask.

Different solutions are available..

Easiest one : threshold on NDWI value

Alternative : use the segmentation framework in OTB. The Large Scale Mean Shift application is able to make very fine segmentation of high resolution satellite images.

We still have some artifacts :

- we could use a Binary Morphological operator to erode a little bit the mask
- Or we can filter small features (ie : small submarine / dark areas)

4th step : polygonize the raster using Gdal and filter islands

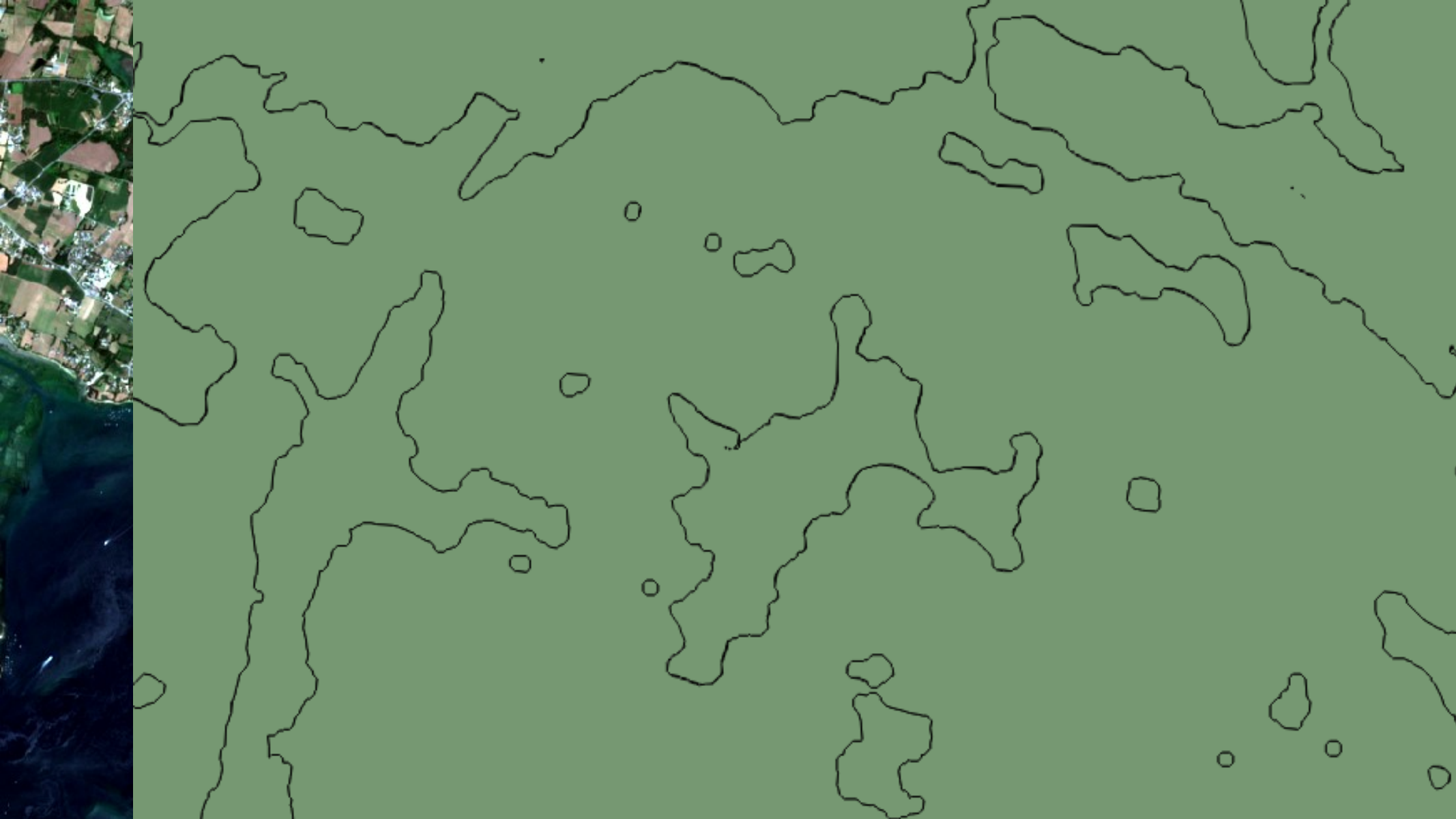
The last script will polygonize the raster file :

