

# *Classifications*

- ☞ *Classifications automatiques (non supervisées)*
- ☐ *Classifications supervisées*
- ☐ *Analyse en Composantes principales*

Pierre-Louis Frison  
pierre-louis.frison@u-pem.fr



# *Références*

- Image analysis, Classification, and change detection in Remote sensing  
Canty, ed. CRC Press, 2010
- Introduction to remote sensing, 5<sup>ième</sup> édition  
Campbell and Wynne, ed. the Guilford Press, 2011
- Remote Sensing and Image Interpretation, 6<sup>ième</sup> édition  
Lillesand, Kieffer and Chipman, ed J. Wiley & sons, 2007

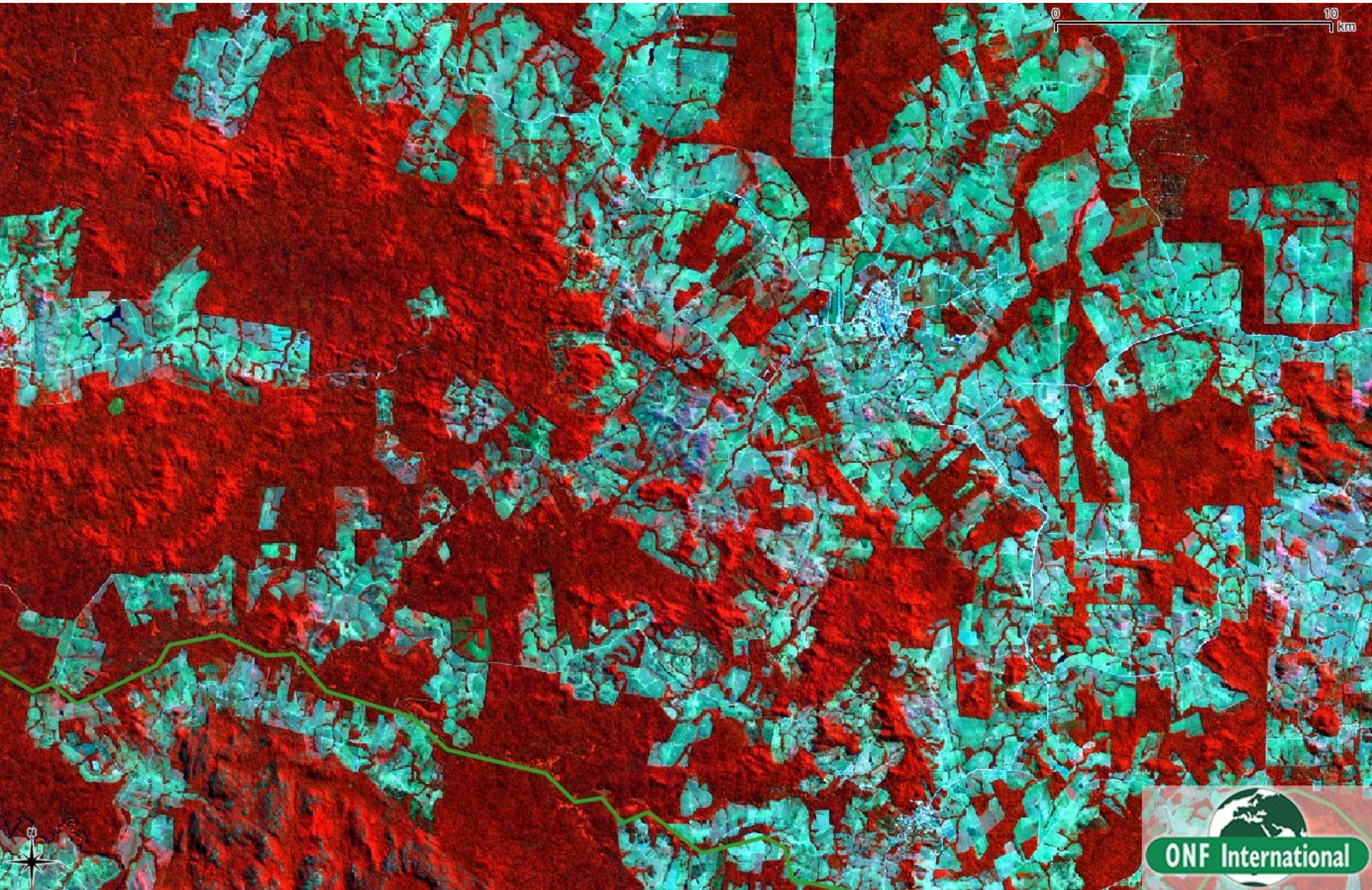
# *CLASSIFICATION*

image de télédétection

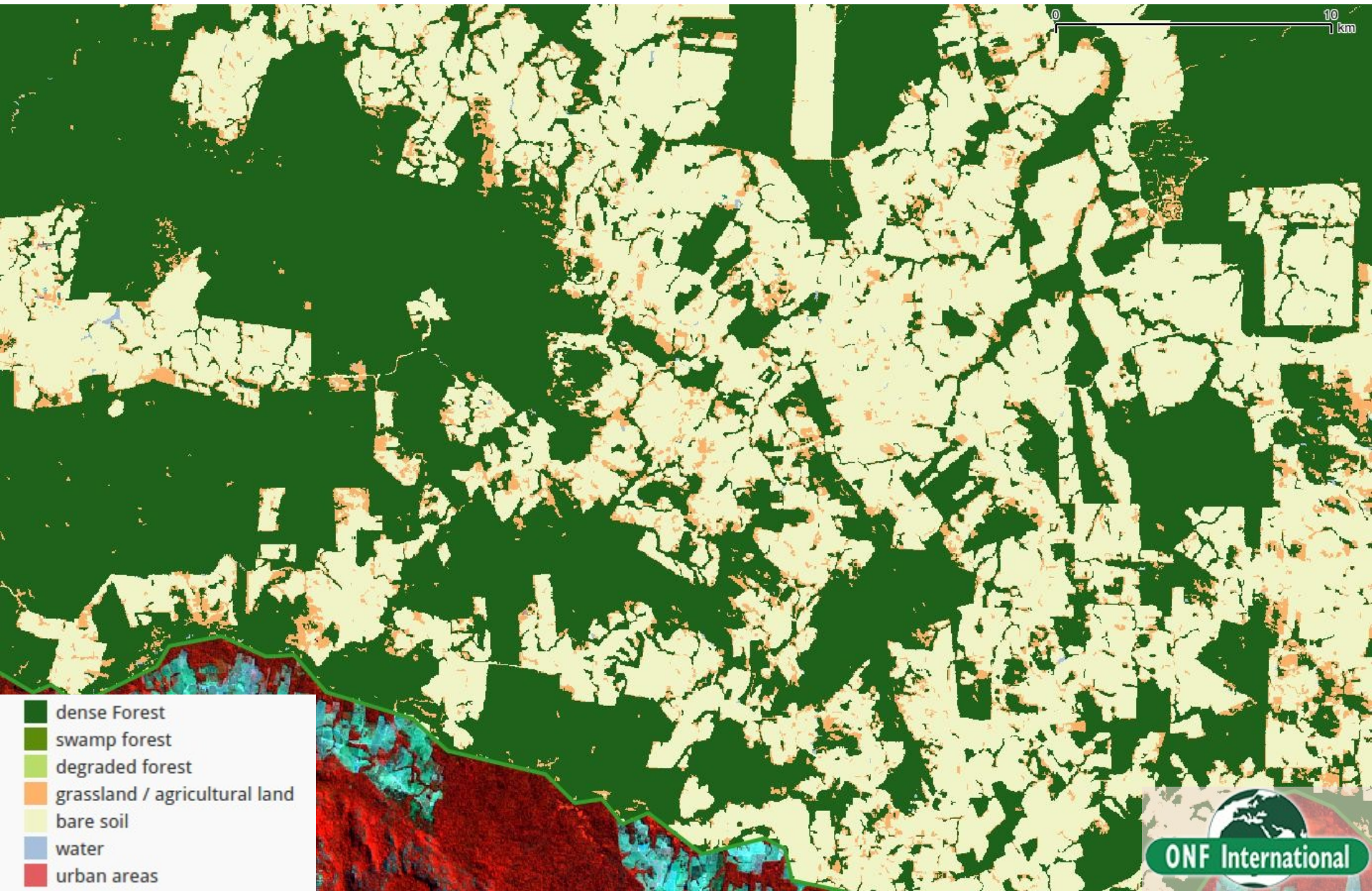


carte thématique  
(Occupation des sols, types de  
nuages, .....)

# *Image LANDSAT TM*



# *Résultat de classification*



# QUELQUES RÉSULTATS

- Détection de changements entre une BD 2D topographique et une image aérienne ou satellite

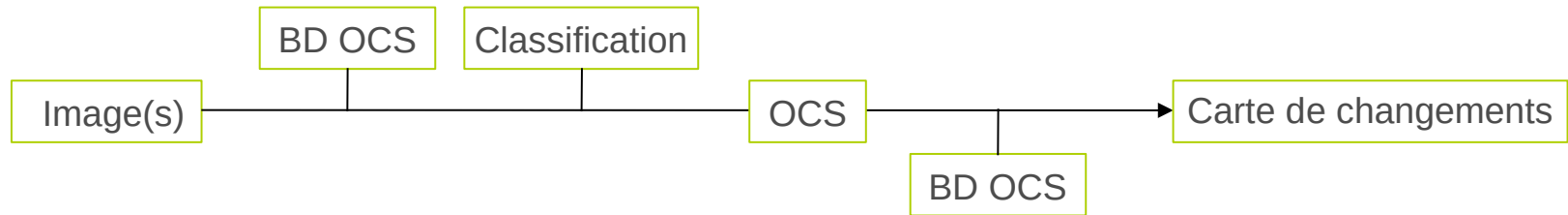
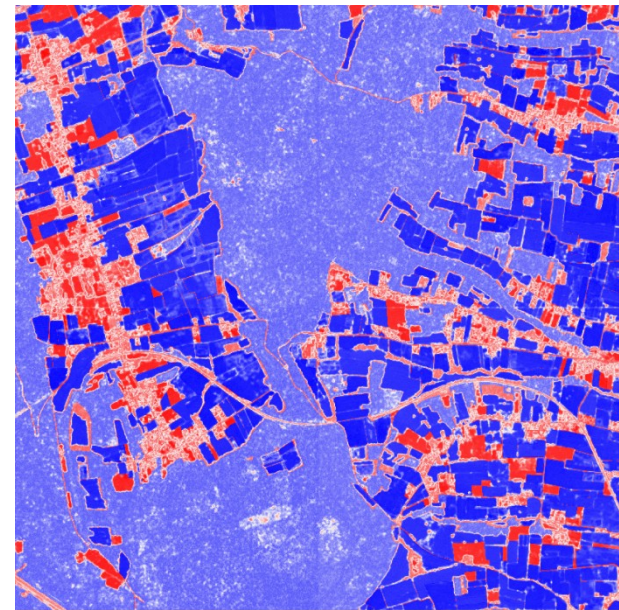


Image Pléiades (0,5m)



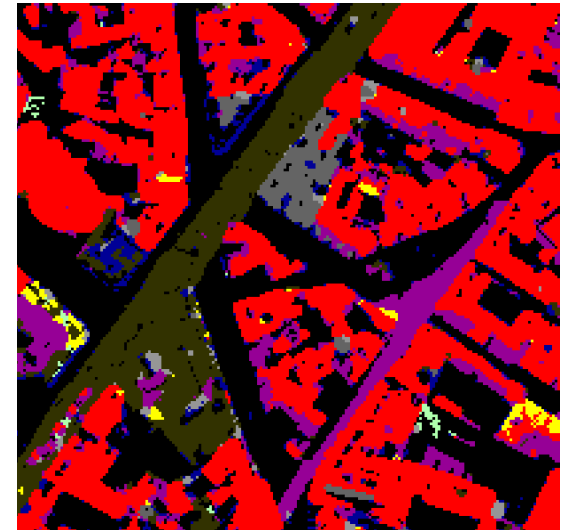
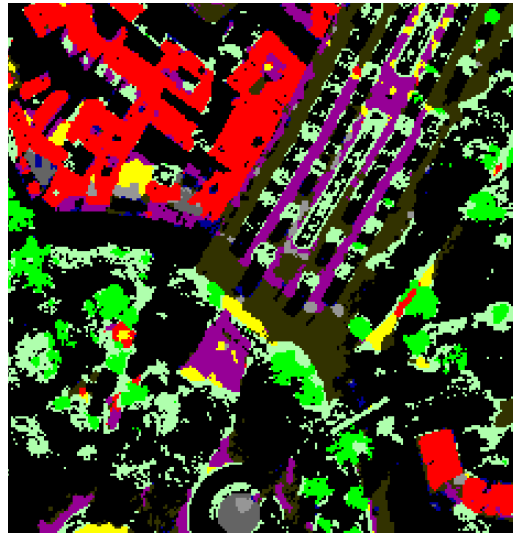
RGFor - RPG



Non changement/changement

# QUELQUES RÉSULTATS

## ■ OCS enrichie par imagerie hyperspectrale



Campagne Umbra  
(IGN-ONERA)  
0,8m → 1,6m

- Tuiles rouges
- Métal 1
- Métal 2
- Ardoises
- Pavés
- Asphalte
- Sol nu
- Végétation haute
- Végétation basse

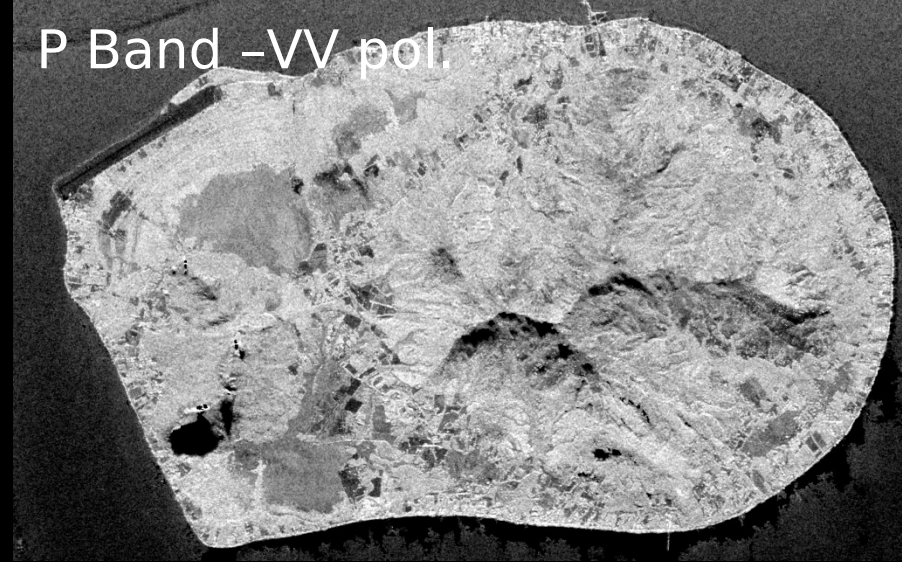
# Frequency diversity - Wavelength

Tubuai Island, French Polynesia

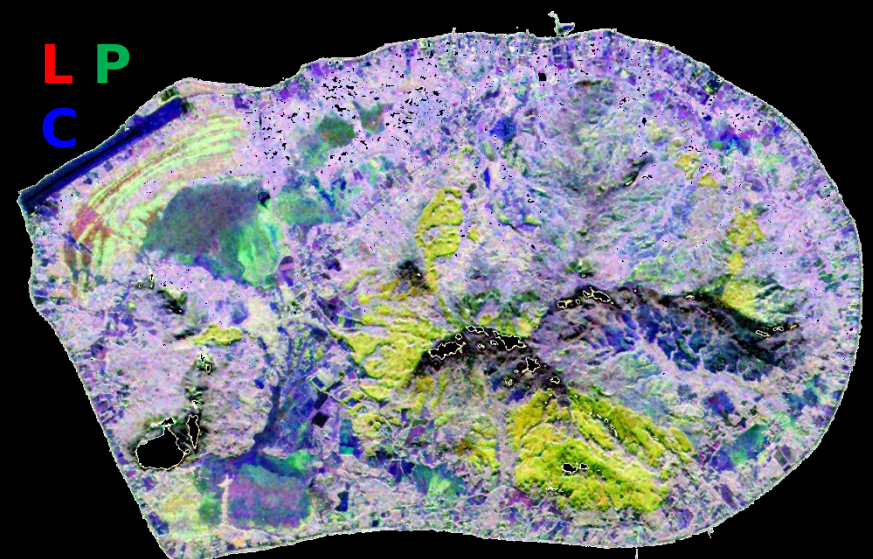
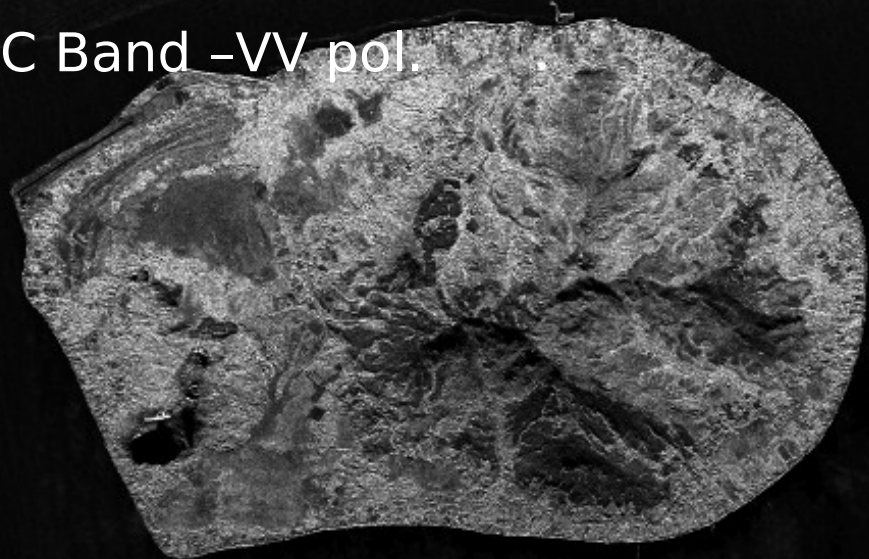
L Band -VV pol.



