

# TP: SEGMENTATION & CLASSIFICATION

**Classification orientée objet**

**Ingénieurs 2ème année**

Janvier 2023

Arnaud LE BRIS & Martin CUBAUD

# Principe

---

- Le TP se décompose en 2 parties :
  - Segmentation d'images (TP1)
  - Classifications orientées objet (TP2)
- Le TP1 se fait à la maison
- Le TP2 se fait là maintenant (et se termine à la maison au besoin)
- **La note finale porte sur le rendu des 2 parties.**
  - Plus de détails sur le dernier transparent.
- Merci de lire l'énoncé du TP en entier avant de commencer

# Outils

---

- On utilisera Python et la boîte à outils OTB (Orfeo Tool Box) disponible :
  1. Sous formes d'exécutables indépendants (ligne de commandes ou GUI)
  2. Dans le logiciel QGIS (fenêtre de droite) dans l'image DIAS/DPTS

# TP1 – SEGMENTATION

# Ligne de Partage des Eaux

## •Utilisation de la boîte à outils Orfeo ToolBox

- Module *Segmentation* :
  - Choisissez la méthode « *watershed* » ;
  - Sélectionnez l'image en entrée en ne conservant pour commencer qu'un seul canal (Rouge par exemple) ;
  - Entrez le nom de l'image en sortie (.tif pour le raster ; .gml pour le vecteur)
  - **Nous conseillons le vecteur** ;
  - Exécutez.
  - Visualisez le résultat sous QGIS
    - Chargez la couche
    - Changez le style (clic droit « Propriétés » puis mettez la couleur de fond en transparent et la bordure en couleur qui flashe)
    - Sauvegardez ce style pour vos différentes expériences

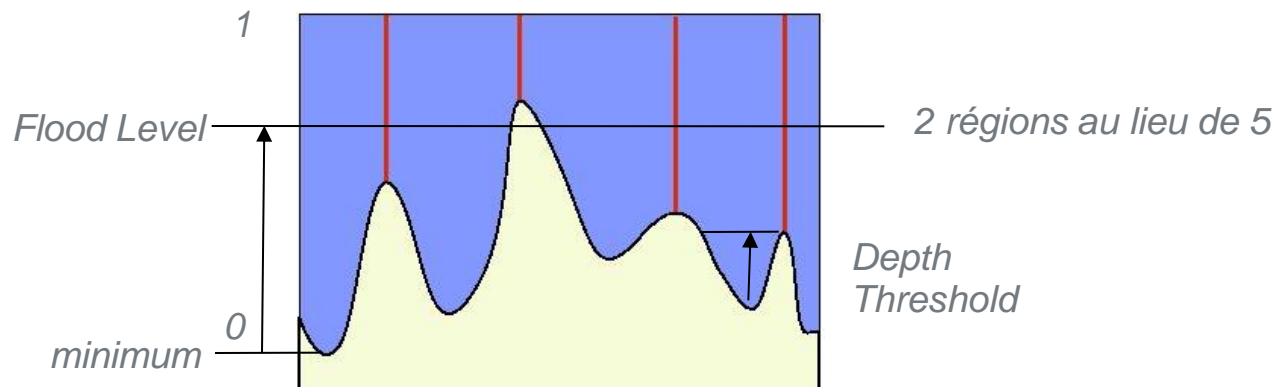
# Ligne de Partage des Eaux

## ■ Paramétrage :

- Altitude : magnitude du gradient de l'amplitude (racine carré de la somme au carré des canaux de l'image)
- *Depth Threshold* : différence en altitude entre le minimum du *watershed* et le point de LPE le plus bas → profondeur maximum d'eau qu'une région peut contenir sans couler chez ses voisins.
- *Flood level* : niveau d'eau pour fusionner ou non des bassins voisins (0=*watershed* de base ; 1=image entière). Fondé sur la différence d'altitude max-min.

# Ligne de Partage des Eaux

- **Paramètres:** « DepthThreshold » et « Flood Level »
  - Ils permettent de fusionner des régions voisines une fois l'algorithme du watershed fini.
  - L'arbre de fusion générée par la LPE est élagué.
  - DepthThreshold = paramètre **local**.
  - Flood Level = paramètre **global**.