The different polygons are whether :

- **Islands !**
- **The ocean or sea**
- **Main land**
- **Lakes / inner water bodies**
- **Small submarine areas, only visible on lowest tides**

The script keeps shapes that intersects a shape of Morbihan gulf…

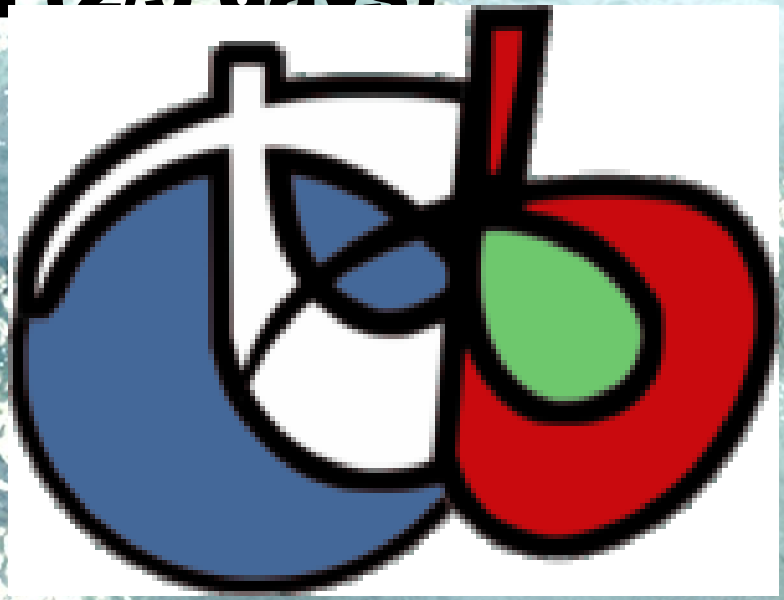
… and eliminates thoses whose area is bigger than « île aux moines » area, and smaller than 1 Ha (100m x 100m)



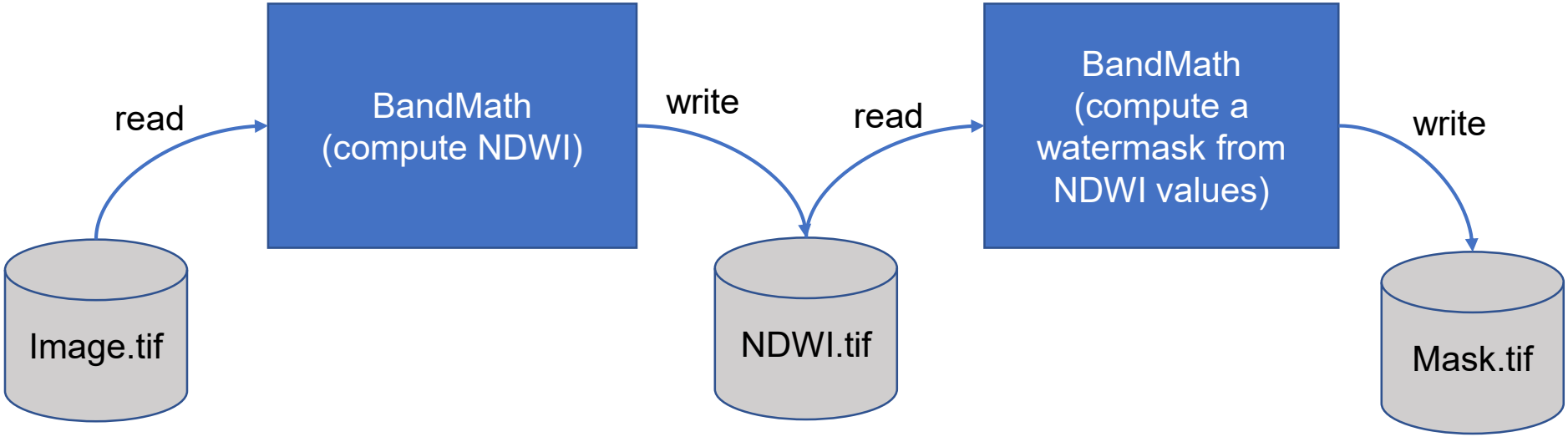
Any Question ?