P Band -VV pol.



C Band -VV pol.



Capteur aéroporté AIRSAR,



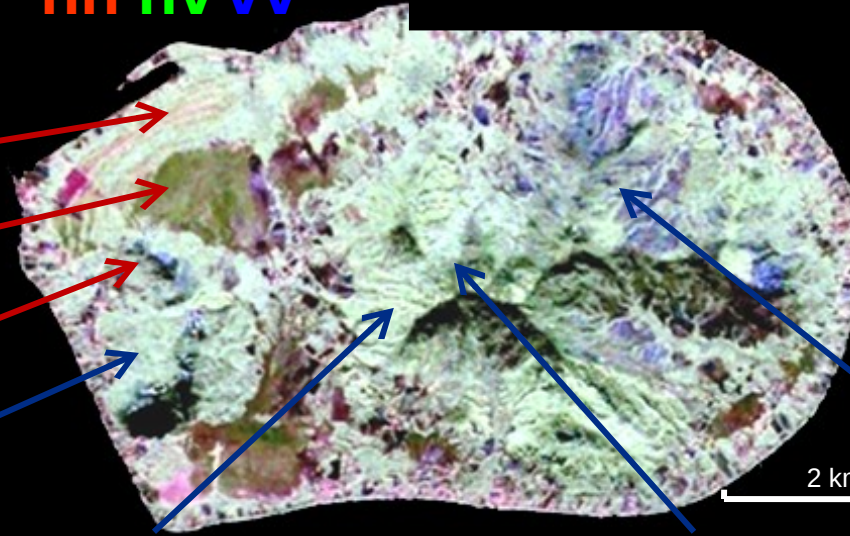
# radar polarimetry for forest cartography

**Tubuai Island, French  
Polynesia**

## 7 different classes:

HH HV VV

- bare soils
- swamps
- fern lands



AIRSAR data  
L Band ( $\lambda = 24$   
cm)  
Aug. 2000

## 4 forest species

• Purau

• Pine

• Falcata

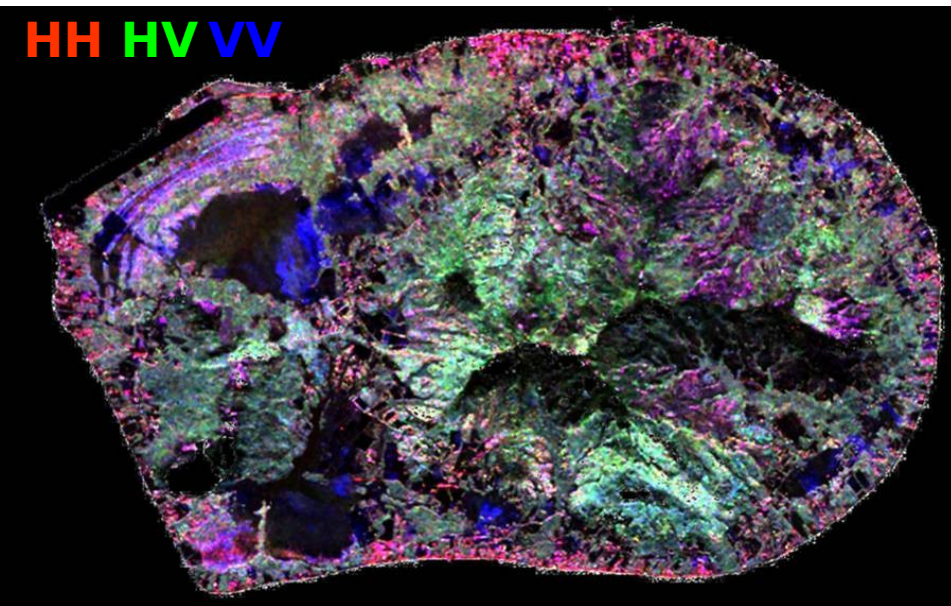
• Guava



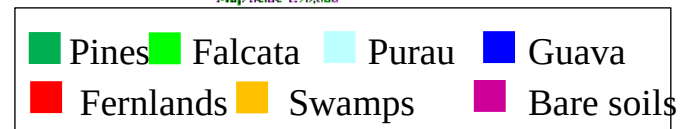
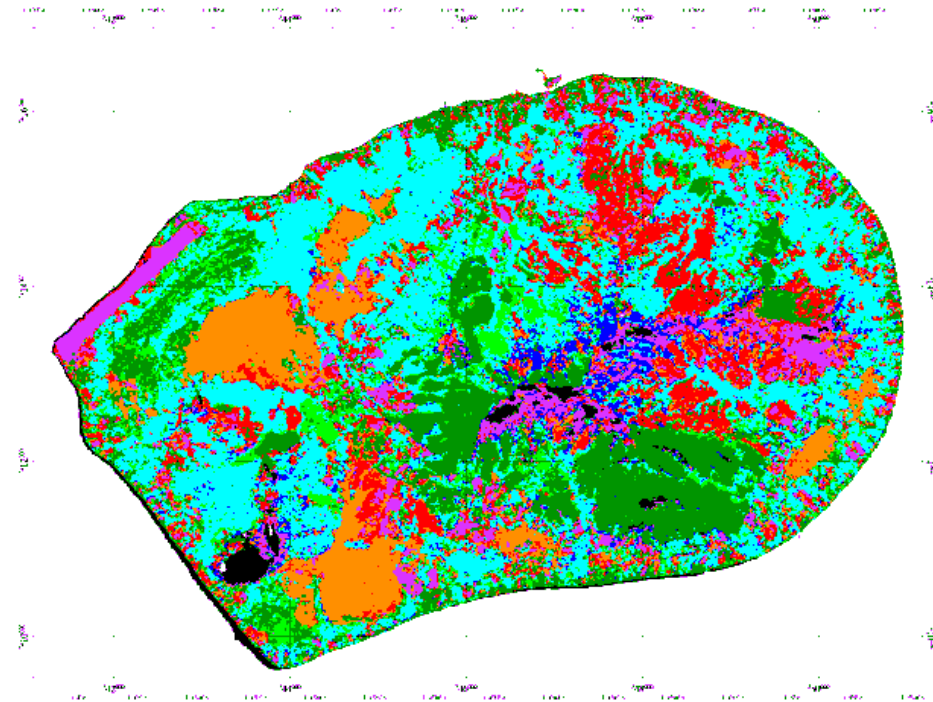
# Classification with radar polarimetry

*Tubuai Island (French Polynesia)*

**AIRSAR data**  
*August 2000*



**Classification**



*(4 Forest types)*

# *CLASSIFICATION*

*basée sur l'exploitation de l'information:*

***spectrale***

***spatiale*** (notion de voisinage, pixels contigus, ***texture***)

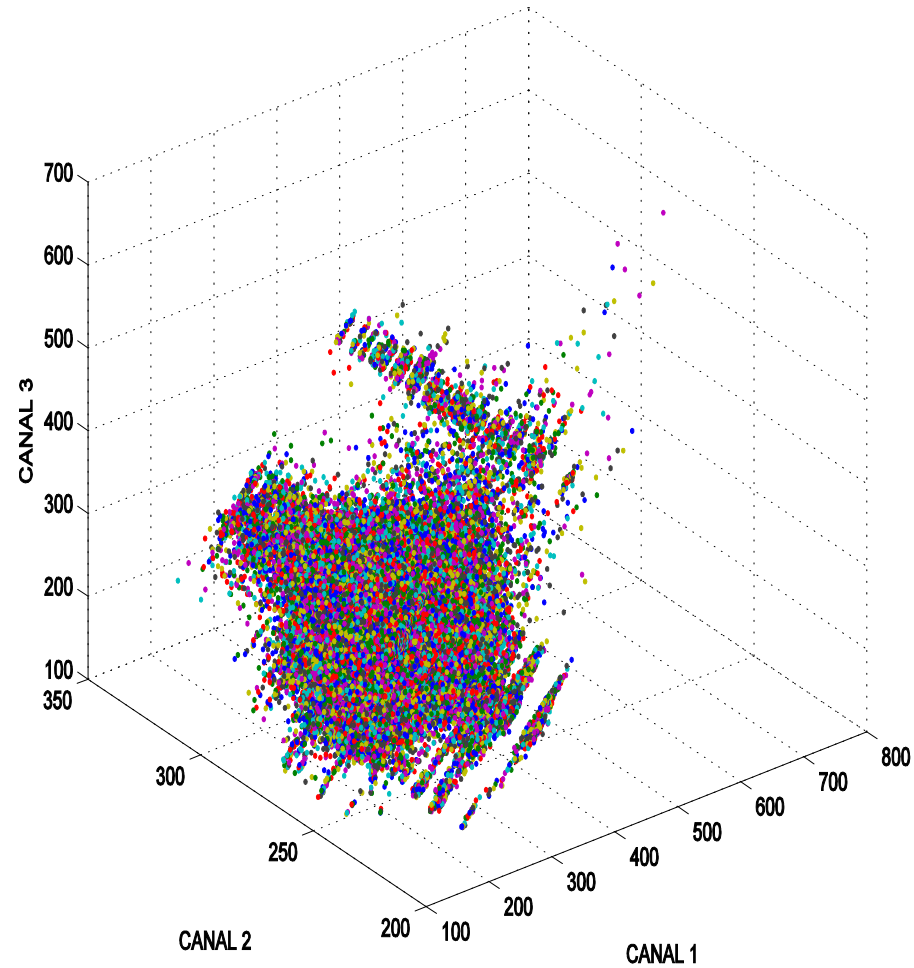
et/ou forme (***orientée objet***)

# *INFORMATION SPECTRALE*

Domaine spatial



Domaine spectral



*Image aéroportée AVIRIS, Cuprite, USA*

# *CLASSIFICATION*

*basée sur l'exploitation de l'information:*

***spectrale***

***spatiale*** (notion de voisinage, pixels contigus, ***texture***)

et/ou forme (***orientée objet***)

## *Example of automatic classification*

### *The k-means algorithm*

*Based on minimum distance criteria in the spectral domain*

- 1) Segmentation of the spectral space into  $k$  sub-spaces  
( $k$  set *a priori*)

Next iteration:

- 2) Computes the barycentre of each sub-space
- 3) Pixel affectation to the class with closest barycentre

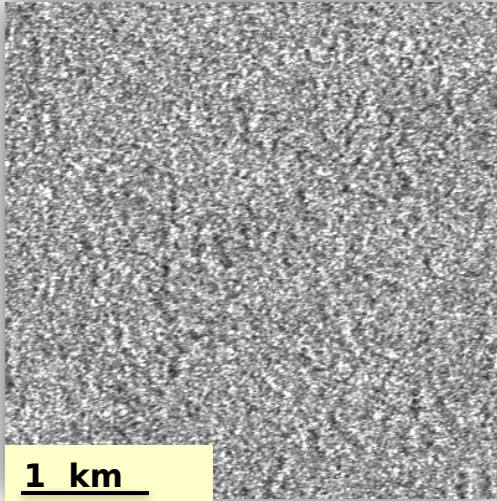
Repeat 2) and 3) until

                  a maximal number of iterations (set *a priori*)  $t$   
                  or       differences between barycentre position  $< \varepsilon$  ( $\varepsilon$   
threshold set *a priori*)

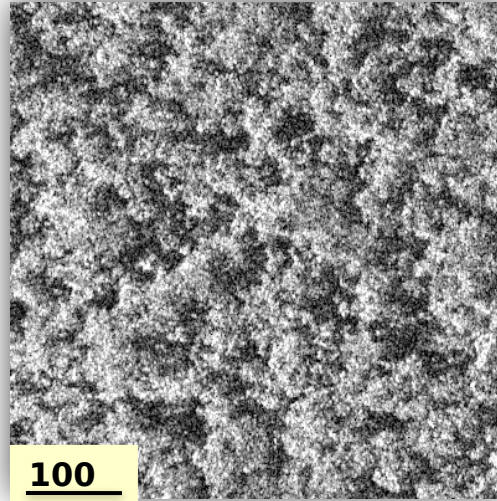
# INFORMATION TEXTURALE

## Données RADAR

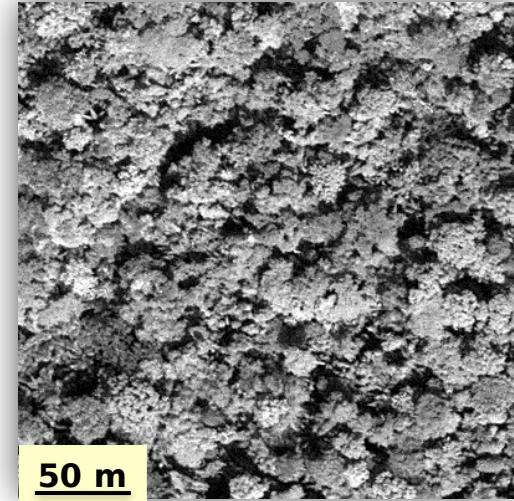
## Donnée Optique



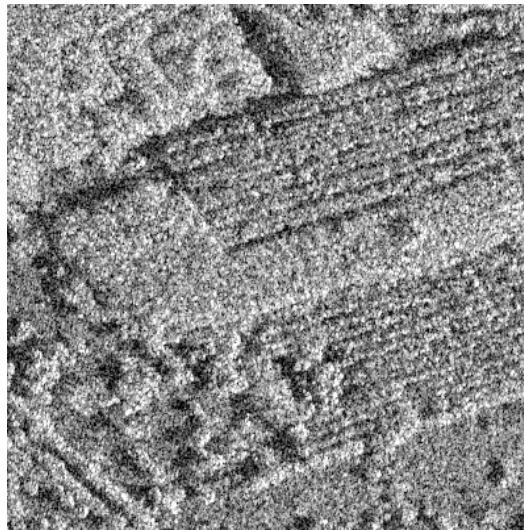
**PALSAR**, HH, mode FBD,  
Résolution : **30 m**



**terraSAR-X**, SpotLight-HH,  
Résolution : **1 m**



**Geoeye**, Panchromatique,  
Résolution : **0.5 m**



# *CLASSIFICATION*

*basée sur l'exploitation de l'information:*

***spectrale***

***spatiale*** (notion de voisinage, pixels contigus, ***texture***)

et/ou forme (***orientée objet***)