# Ligne de Partage des Eaux

## ■ Tests : Château de Chambord

- Lancez le watershed sur une des 2 zones d'intérêt (RVB ou IRC, comme vous voulez) avec un seul canal pour commencer ;
  - Regardez l'influence des paramètres ;
  - Une fois que vous avez compris, passez à 3 canaux.
  - Quel(s) pré-traitement(s) faire pour améliorer les résultats ? Utilisez OTB pour cela
    - Lissage de l'image, rehaussement de contours (*Smoothing*) ?
  - (BONUS) Quels attributs extraire et/ou combiner pour améliorer les résultats ? (FeatureExtraction dans otb, par exemple *HaralickTextureExtraction*)
    - La concaténation d'images se fait avec l'outil OTB *Concatenation* si vous voulez regrouper plusieurs attributs
    - Les opérations mathématiques entre bandes d'une ou plusieurs images se fait avec *Band Math* (OTB) si vous voulez calculer des attributs vous-mêmes.
- (BONUS) Proposer une solution optimale et l'argumenter svp.



# TP2 – ANALYSE OBIA

# OBJECTIFS

---

- ▶ Tester différentes manières de fusionner classifications et segmentations
- ▶ Utilisation de **Python et QGIS** pour la visualisation des résultats.

Par la suite on parle :

- Image de télédétection : c'est l'image initiale 3/4 canaux
- Image de classification : chaque pixel de cette image correspond à une classe (valeur discrète entre 1 et 5, par exemple)
- Image de segmentation : chaque pixel correspond à l'identifiant du segment (valeur discrète entre 1 et 10000 par exemple).
  - les pixels avec la même valeur appartiennent au même segment
  - il y a moins de segments que la valeur de l'identifiant max (car certaines régions ont disparu dans le processus)

## EXERCICE 2.1

### ► Régularisation d'une classification pixellaire avec une segmentation :

*Chaque pixel de l'image de classification est réaffecté à la classification **majoritaire** dans le segment auquel il appartient.*

Entrée : une image de classification **image**, une de segmentation **image**

Sortie : une nouvelle image de classification régularisée

Avec Python : parcours de l'image de classification (IRC\_classif.tif).

*Pour chaque pixel,*

- Récupération des classes des pixels appartenant au même segment (via IRC\_Segmentation.tif);
- Calcul de la classe majoritaire ;
- Affectation de cette valeur dans l'image de sortie.

## EXERCICE 2.1

### ► Algorithmiquement :

- Récupérer le nombre de classes :  $nbc$
- Récupérer le nombre de segments :  $nbs$
- Créer une matrice  $M$  de zéros de taille  $(nbc, nbs)$
- Parcourir conjointement les images de segmentation et de classification:
  - Pour chaque pixel, ajouter +1 à l'élément (classe,  $id\_segment$ ) de  $M$   
 $classe = \text{valeur du pixel dans l'image de classification}$
- Créer un vecteur  $V$  colonne de taille  $(nbs)$  et récupérer l'identifiant de la valeur max pour chaque segment
- Créer une image vide de taille identique à celle de segmentation
  - Pour chaque pixel, affecter la valeur de la ligne  $V[id\_segment]$ .

Attention: en pratique la numérotation des régions de la segmentation ne commence pas à 0 et n'est pas continue... donc au choix :

on renumérote les régions de manière à avoir des identifiants compris entre 0 et  $nbs$   
ou on prend  $nbs$  comme étant l'identifiant maximal

## EXERCICE 2.2 (1/2)

### ► Régularisation des attributs pour une classification

*Chaque pixel de l'image de segmentation est réaffecté à la valeur moyenne de sa radiométrie dans le segment auquel il appartient.*

Entrée : une image IRC, une de segmentation image

Sortie : une image IRC moyennée (3 canaux également)

Avec Python : parcours de l'image IRC.

*Pour chaque pixel,*

- Récupération des valeurs des 3 canaux I/R/C des pixels appartenant au même segment ;
- Calcul de la moyenne pour chaque canal;
- Affectation de ces valeurs dans l'image de sortie.

## EXERCICE 2.2 (2/2)

### ► Régularisation des attributs pour une classification

*L'image 3 canaux générée à l'étape 1 est ici classée.*

Entrée : une image IRC moyennée, un fichier d'apprentissage  
(*classification\_oraison.model*)

Sortie : une image classée (les classes sont celles qui sont indiquées dans le  
fichier ci-dessus, les mêmes que pour le TP 2.1).

Avec OTB (via OTB ou pas): utilisez l'application « Image Classification »

- Input Image = l'image créée à l'étape 1
- Model File = le fichier « *classification\_oraison.model* » présent avec les données du TP.
- Output image = l'image qui va être générée.