To go further with OTB :

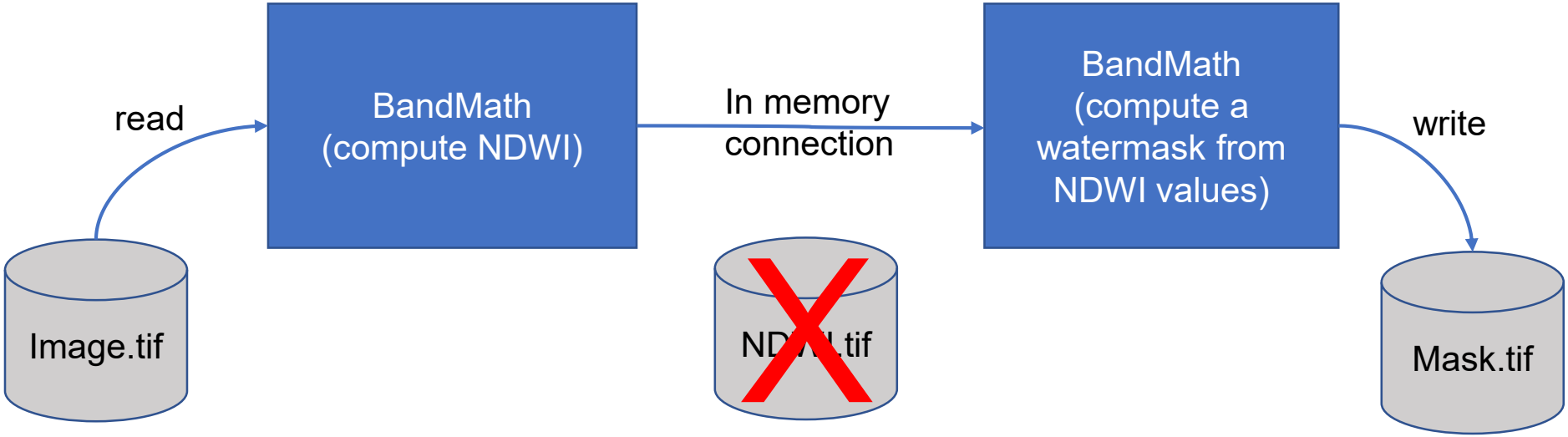
- Documentation : CookBook
- Full OTB training material
- OTB training session (2/3 days)
- User forum
- Gitlab
- OTB Days



OTB Python API : Let's play with the pipeline !



OTB Python API : Let's play with the pipeline !



OTB Python API : Let's play with the pipeline !

```
[…]
app1.SetParameterString("out", output_image)
app1.Execute()
```
→ wire the pipeline but does not execute yet !!

```
[…]
app2.SetParameterInputImage("in", app1.GetParameterOutputImage("out"))
```
→ set the output of app1 as the input of app2

Or `app2.AddImageToParameterInputImageList("in", app1.GetParameterOutputImage("out"))`

→ same for an application that takes a list of input

`app2.ExecuteAndWriteOutput()` -> resolve the pipeline, launch all processings and write output

OTB Python API : Let's play with the pipeline !