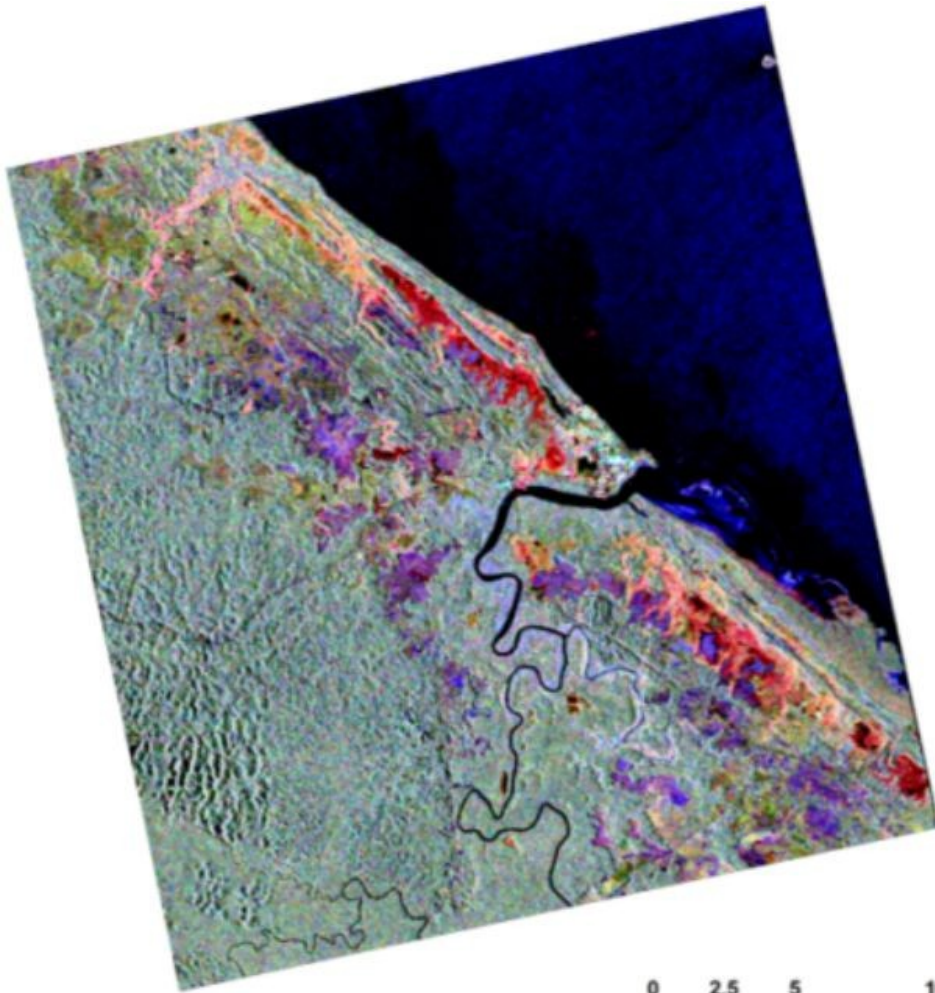
# ***CLASSIFICATION ORIENTEE OBJET***



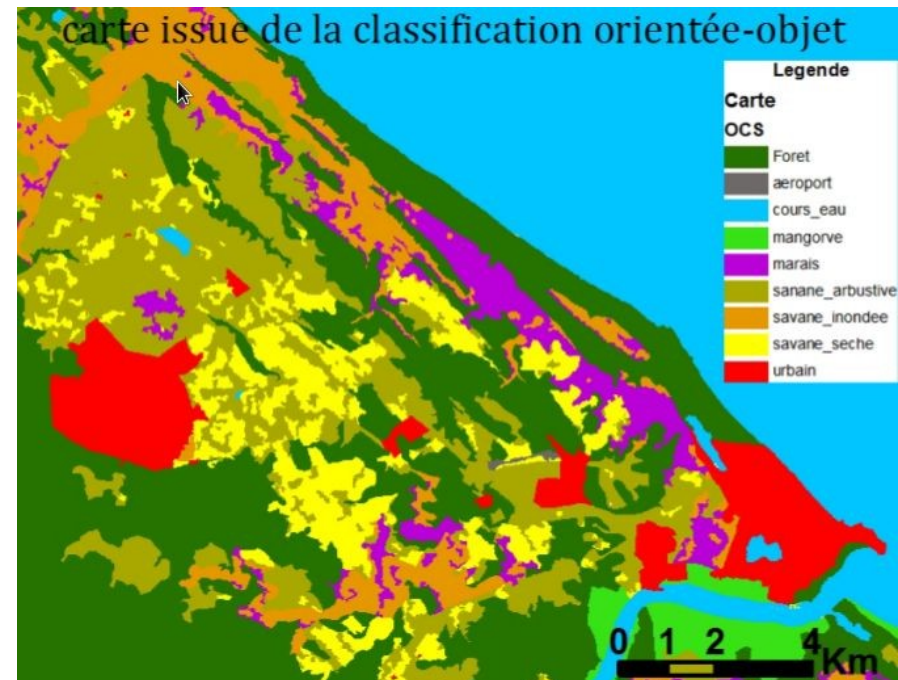
Source: Geosystems

# CLASSIFICATION ORIENTEE OBJET

Image RADARSAT2



0 2.5 5 10 Km



Legende

Carte

OCS

- Foret
- aeroport
- cours\_eau
- mangrove
- marais
- sanane\_arbustive
- savane\_inondee
- savane\_seche
- urbain

0 1 2 4 Km

# CLASSIFICATION

*2 types d'algorithmes*

## ***Classifications automatiques (non supervisées)***

*L'algorithme détermine des classes automatiquement*

## ***Classifications supervisées***

*Des zones d'intérêt sont définies par l'utilisateur  
(relevés terrain, photo-interprétation)*

*L'algorithme calcule les propriétés de chaque classe d'apprentissage*

*Il généralise la classification à l'ensemble de l'image à partir de*

# CLASSIFICATION AUTOMATIQUE

*algorithme des k-moyennes (k-means)*

*Basé sur le critère de **distance minimale dans le domaine spectral***

- 1) segmentation de l'espace spectral en k sous-espaces  
*k* nombre fixé *a priori*

Itération de la boucle suivante:

- 2) calcul du barycentre de chaque sous-espace
- 3) Affectation des pixels à la classe dont le barycentre est le plus proche

Jusqu'à ce que

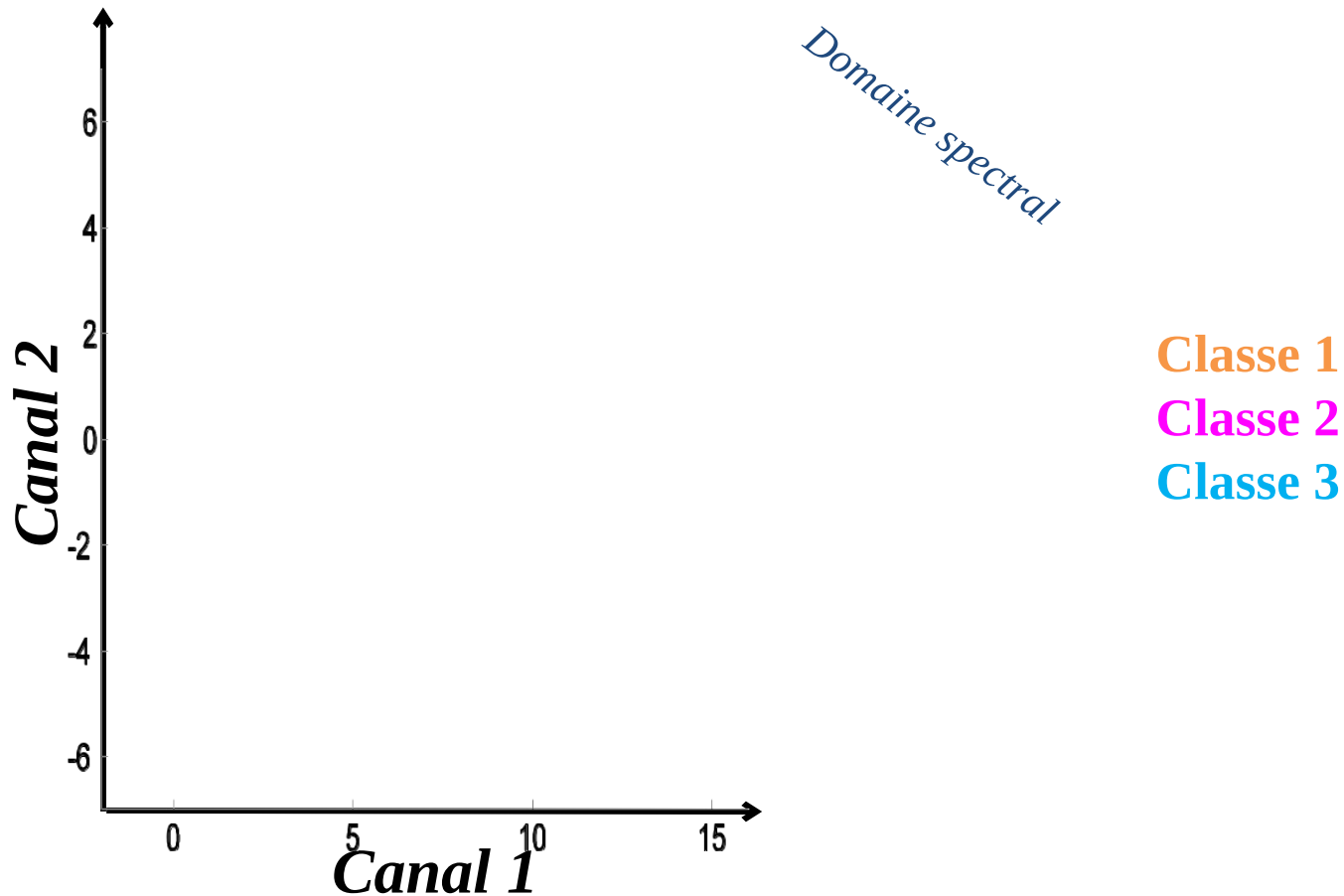
soit atteint le nombre d'itérations maximal (arbitraire *a priori*)

ou l'évolution de la position des barycentres  $< \varepsilon$  ( $\varepsilon$  seuil fixé *a priori*)