## EXERCICE 2.2 (2/2)

### ► Régularisation des attributs pour une classification

*Comparez vos résultats avec celui de l'exercice 2.1*

*Faites pour cela une carte des différences binaire (0/1 : pareil/pas pareil) (BONUS) puis classe à classe (1 entier pour chaque transition).*

*Quelle solution préférez-vous ? Pourquoi ?*

## EXERCICE 2.3

### ► Amélioration sémantique d'une classification:

*Tous les pixels d'un segment sont affectés à une classe qui dépend de la proportion des classes présentes dans le segment.*

Entrée : une image de classification image, une de segmentation image

Sortie : une image de nouvelle classification régularisée

Avec Python : parcours de l'image de segmentation grossière.

*Pour chaque pixel,*

- Récupération des classes des pixels appartenant au même segment ;
- Choix de la nouvelle classe (selon vos critères);
- Affectation de cette valeur dans l'image de sortie.

*Quelles classes choisissez vous ?*

*Selon quels critères (à justifier) ?*

*On pourra s'intéresser au voisinage des régions, ou au mélange au sein des régions (pures/mixte) et essayer d'en déduire de nouvelles classes.*

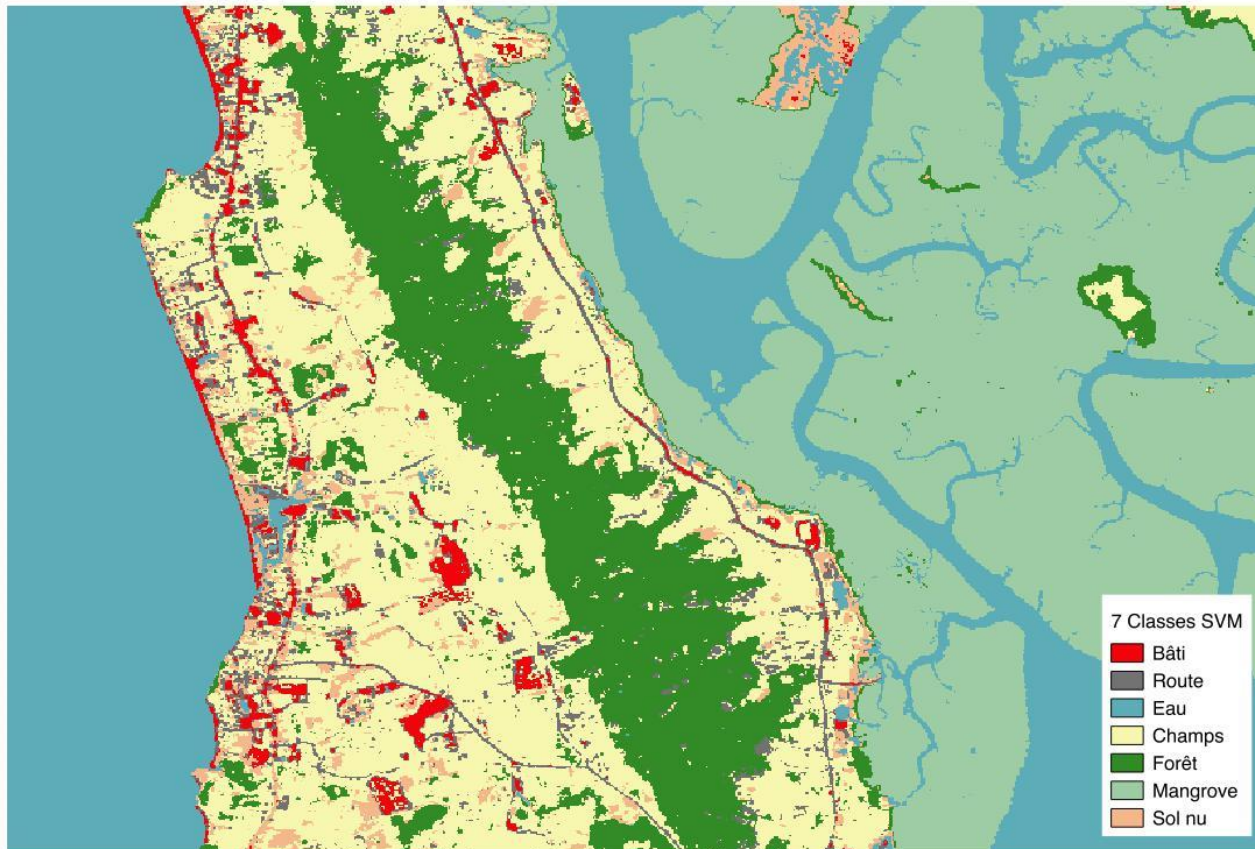


## EXERCICE 2.3



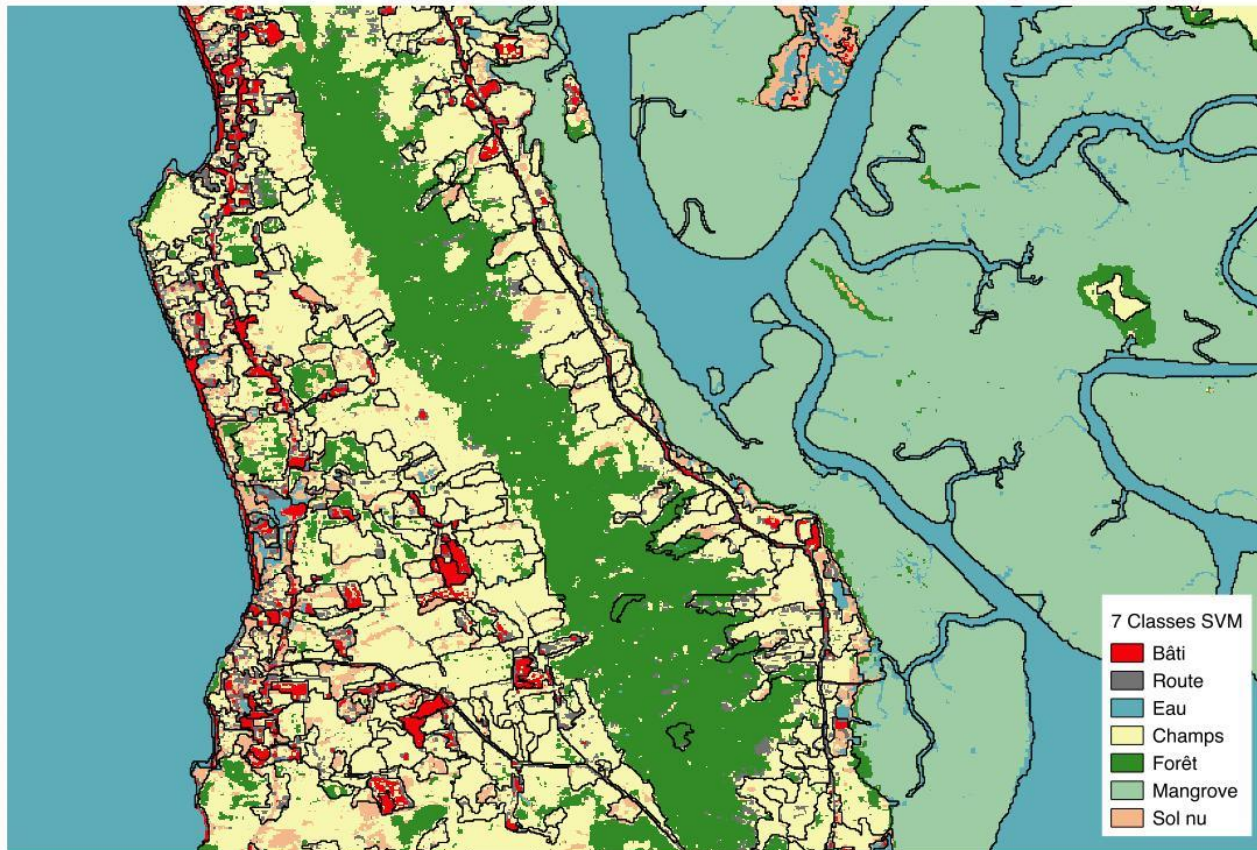
Ko Lanta: Image IRC

## EXERCICE 2.3



Ko Lanta: Classification

## EXERCICE 2.3



Ko Lanta: Segmentation (contours noirs)

# RENDU FINAL

## ► Compte-rendu écrit des 2 TP :

- Remplace l'examen écrit pour cette partie ;
- Vaut 7 points (+ bonus si plus de travail) sur 20 de l'examen final.
- **Rendu : de préférence par groupe de 2**
  - Rapport en PDF ;
  - De taille raisonnable ;
  - Avec analyses consolidées synthétiques et non un large éventail de résultats quasi similaires ;
  - Code ou pseudo-code (dans le PDF) accompagnant les résultats de la partie 2.3 ;
  - Pour le 18 février ;
  - Rendus à déposer sur :  
formationTemp\ING2\Traitement\_image\TP\_OBIA\Rendus
  - Quand le rendu est déposé ou en cas de question,  
envoyer un courriel à [arnaud.le-bris@ign.fr](mailto:arnaud.le-bris@ign.fr)