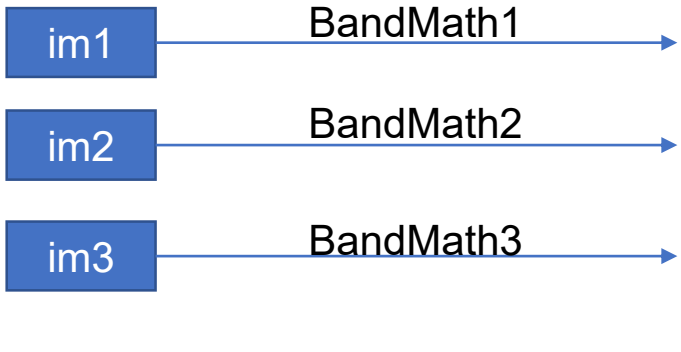


Image List : il = [BandMath1.GetParameterOutputImage(« out »),
BandMath2.GetParameterOutputImage(« out »),
BandMath3.GetParameterOutputImage(« out »)]

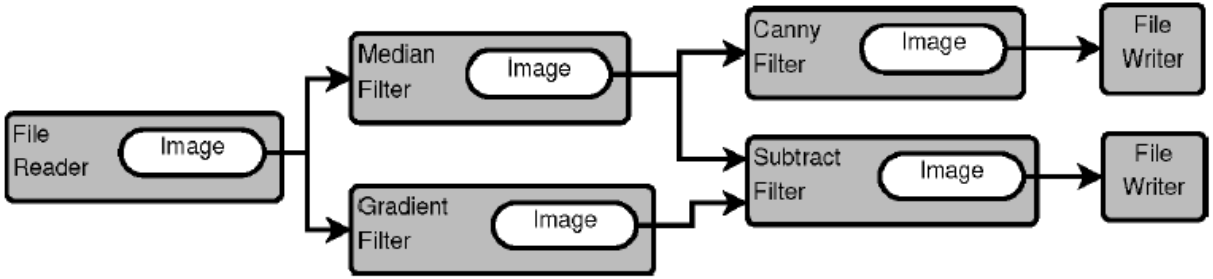
Expression : exp = « f(im1b1, im2b1, im3b1) »

Compute Radiometric Indice (NDWI2)…
…. But does not write file !

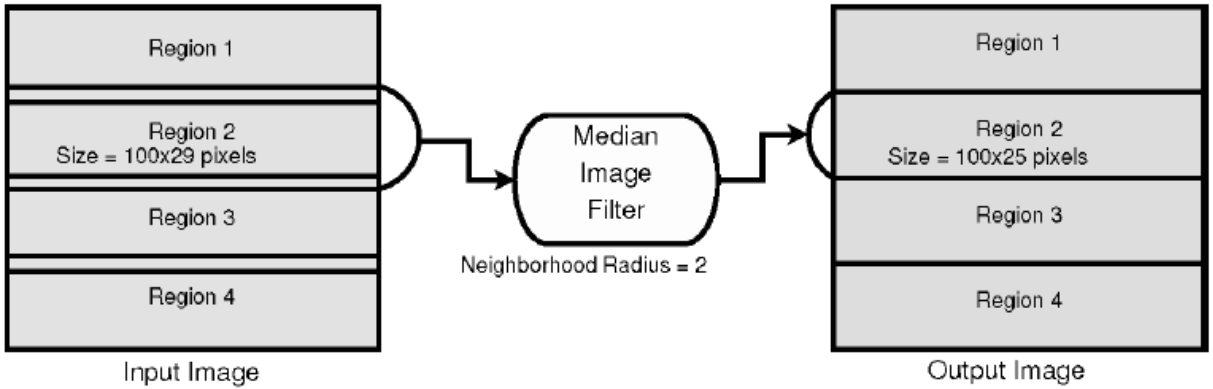
Makes a synthesis of the time serie of NDWI2…
… and writes the image !

A few words on the OTB pipeline (inherited from ITK)

Le modèle de *Pipeline*



Streaming



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