# ALGORITHME DU K-MEANS

*Exemple pour une image à 2 canaux et 3 classes*

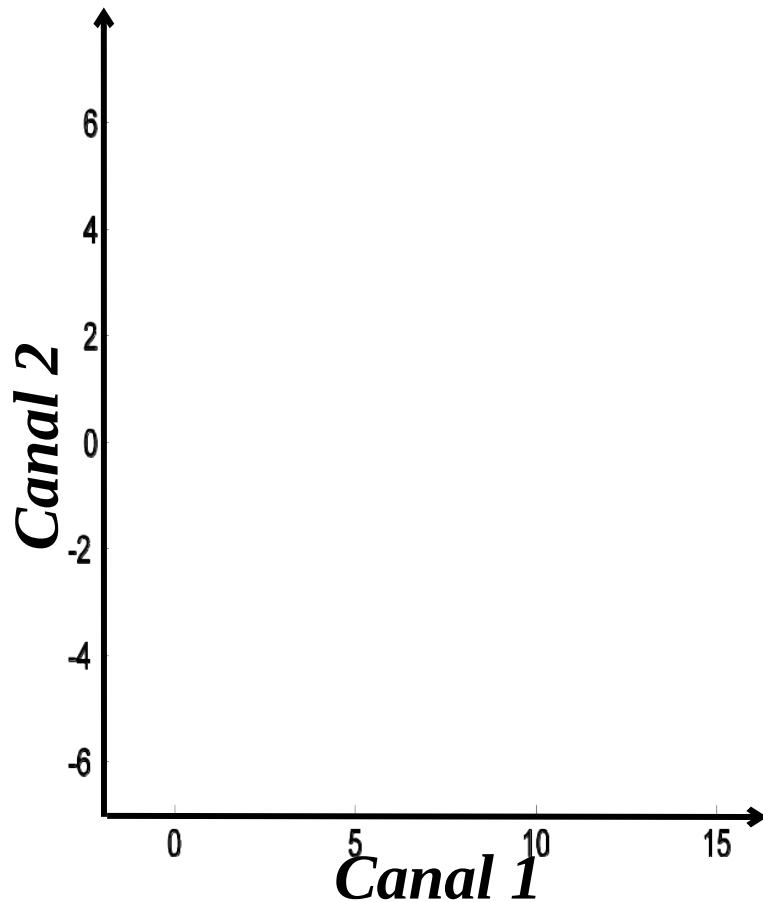
*Données à traiter*



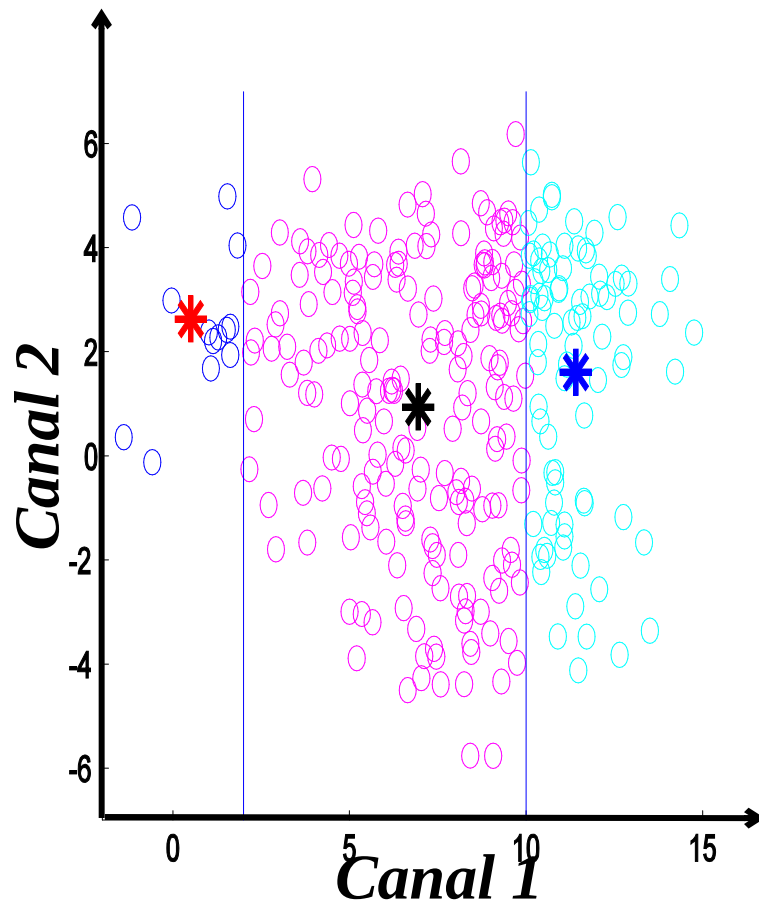
# ALGORITHME DU K-MEANS

*Exemple pour une image à 2 canaux et 3 classes*

*Données à traiter*



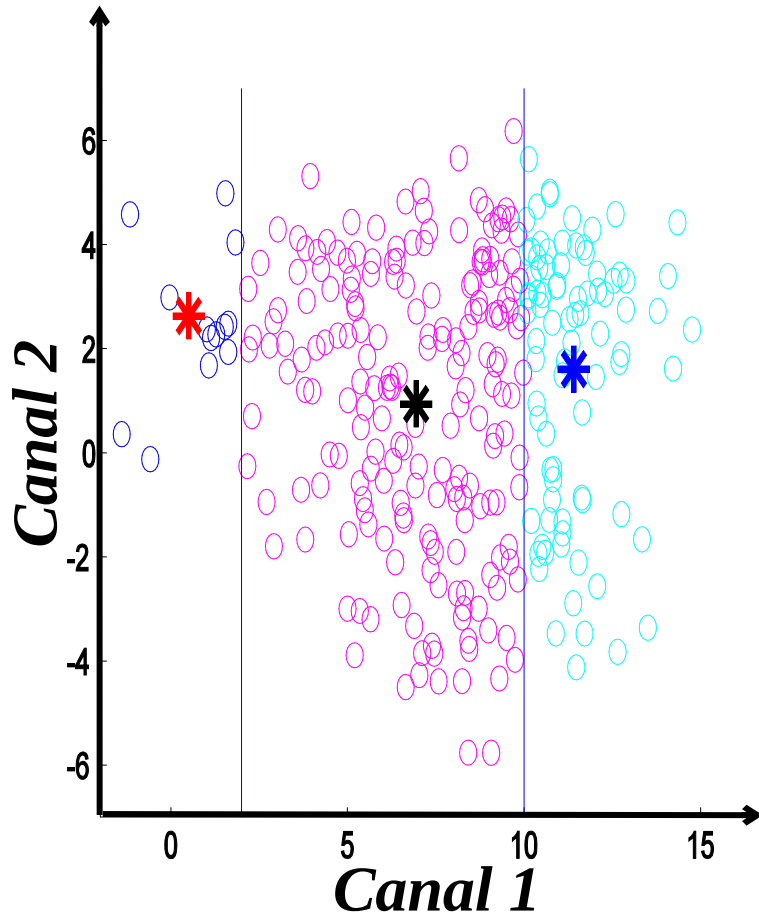
*Initialisation*



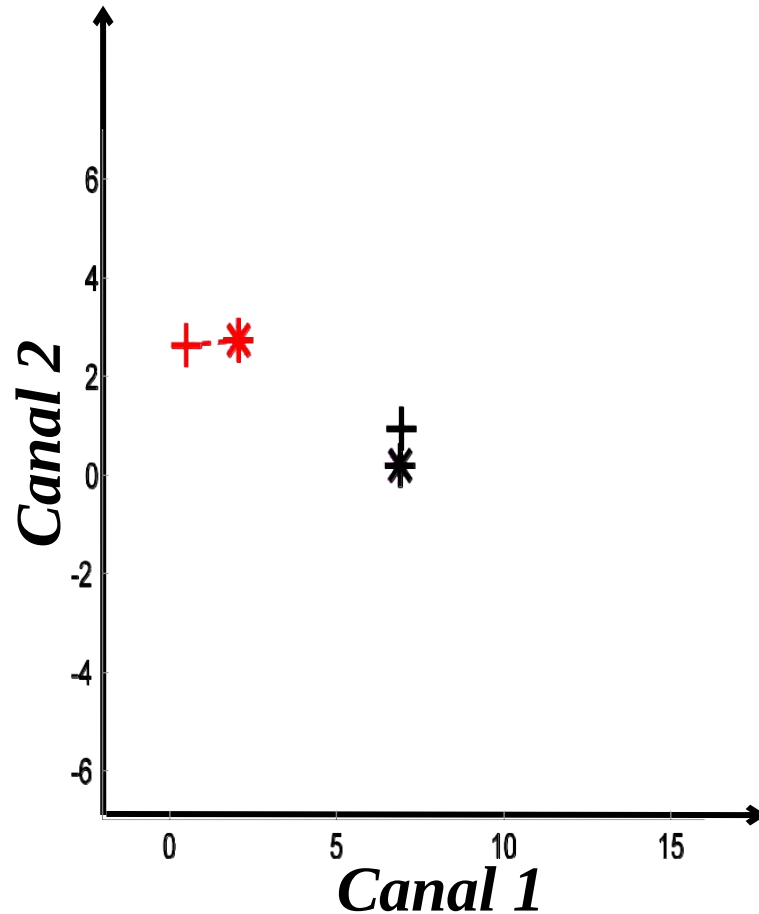
# ALGORITHME DU K-MEANS

*Exemple pour une image à 2 canaux et 3 classes*

*Initialisation*



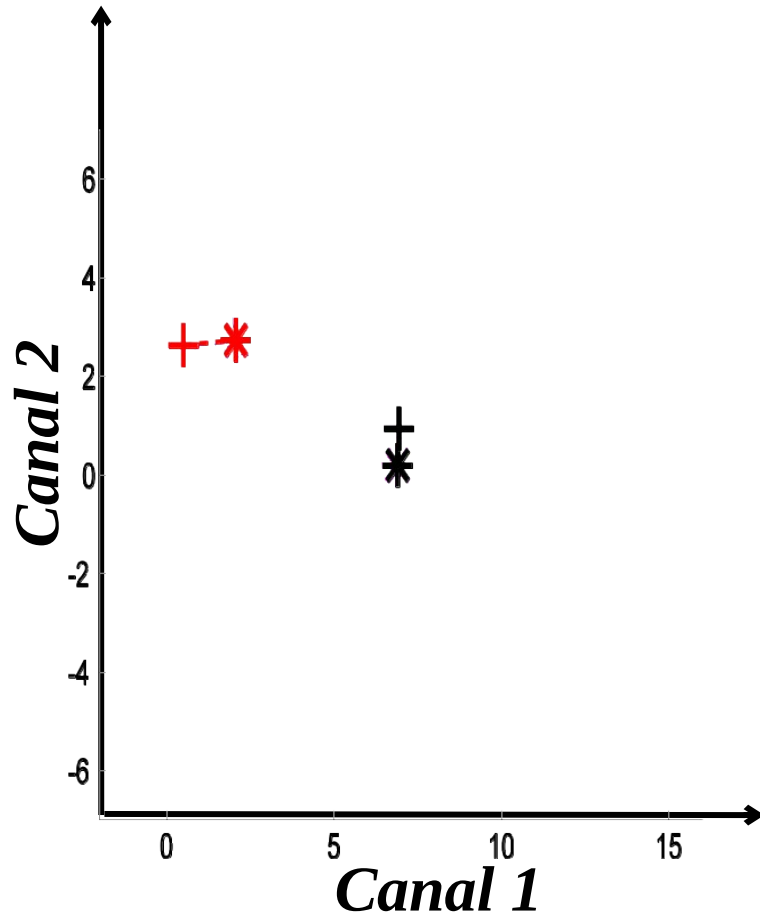
*1<sup>ère</sup> itération*



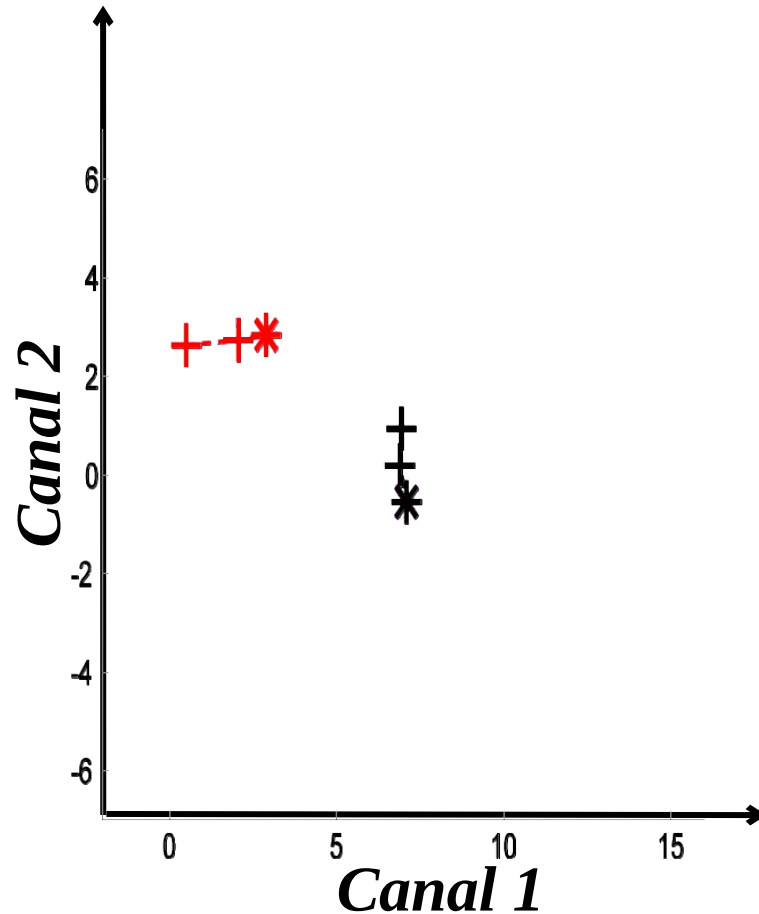
# ALGORITHME DU K-MEANS

*Exemple pour une image à 2 canaux et 3 classes*

*1<sup>ère</sup> itération*



*2<sup>ième</sup> itération*

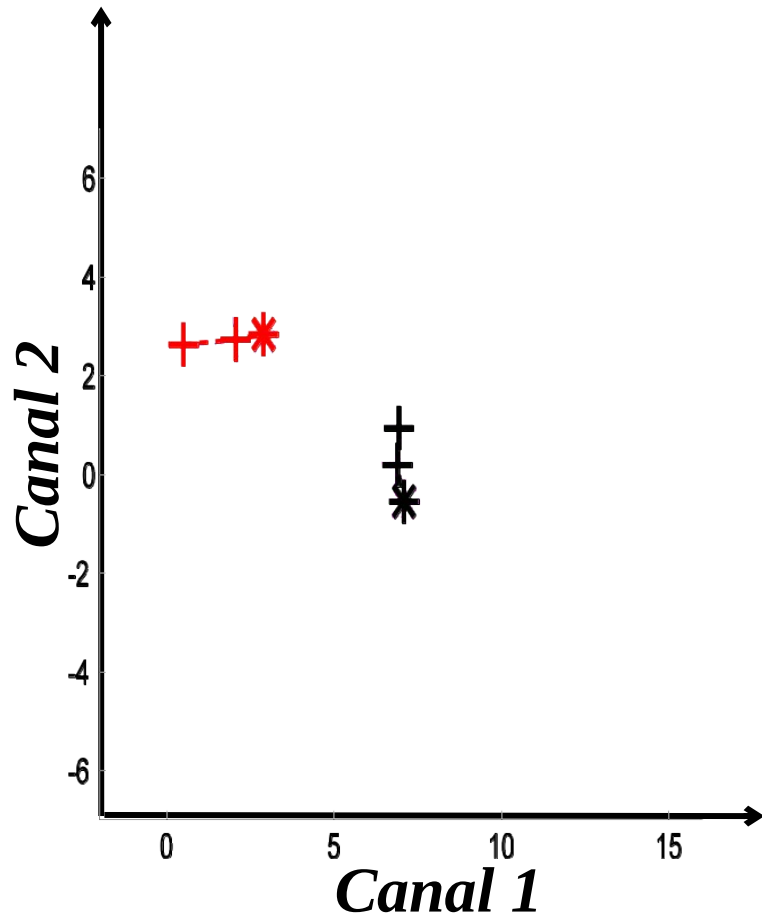




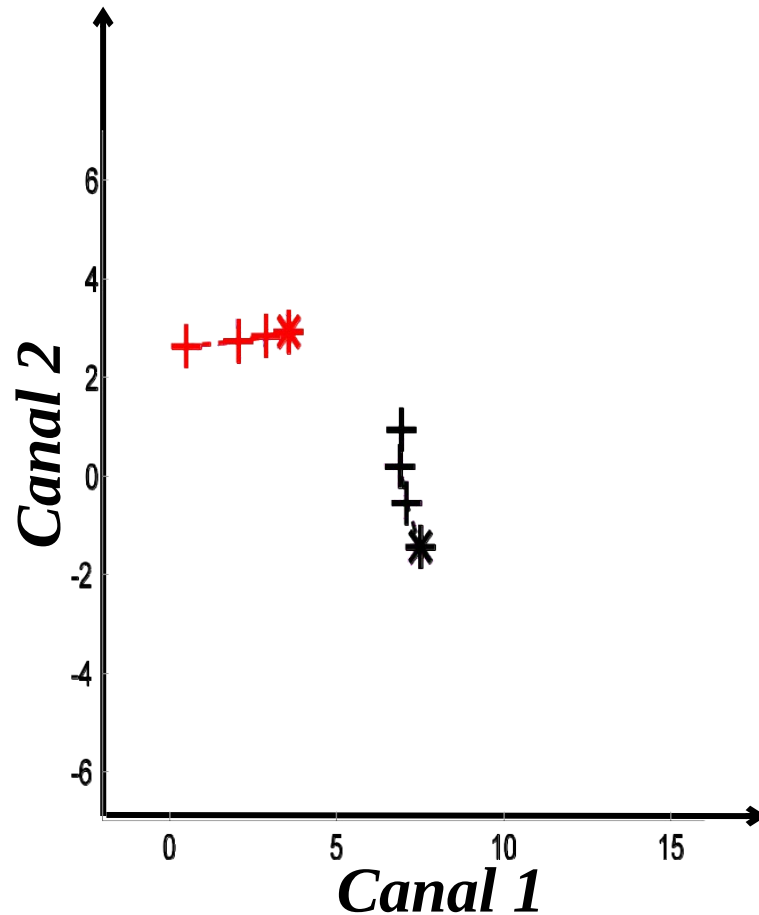
# ALGORITHME DU K-MEANS

*Exemple pour une image à 2 canaux et 3 classes*

*2<sup>ième</sup> itération*



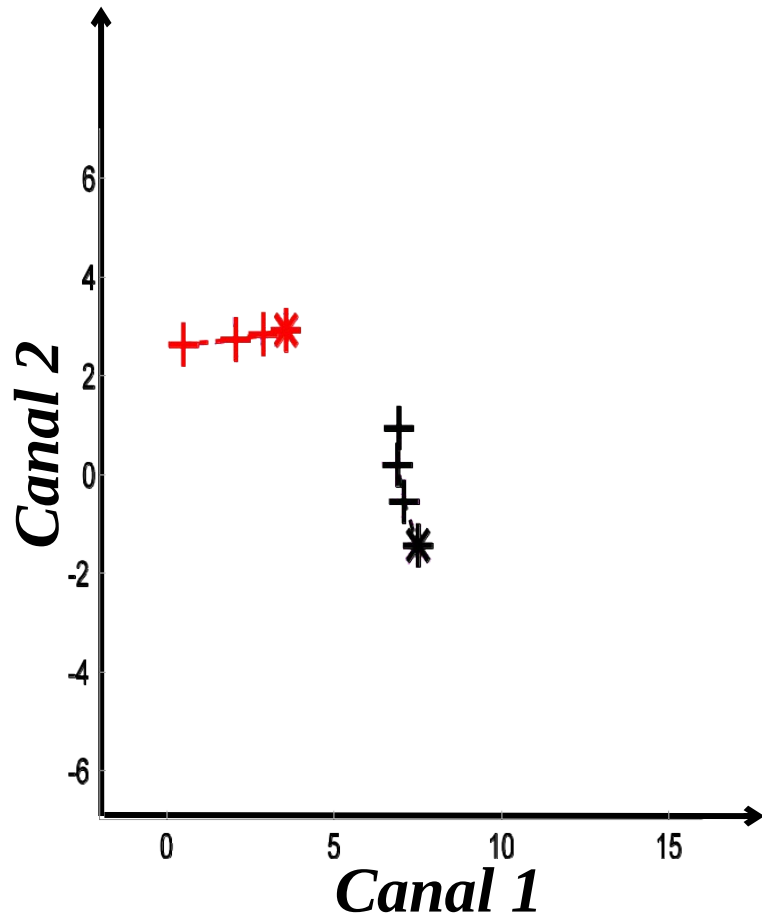
*3<sup>ième</sup> itération*



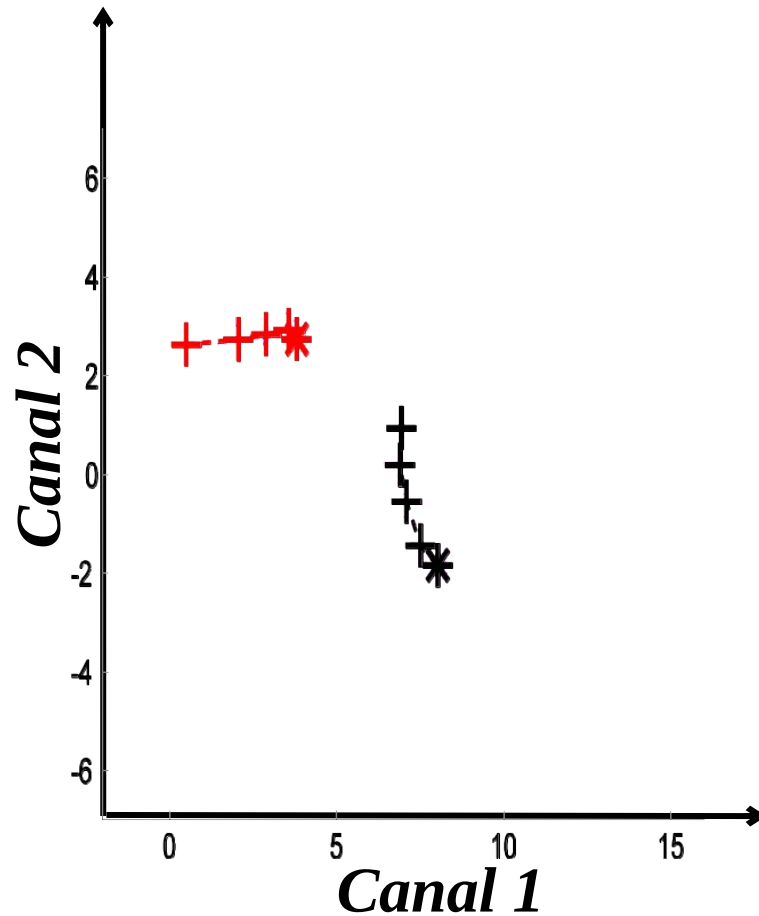
# ALGORITHME DU K-MEANS

*Exemple pour une image à 2 canaux et 3 classes*

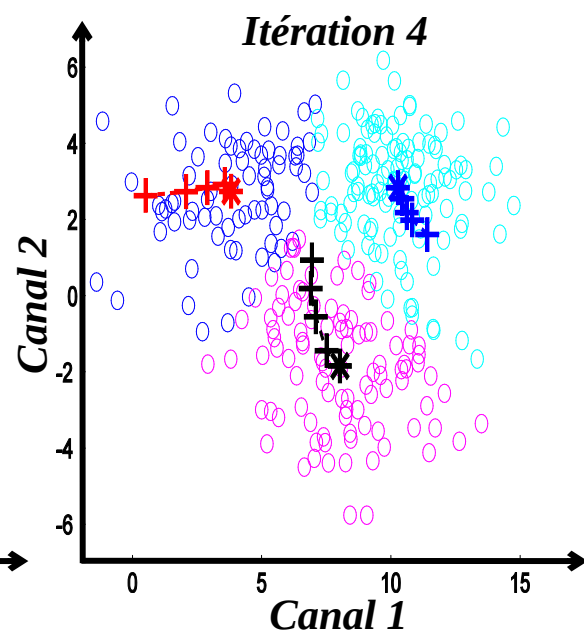
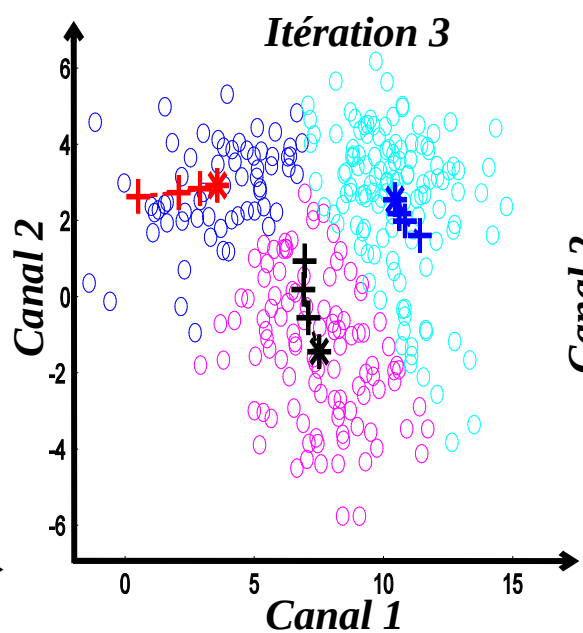
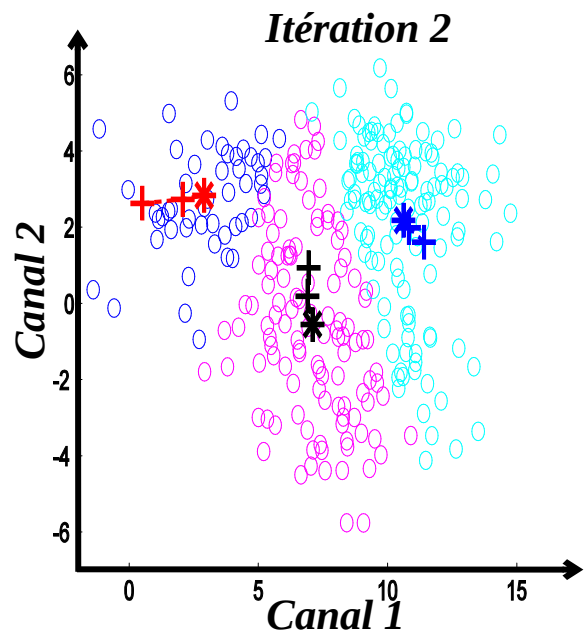
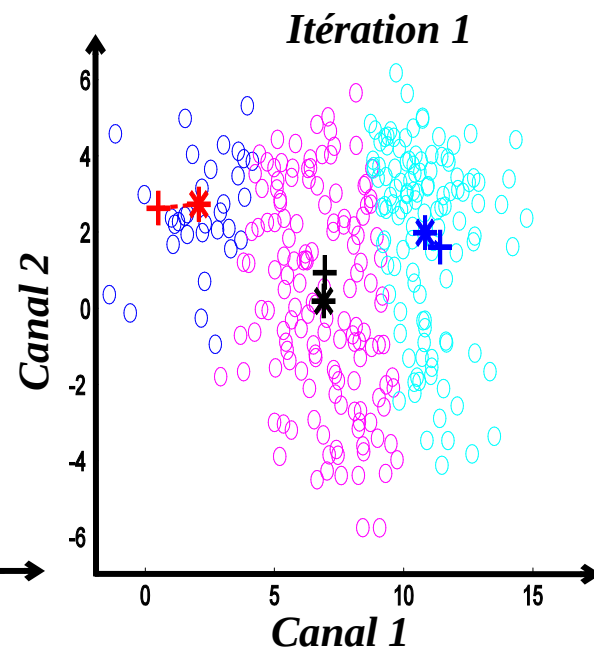
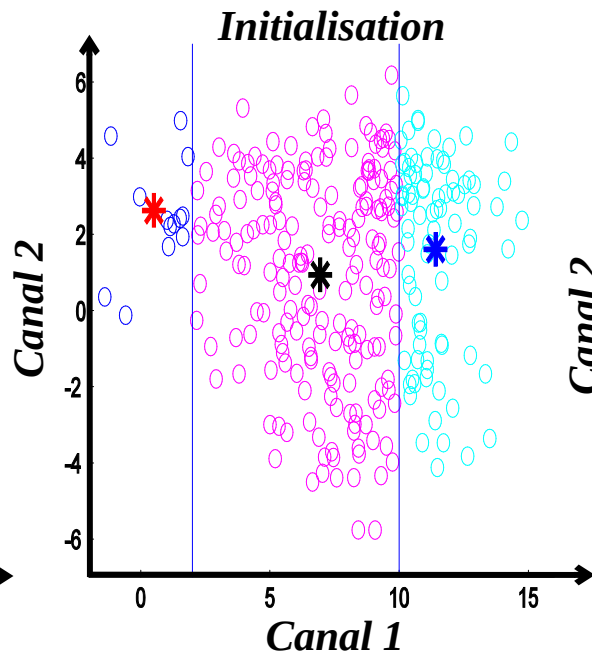
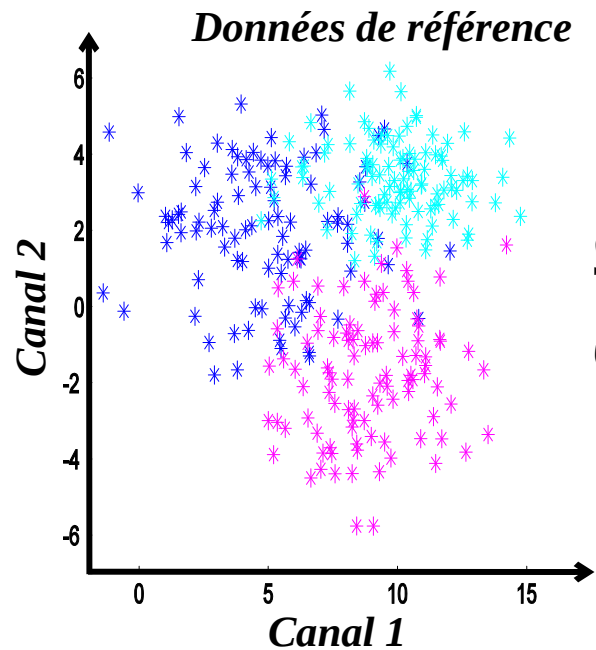
*3<sup>ème</sup> itération*



*4<sup>ème</sup> itération*



# ALGORITHME DU K-MEANS



# CLASSIFICATION SUPERVISEE

1) Un opérateur défini des **polygones** sur l'image

Plusieurs polygones représentatifs des différentes classes souhaitées

(photo-interprétation, relevé terrain,....)

Ces polygones sont divisés en 2 ensembles: - des **polygones d'entraînement**

(ou d'apprentissage)

(ou

- des **polygones de**

**validation**

2) a) L'algorithme analyse les propriétés spectrales associées à

chaque classe d'apprentissage

**Entraînement de la classification (polygones d'entraînement)**

□ Mise au point du **modèle de classification**

b) **Évaluation** qualité du modèle sur des polygones

# SUPERVISED CLASSIFICATION

1) The user defines **polygons** on the image

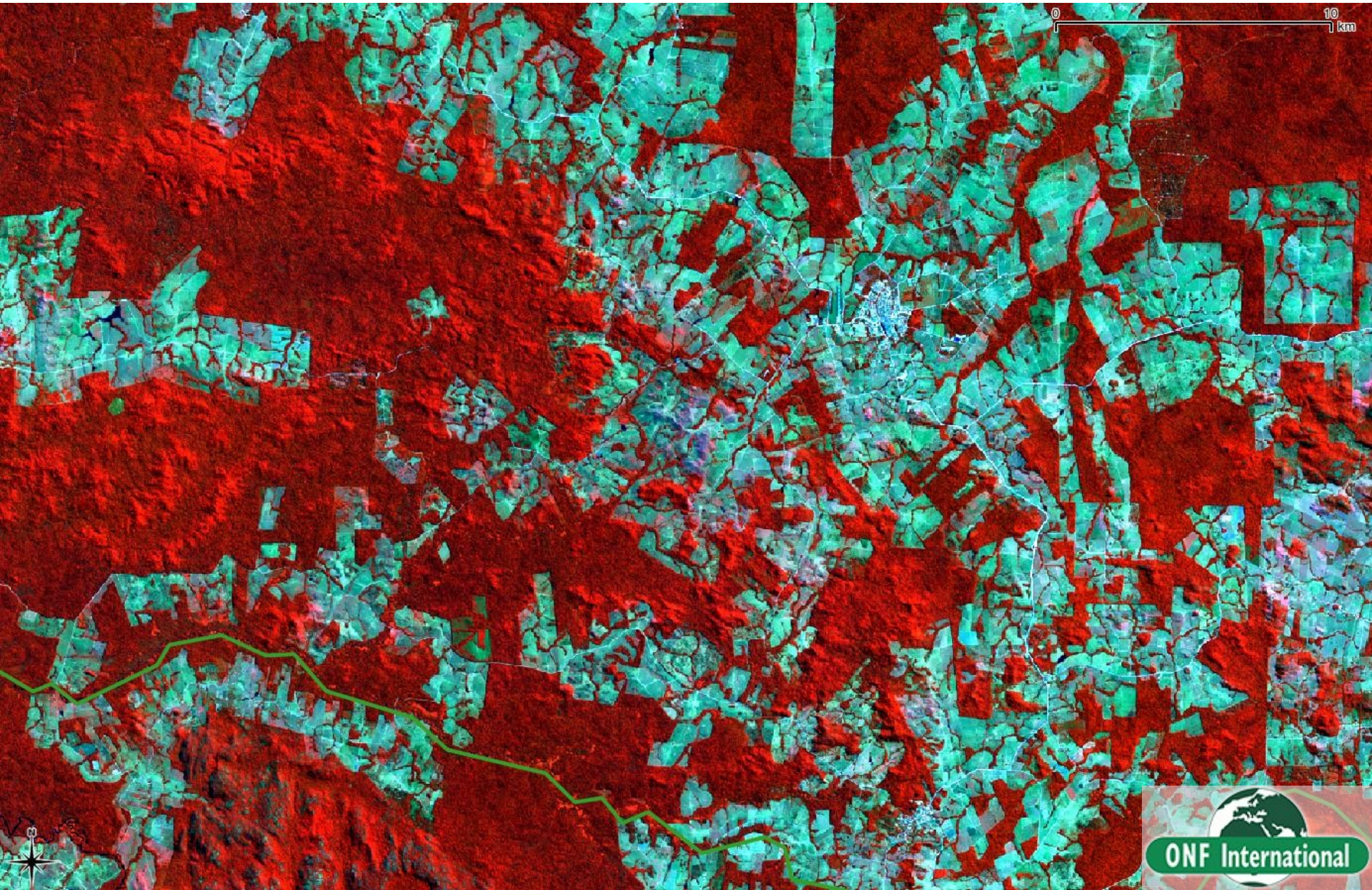
Several polygons representative of each classes to be determined

(photo-interpretation, ground survey,.....)

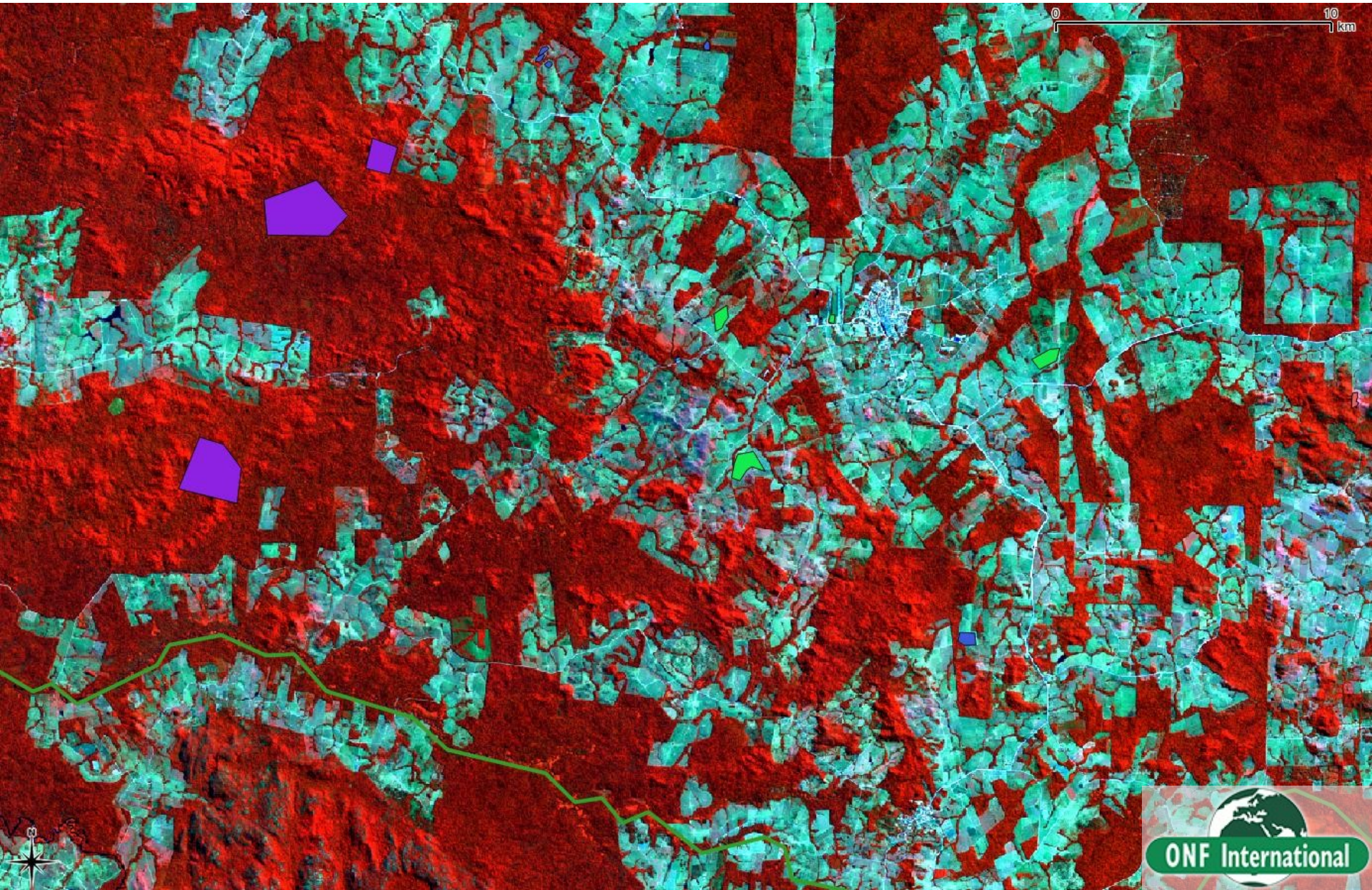
2) The algorithm analyses **the spectral** properties of each training class

**Classification training (training polygons)**

# *Image LANDSAT TM*



# *Image LANDSAT TM*




# SUPERVISED CLASSIFICATION

1) The user defines **polygons** on the image

Several polygons representative of each classes to be determined

(photo-interpretation, ground survey,.....)



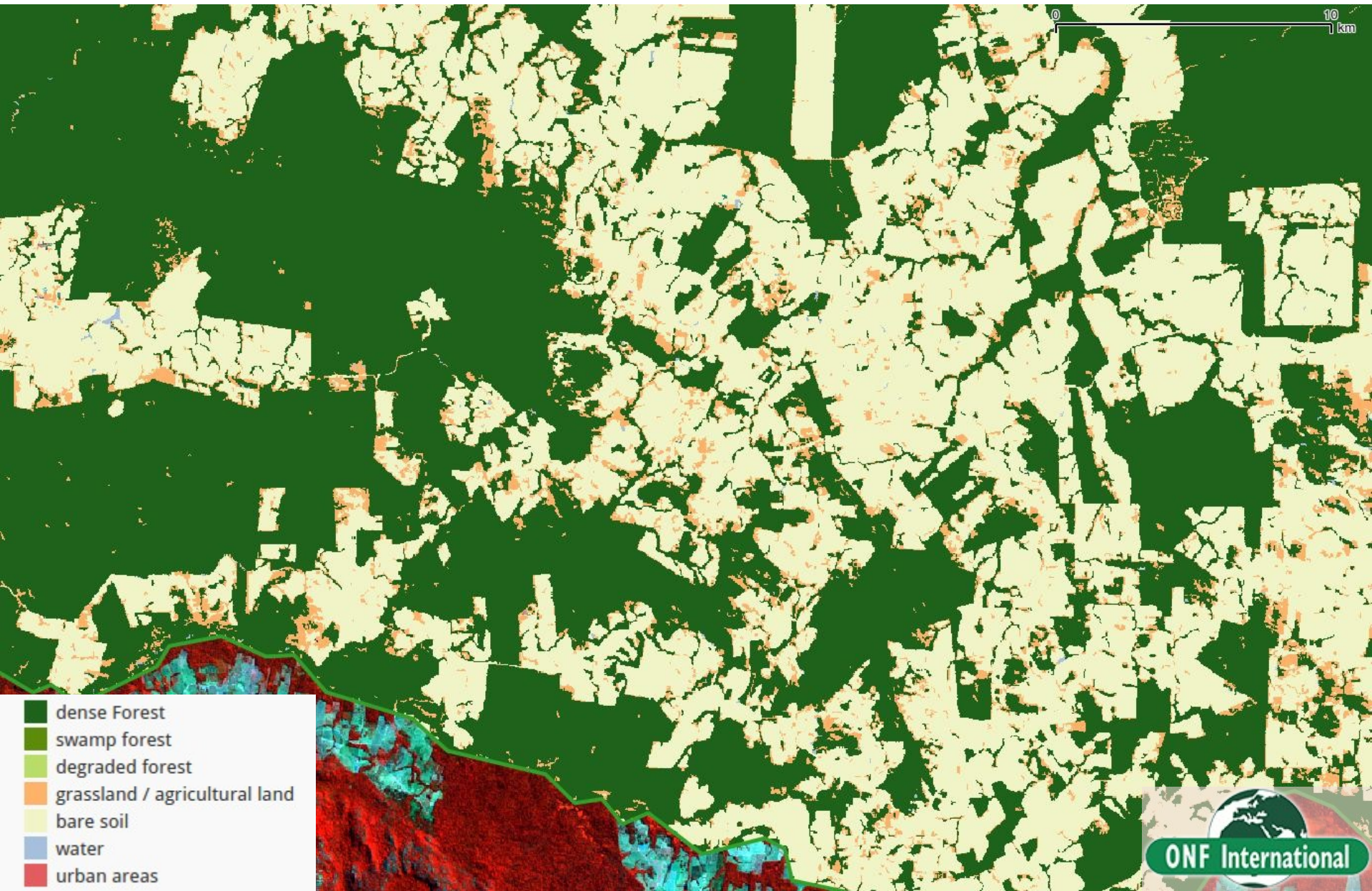
2) The algorithm analyses **the spectral** properties of each training class

**Classification training (training polygons)**

3) The algorithm generalises the training on the whole image

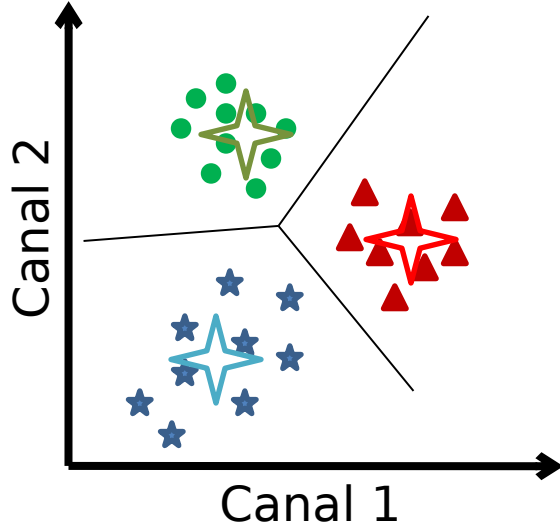


# *Résultat de classification*

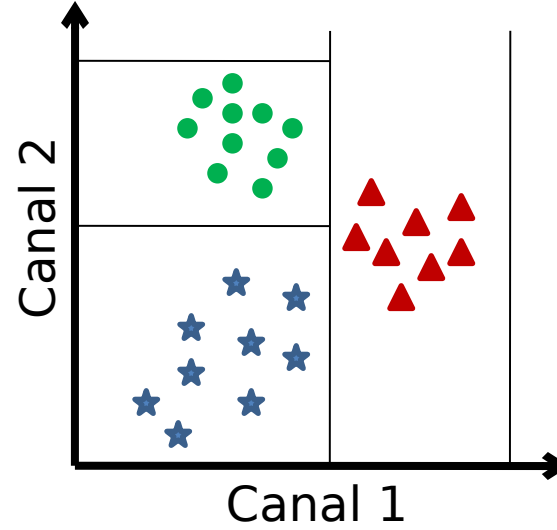


# Types d'algorithmes

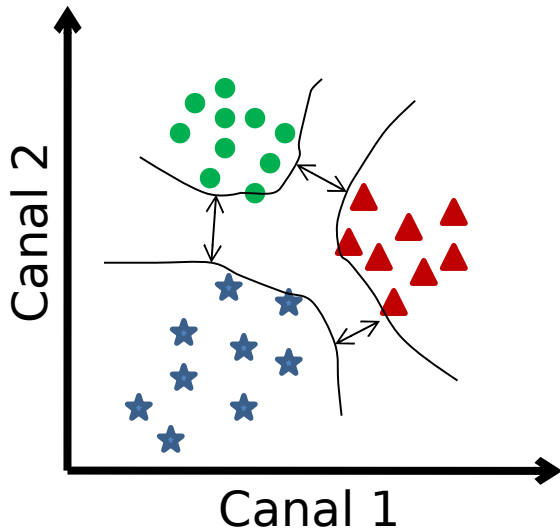
## *Distance minimale*



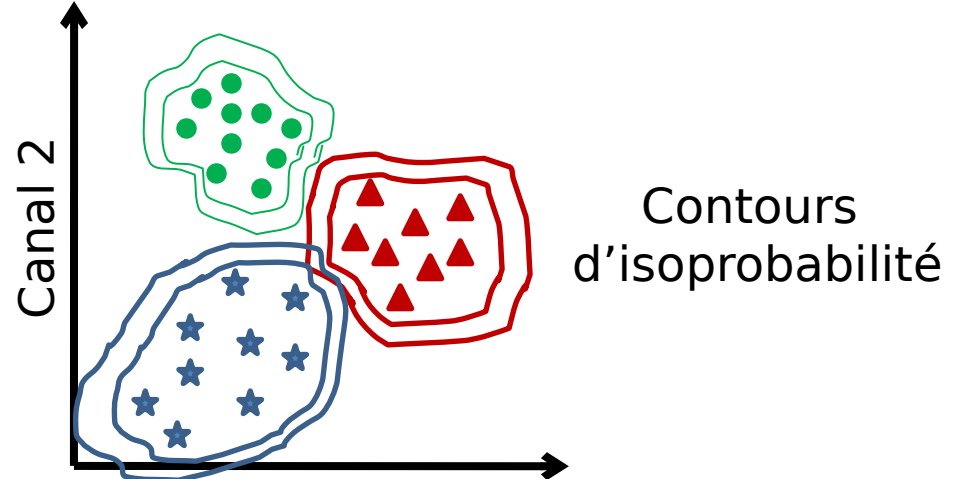
## *parallélépipèdes*



## *Vaste Marge (SVM)*



## *Maximum de Vraisemblance*



# Maximum de Vraisemblance

Basée sur le théorème de Bayes:  $P(C_i/g) = \frac{P(g/C_i)P(C_i)}{P(g)}$

$g$ : valeur du pixel (vecteur à  $N$  composantes)

$N$ : nombre de canaux de l'image

$C_1, C_2, \dots, C_k$ : classes recherchées

En fait,  $P(g) = \sum_{i=1}^k P(g/C_i)P(C_i)$  Indépendant des classes  $C_i$

D'où:  $g \in C_i \text{ si } P(g/C_i)P(C_i) \geq P(g/C_j)P(C_j) \quad \forall j=1,2,\dots,k$

Supposant que chacune des classes suive une

**distribution normale multivariée**, moyenne  $\mu$ , matrice de covariance

$$P(g/C_j) = \frac{1}{(2\pi)^{N/2} \sqrt{|\Sigma_j|}} \exp\left(-\frac{1}{2}(g - \mu)^T \Sigma_j^{-1}(g - \mu)\right)$$

$$d_j(g) = -\log|\Sigma_j| - (g - \mu)^T \Sigma_j^{-1}(g - \mu)$$

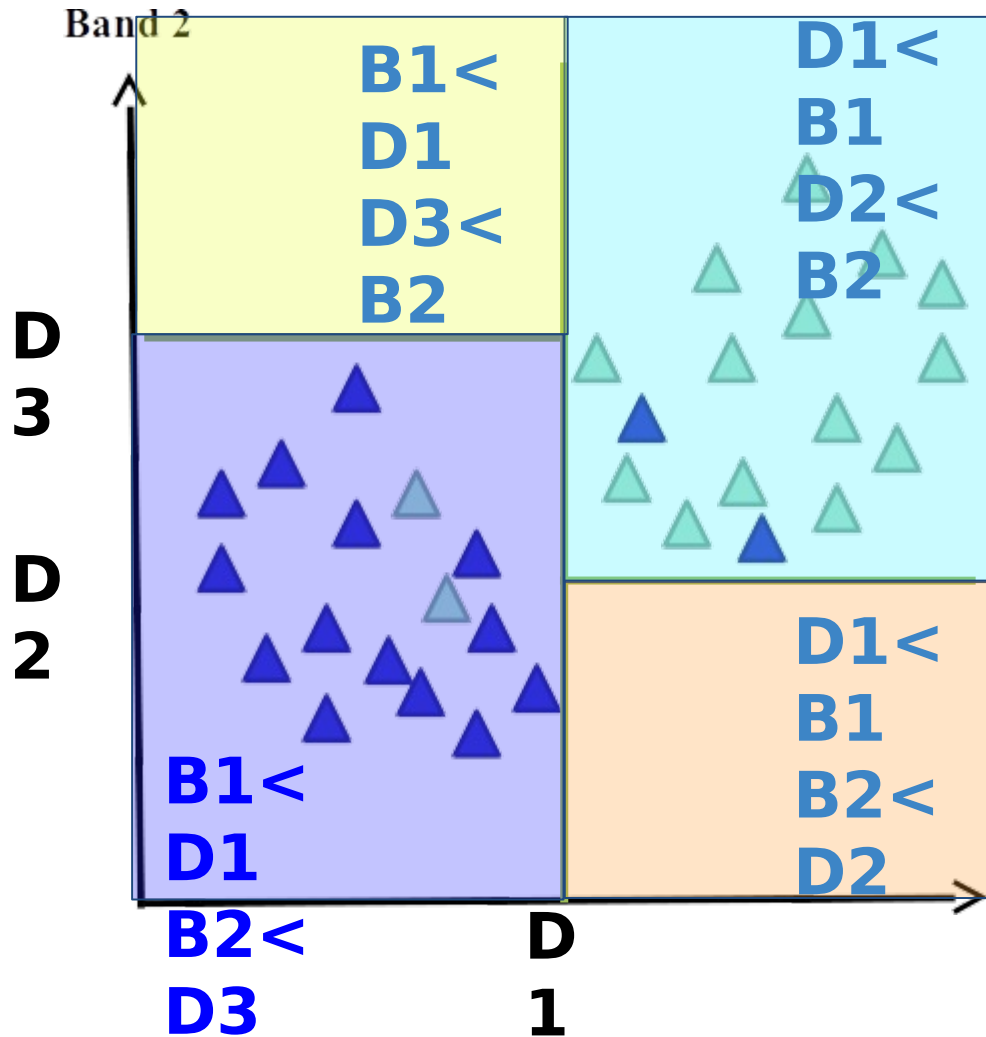
$$g \in C_i \text{ si } d_i(g) \geq d_j(g) \quad \forall j=1,2,\dots,k$$



**Rapide et bons résultats quand peu de bandes**

# Algorithme RANDOM FOREST

## Exemple d'un arbre de décision



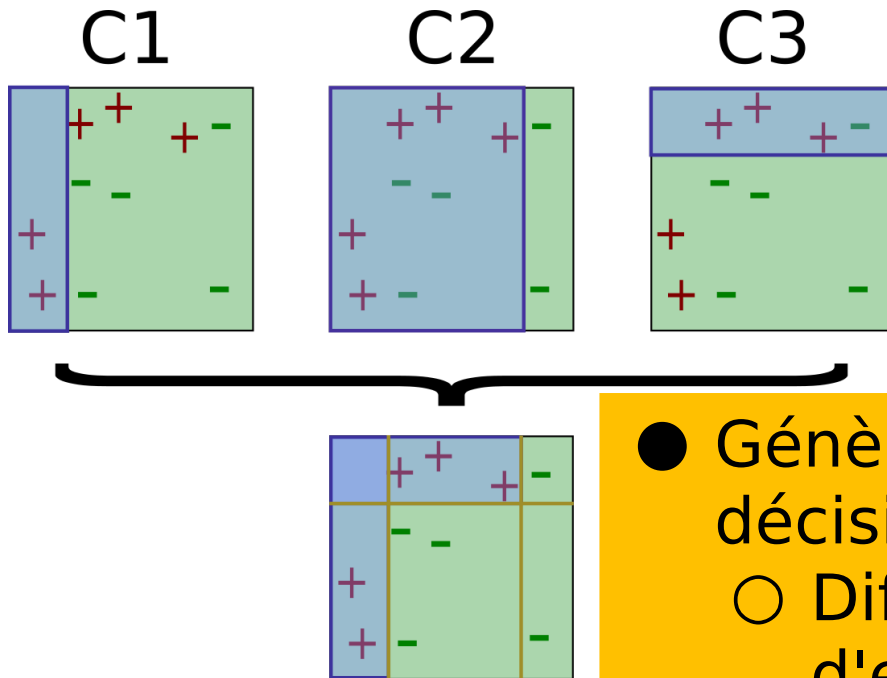
### *Limitation:*

- Sur apprentissage
- Frontière toujours linéaire

# Algorithme RANDOM FOREST

## Exemple de Random Forest (Machine Learning)

- Utilise la classification par arbre de décision
- Pour améliorer la prédiction, utilise le principe du boosting

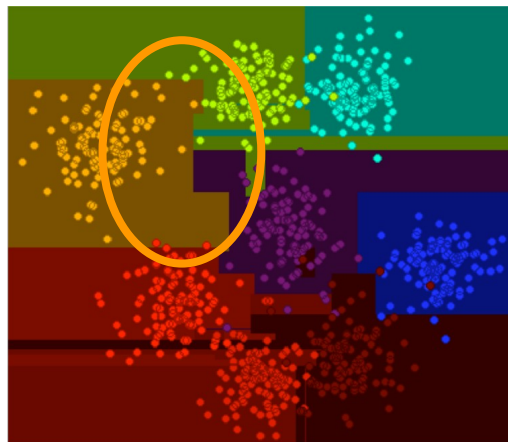


- 3 classifications de performances modérée
- Combinaison par vote majoritaire
- Résultats obtenus avec de meilleures performances

- Génère grand nombre d'arbres de décision (une Forêt)
  - Différents échantillons d'entraînements aléatoires
  - Différentes bandes aléatoires
- Combinaison par vote majoritaire

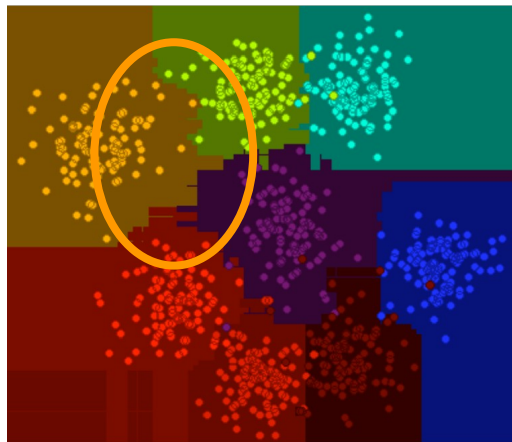
# Algorithme RANDOM FOREST

## Exemple de Random Forest (Machine Learning)



1 Tree

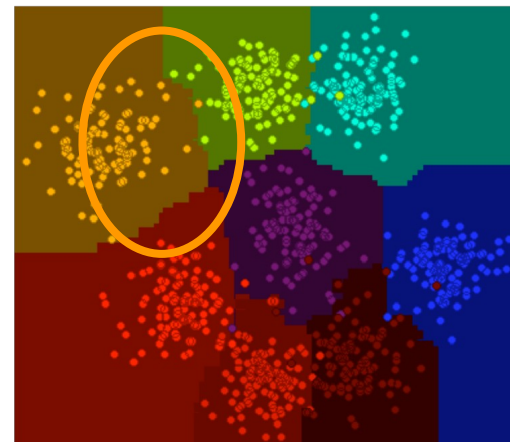
**A partir de 100 arbres  
les frontières de  
classes  
sont plus précise**



10 Trees



100 Trees



500 Trees

# Algorithme RANDOM FOREST

## En résumé, pourquoi Random Forest ?

- Nécessite ***un nombre raisonnable de points d'entraînement*** (mais toujours représentatif)
- Possible d'utiliser ***n'importe quels indices*** quel que soit ses statistique (Optical, Radar, Texture...)
- ***Autant d'indices*** que nécessaire : 1-10-100 et plus !!!
- ***Simple d'utilisation !***
- ***Très rapide***

# Comment définir ses ROIs ?

## 1. Choisir les classes

- a. Qu'est ce que je cherche ?*
- b. Quel taille font mes classes ?*
- c. A priori que peu discriminer mes données ?*

## 2. Créer ROI

- a. Doivent être représentatif (dans les données et dans les paysages)*
- b. Le plus homogène possible (pas trop non plus ;) )*





# Comment définir ses ROIs ?

## 1. Choisir les classes

- a. Qu'est ce que je cherche ?*
- b. Quel taille font mes classes ?*
- c. A priori que peu discriminer mes données ?*

## 2. Créer ROI

- a. Doivent être représentatif (dans les données et dans les paysages)*
- b. Le plus homogène possible (pas trop non plus ;) )*
- c. Faire simple*

*Décomposer en classe radiométrique (si possible)*

*Commencer avec les classes les plus simple*

*Ensuite ajouter de la complexité avec plus de classes*

- d. Processus itératif: Tester et améliorer les*

# Comment améliorer sa classification ?

**1. # de pts d'entraînement > 200pts/classe**

*Ajouter des polygones*

**2. Contrôler l'homogénéité des ROI**

*environ 80% du polygone doit correspondre à la classe annoncée*

**3. Les stats du classifieur et la qualité globale de la classification doivent avoir les mêmes tendance**

**Si excellentes stats  $\simeq$  ROI non**

*représentatifs → ajouter des polygones dans des zone d'erreurs*

**4. Si mauvais résultats**

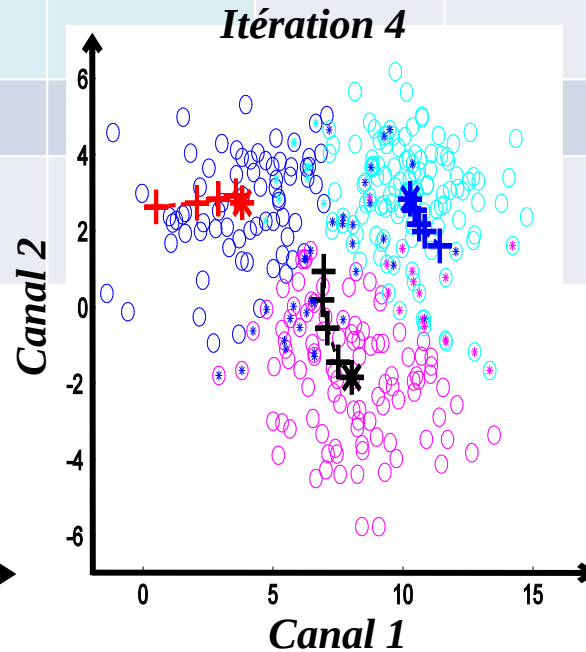
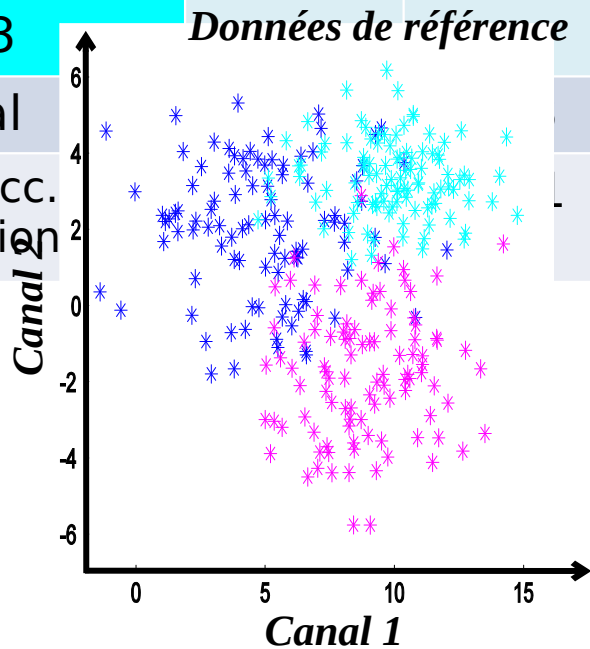
*ajouter des polygones dans des zone d'erreurs*

**5. Si toujours mauvais résultats**

# Evaluation de la performance d'une classification

## MATRICE DE CONFUSION

		Classes estimées				Prod. Acc. (Recall)
		Classe 1	Classe 2	Classe 3	Total	
Classes réelles	Classe 1	65	19	16	100	0.65
	Classe 2	0	84	16	100	0.84
	Classe 3	9	0	91	100	0.91
Total		74	103	103	280	
User Acc. (Precision)		0.879	0.816	0.883		



# QUALITE DE LA CLASSIFICATION

**MATRICE DE CONFUSION**

		Classes estimées				Prod. Acc. Recall
		Classe 1	Classe 2	Classe 3	Total	
Classes réelles	Classe 1	65	19	16	100	0.65
	Classe 2	0	84	16	100	0.84
	Classe 3	9	0	91	100	0.91
Total		74	103	123	300	
User Acc. Precision Précision totale		0.88	0.81	0.74		

$$\text{Producer Accuracy} = \frac{\# \text{ pixels bien classés}}{\# \text{ pixels classe réelle}}$$

**Recall**

$$\text{Omission error} = \frac{\# \text{ pixels omis}}{\# \text{ pixels classe réelle}}$$

**OE = 1 - PA**

$$\text{User Accuracy} = \frac{\# \text{ pixels bien classés}}{\# \text{ pixels classe estimée}}$$

**Precision**

$$\text{Commission error} = \frac{\# \text{ pixels mal classés}}{\# \text{ pixels classe estimée}}$$

**CE = 1 - UA**

# QUALITE DE LA CLASSIFICATION

**MATRICE DE CONFUSION**

		Classes estimées				Prod. Acc. Recall
		Classe 1	Classe 2	Classe 3	Total	
Classes réelles	Classe 1	65	19	16	100	0.65
	Classe 2	0	84	16	100	0.84
	Classe 3	9	0	91	100	0.91
Total		74	103	123	300	
User Acc. Précision totale		0.88	0.81	0.74		

Producer Accuracy =  $\frac{\text{\# pixels bien classés}}{\text{\# pixels classe réelle}}$   
**Recall**

*% qu'un pixel de la classe réelle soit bien classé (PD)*

User Accuracy =  $\frac{\text{\# pixels bien classés}}{\text{\# pixels classe estimée}}$   
**Precision**

*% qu'un pixel de la classe estimée soit bien classé (1 - FA)*

# QUALITE DE LA CLASSIFICATION

**MATRICE DE CONFUSION**

		Classes estimées				Prod. Acc. Recall
		Classe 1	Classe 2	Classe 3	Total	
Classes réelles	Classe 1	65	19	16	100	0.65
	Classe 2	0	84	16	100	0.84
	Classe 3	9	0	91	100	0.91
Total		74	103	123	300	
<b>Precision</b>		$\frac{74}{103} = 0.88$	$\frac{103}{123} = 0.81$	$\frac{123}{300} = 0.74$		
<b>Precision</b>		$\frac{0.88 + 0.81 + 0.74}{3} = 0.81$				

$$\text{Recall} = \frac{\text{Recall}_1 + \text{Recall}_2 + \text{Recall}_3}{3} = \frac{0.65 + 0.84 + 0.91}{3} = 0.80$$

$$\text{F-Score} = 2 \frac{\text{Precision} \cdot \text{Recal}}{\text{Precision} + \text{Recal}} = 2 \frac{0.81 \cdot 0.80}{0.81 + 0.80} = 0.80$$

# QUALITE DE LA CLASSIFICATION

**MATRICE DE CONFUSION**

		Classes estimées				
		Classe 1	Classe 2	Classe 3	Total	Prod. Acc.
Classes réelles	Classe 1	65	19	16	100	0.65
	Classe 2	0	84	16	100	0.84
	Classe 3	9	0	91	100	0.91
Total		74	103	123	300	

Un pixel de l'image classe 1) =  $100 / 300 = 33\%$  Un pixel de la classif classe 1) =  $74 / 300 = 25\%$

Un pixel de la classe 1 soit bien classé par chance) =  $0.33 * 0.25 = 8\%$

*bon accord par chance*

$$P_e = \frac{100 \cdot 74}{300^2} + \frac{100 \cdot 103}{300^2} + \frac{100 \cdot 123}{300^2} = 33\%$$

*bon accord observé*

$$P_0 = \frac{65 + 84 + 91}{300} = 80\%$$

$$Kappa = \frac{P_0 - P_e}{1 - P_e} = 73\%$$

**Comparaison entre résultats obtenus et d'une classification totalement aléatoire**

# QUALITE DE LA CLASSIFICATION

## MATRICE DE CONFUSION

		Classes estimées				
		Classe 1	Classe 2	Classe 3	Total	Prod. Acc.
Classes réelles	Classe 1	65	19	16	100	0.65
	Classe 2	0	84	16	100	0.84
	Classe 3	9	0	91	100	0.91
Vrais Positifs 1 = 65		74			103	
Faux Négatifs 1 = 19 + 16 = 35		123			300	
User Acc 1 = 9		0.88			0.81	
Faux Positifs 1 = 9		0.74			0.74	

Vrais Négatifs 1 = 84 + 16 + 91 = 191

Vrais Négatifs 2 = 65 + 9 + 16 + 91 = 181

Vrais Positifs 3 = 91

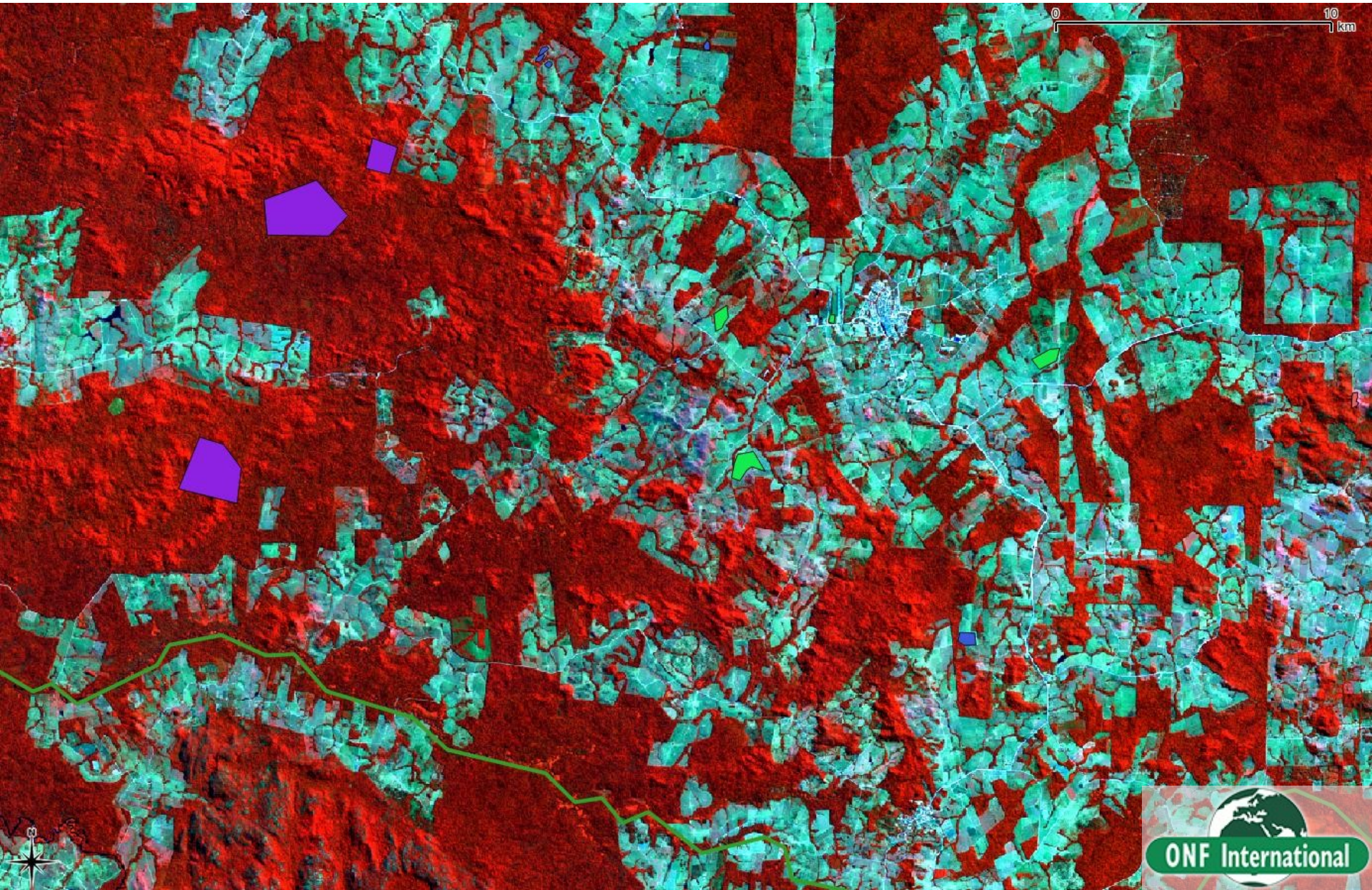
Faux Négatifs 3 = 16

Faux Positifs 3 = 16 + 16 = 32

Vrais Négatifs 3 = 65 + 19 + 84 = 168



# *Image LANDSAT TM*

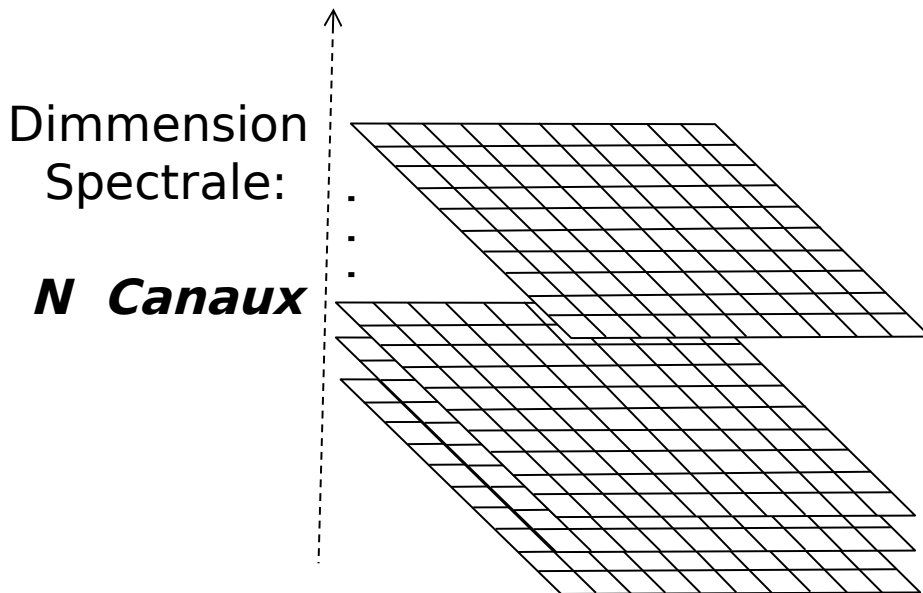


# REDUCTION DE DIMENSION SPECTRALE

Exemple: image hyperspectrale 50 canaux  
1.99 - 2.48  $\mu\text{m}$   
Résolution spatiale: 10 nm

**Perception œil: 3 parmi N canaux**

Bandes: 12 - 22 - 31



**Image aéroportée AVIRIS, Cuprite,**

# ***REDUCTION DE DIMENSION SPECTRALE***

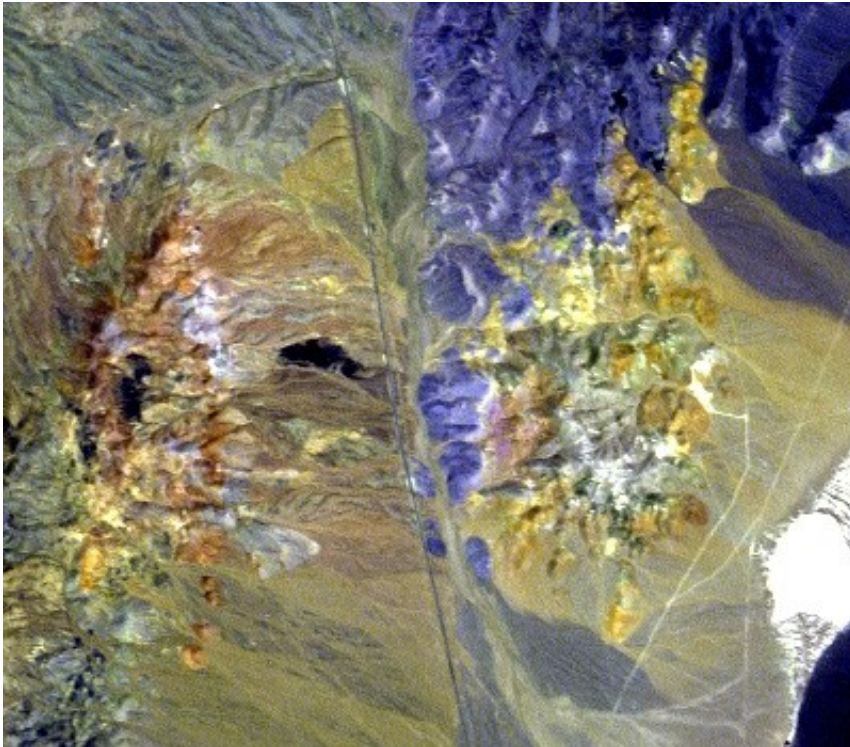
Exemple: image hyperspectrale 50 canaux

1.99 - 2.48  $\mu m$

*Résolution spatiale: 10 nm*

Perception œil: 3 parmi N canaux

Bandes: 1 - 11 - 50



Bandes: 12 - 22 - 31



***Image aéroportée AVIRIS, Cuprite,***

# ***REDUCTION DE DIMENSION SPECTRALE***

Exemple: image hyperspectrale 50 canaux

1.99 - 2.48  $\mu m$

*Résolution spatiale: 10 nm*

Perception œil: 3 parmi N canaux

Bandes: 1 - 2 - 3



Bandes: 12 - 22 - 31

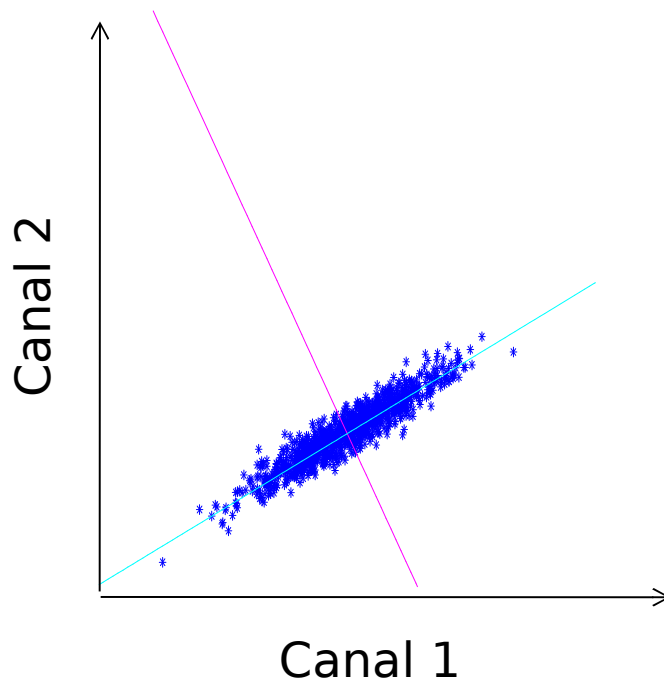


***Image aéroportée AVIRIS, Cuprite,***

# REDUCTION DE DIMENSION SPECTRALE

## **Analyse en Composantes Principales (ACP)**

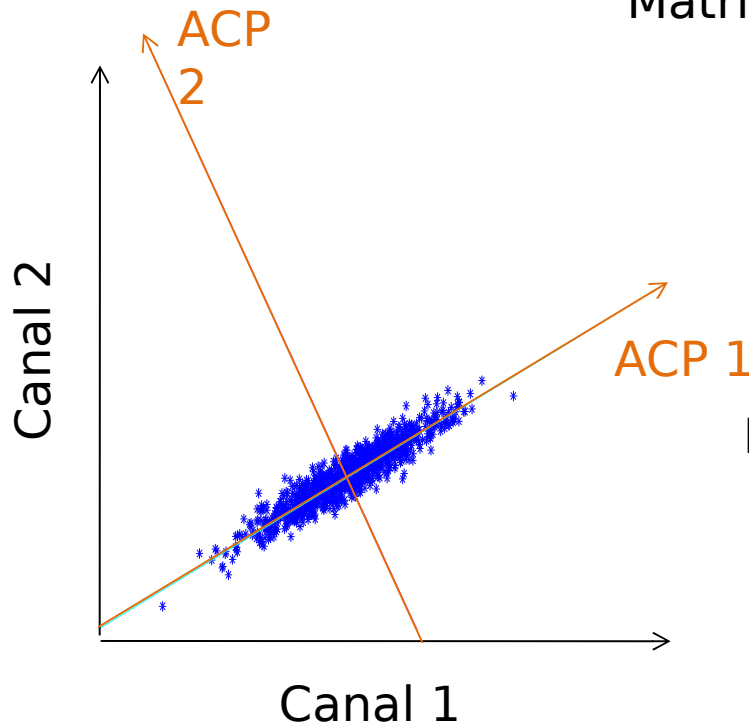
réduire la dimension spectrale de l'image pour ne garder que  
les composantes contenant le **plus d'information = plus grande variance**



# REDUCTION DE DIMENSION SPECTRALE

## Analyse en Composantes Principales (ACP)

Minimiser la dimension spectrale de l'image pour ne garder que  
les composantes contenant le **plus d'information = plus grande variance**



Matrice de covariance (Canal1, Canal2)

$$C \begin{pmatrix} 99.8 & 49.7 \\ 49.7 & 28.7 \end{pmatrix}$$

Matrice de covariance (ACP1, ACP2)

$$C \begin{pmatrix} 125.4 & 0 \\ 0 & 3.1 \end{pmatrix}$$

# REDUCTION DE DIMENSION SPECTRALE

## Analyse en Composantes

### Principales (ACP)

réduire la dimension spectrale de l'image pour ne garder que les composantes contenant le **plus d'information = plus grande variance**

**Principe: diagonaliser matrice de covariance entre canaux**

Exemple: image à **4 canaux**

$$\text{Matrice de covariance: } C = \begin{pmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & C_{34} \\ C_{41} & C_{42} & C_{43} & C_{44} \end{pmatrix}$$

(matrice symétrique)

$$C_{ij} = \text{cov}(x_i, x_j) = \frac{1}{\text{nb\_pixels}} \sum_{k=1}^{\text{nb\_pixels}} (x_{i,k} - \bar{x}_i)(x_{j,k} - \bar{x}_j)$$

$$C_{ii} = \text{var}(x_i) = \frac{1}{\text{nb\_pixels}} \sum_{k=1}^{\text{nb\_pixels}} (x_{i,k} - \bar{x}_i)^2$$

# REDUCTION DE DIMENSION SPECTRALE

## Analyse en Composantes Principales (ACP)

$$P^{-1}CP = \begin{pmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 \\ 0 & 0 & 0 & \lambda_4 \end{pmatrix} \quad \begin{array}{l} \lambda_i \text{ Valeurs propres de } C \\ \lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \lambda_4 \end{array}$$

$$P = \begin{pmatrix} \vec{V}_1 & \vec{V}_2 & \vec{V}_3 & \vec{V}_4 \\ P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P_{41} & P_{42} & P_{43} & P_{44} \end{pmatrix}, \quad \text{vecteurs propres de } C$$

$P$  : Matrice de changement de base  
Matrice de vecteurs propres

$$\left( \vec{e}_1, \vec{e}_2, \vec{e}_3, \vec{e}_4 \right) \rightarrow \left( \vec{e}'_1, \vec{e}'_2, \vec{e}'_3, \vec{e}'_4 \right)$$

$$\vec{e}'_2 = P_{12} \vec{e}_1 + P_{22} \vec{e}_2 + P_{32} \vec{e}_3 + P_{42} \vec{e}_4$$



# REDUCTION DE DIMENSION SPECTRALE

## Analyse en Composantes Principales (ACP)

$$P^{-1}CP = \begin{pmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 \\ 0 & 0 & 0 & \lambda_4 \end{pmatrix} \quad \begin{array}{l} \lambda_i \text{ Valeurs propres de } C \\ \lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \lambda_4 \end{array}$$

$\frac{\lambda_i}{\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4}$  Donne la proportion (%) de la variance portée par l'a

# REDUCTION DE DIMENSION SPECTRALE

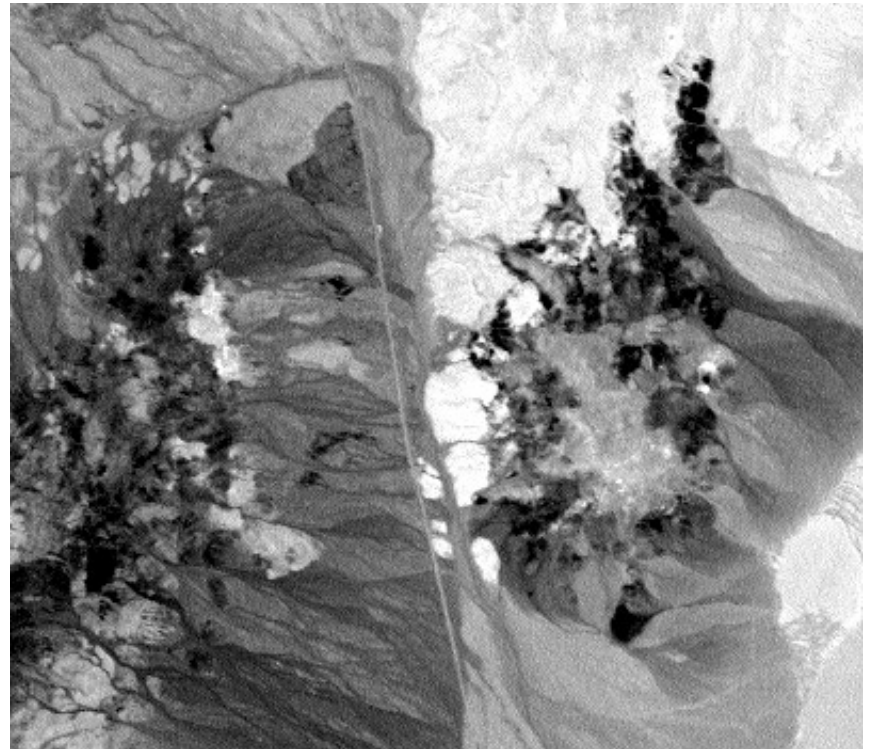
## Analyse en Composantes Principales (ACP)

réduire la dimension spectrale de l'image pour ne garder que les composantes contenant le plus d'information

ACP - Axe 1 (90%)



ACP - Axe 2 (6%)



$$(\lambda_1=58665, \lambda_2=4139, \sum \lambda_i = 6518)$$

# ***REDUCTION DE DIMENSION SPECTRALE***

Exemple: image hyperspectrale 50 canaux

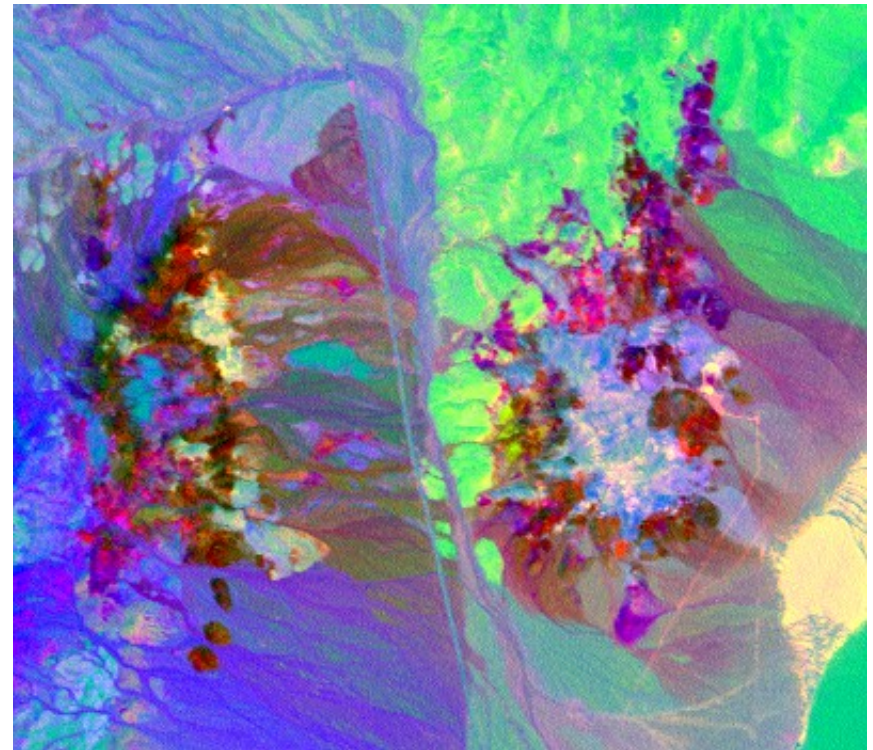
1.99 - 2.48  $\mu m$

*Résolution spatiale: 10 nm*

Perception œil: 3 parmi N canaux

Bandes 12 - 22 - 31

ACP: Axe 1 - Axe 2 - Axe 3 (98%)



**RELIQUATS**

# QUELQUES RÉSULTATS

## ■ Détection de zones de végétation (RapidEye - 5m)

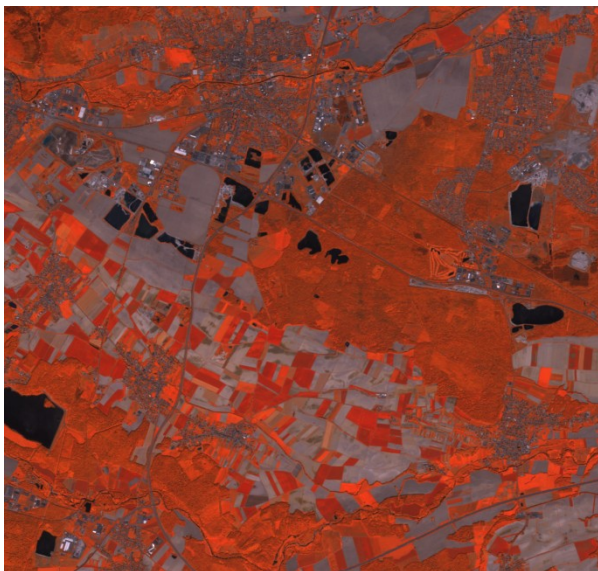
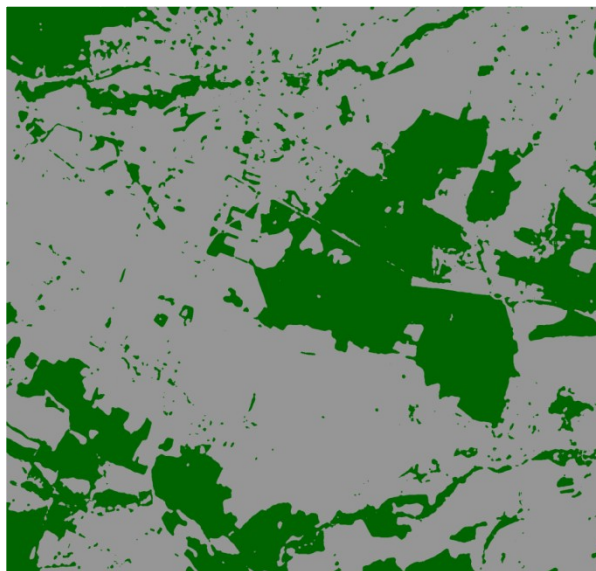
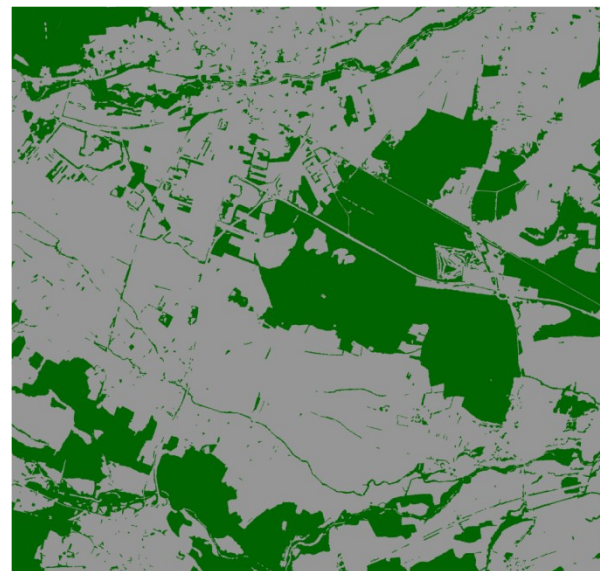


Image RapidEye  
{Red, rededge, ired}



Classification



Vérité Terrain (BDU)

# QUELQUES RÉSULTATS

## ■ Discrimination feuillus/résineux (RapidEye - 5m)

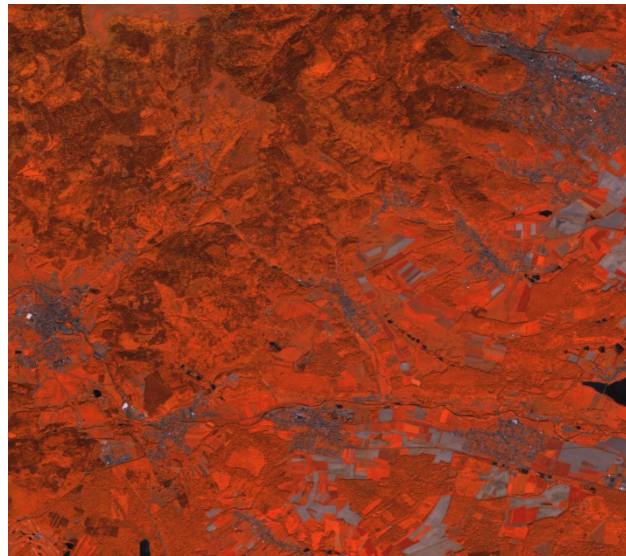
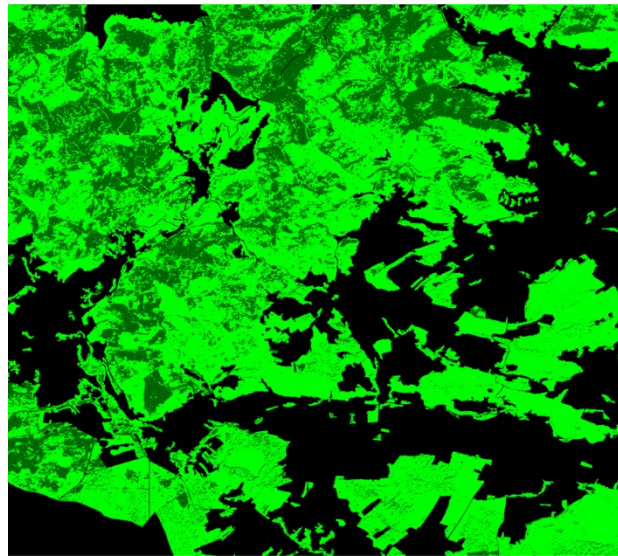
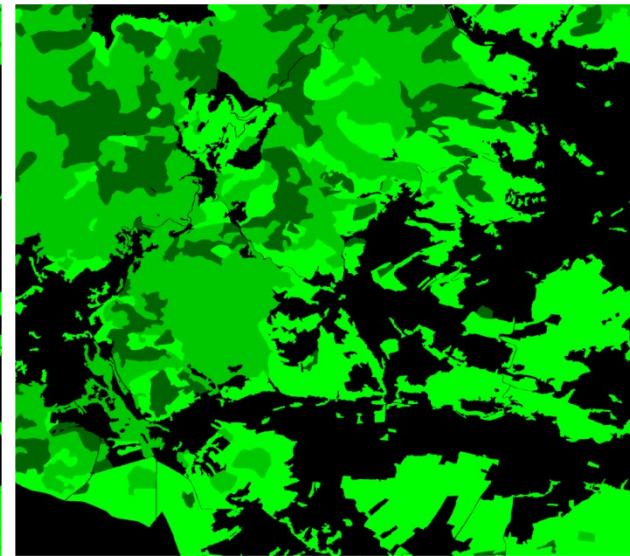


Image RapidEye  
{Red, reledge, ired}



Classification  
Feuillus - résineux



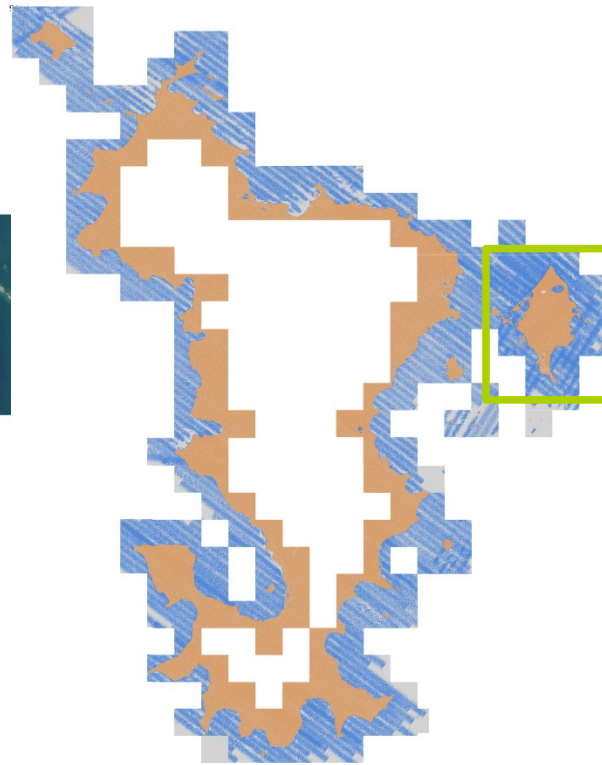
Vérité Terrain (BDU)  
Feuillus – mixte – résineux

# QUELQUES RÉSULTATS

- Classification Terre/Mer en zones littorales par lidar aéroporté



Mayotte  
(800 Mpts)



Classification Terre-Mer

