

Analyse d'images Pleiades PHR avec l'ORFEO ToolBox

Manuel Grizonnet (CNES), Julien Michel (CNES), Jordi Inglada (CNES/CESBIO)

Décembre 2013

Contents

1	Avant-propos	2
1.1	Données	2
1.2	Les logiciels	2
2	Exercices	4
2.1	Monteverdi et OTB applications	4
2.1.1	Description	4
2.1.2	Étapes	4
2.1.3	Solutions	6
2.2	Pré-traitements : corrections géométriques et radiométriques	9
2.2.1	Description	9
2.2.2	Étapes	10
2.2.3	Solutions	11
2.3	Segmentation	12
2.3.1	Description	12
2.3.2	Étapes	12
2.3.3	Solutions	15
2.4	Extraction de primitives	20
2.4.1	Description	20
2.4.2	Étapes	20
2.4.3	Solutions	23
2.5	Apprentissage et classification au niveau pixel	26
2.5.1	Description	26
2.5.2	Étapes	26
2.5.3	Solutions	28
2.6	Apprentissage et classification sur des objets	29
2.6.1	Description	29
2.6.2	Étapes	29
2.6.3	Solutions	31

1 Avant-propos

1.1 Données

Les images utilisées pour cette formation sont des extraits de produits de démonstration Pléiades mis à disposition à des fins d'évaluation. Afin d'obtenir les produits complets, vous pouvez aller sur ce site web. Les produits utilisés sont :

- Pleiades Pan-sharpened ORTHO Compression REGULAR
- Pleiades TRISTEREO Bundle PRIMARY

Ils sont sous copyright CNES.

Les autres données nécessaires pour les exercices ainsi que les solutions des exercices sont fournies dans le paquet *data*.

1.2 Les logiciels

Afin de réaliser les exercices, les logiciels suivants sont nécessaires :

- **Orfeo Toolbox** 3.14 ou plus récent avec les applications
- **Monteverdi** 1.6 ou plus récent
- **QGis** 1.8 ou plus récent

Pour ce qui concerne l'installation de l'**Orfeo Toolbox** et de **Monteverdi**, vous pouvez consulter l'Orfeo Toolbox Cookbook.

Pour installer **QGis** consultez la documentation du logiciel qui est disponible sur le site du projet.

Creative Commons Attribution-ShareAlike 3.0 Unported License



2 Exercices

2.1 Monteverdi et OTB applications

2.1.1 Description

Résumé Cet exercice permet de se familiariser avec l'utilisation de **Monteverdi** et les outils disponibles dans **OTB Applications**.

Pré-requis

- Connaissances de base en télédétection et traitement des images
- Être capable de lancer des programmes en ligne de commande

Objectifs

- Visualisation de données dans **Monteverdi**
- Traitements simples avec **Monteverdi**
- Traitements simples avec **OTB Applications** en mode graphique
- Traitements simples avec **OTB Applications** en ligne de commande

2.1.2 Étapes

Monteverdi: lecture de données et visualisation Dans cette partie de l'exercice, l'image suivante sera utilisée : `phr_pxs_melbourne.tif`

1. Lancement de **Monteverdi** : ouvrir un terminal et taper

```
$ monteverdi
```

2. Ouvrir l'image (avec le menu *File/Open dataset*)
3. Trouver une façon d'afficher l'image (il y a 2 façons de faire).
4. Se déplacer dans l'image :
 - (a) Changer la zone affichée à pleine résolution;
 - (b) Changer la zone affichée dans la fenêtre de zoom;
 - (c) Changer le niveau de zoom;
 - (d) Quelles sont les informations affichées à propos du pixel qui se trouve sous le pointeur de la souris?
5. À l'aide du panneau de configuration de la visualisation dans l'onglet *Setup* :
 - (a) Changer la configuration pour visualiser la 4ème bande;
 - (b) Affichage en fausses couleurs (mettre le proche infra-rouge dans l'affichage rouge, la bande rouge dans l'affichage vert et la bande verte dans l'affichage bleu);



- (c) Changer la configuration pour revenir à un affichage en couleurs naturelles;
- (d) Améliorer le contraste en utilisant la zone à pleine résolution;
- (e) Améliorer le contraste en utilisant la zone de zoom;
- (f) Revenir aux ajustements de contraste par défaut.

Trucs et astuces:

- Les flèches du clavier peuvent être utilisées pour naviguer dans les images.
- L'ordre des bandes Pléiades est : rouge, vert, bleu et proche infra-rouge.

Monteverdi : traitements basiques Dans cette partie l'image suivante est utilisée :

`phr_xs_melbourne.tif`

1. Ouvrez l'image dans **Monteverdi**.
2. Trouvez le module *BandMath* dans les menus. Ouvrez l'image dans ce module. Quel type de traitement est proposé?
3. À l'aide de ce module, calculez le NDVI de l'image :

$$NDVI = \frac{PIR - Rouge}{PIR + Rouge} \quad (1)$$

4. En utilisant le même module, créez un masque des pixels dont la valeur dans le PIR est inférieure à 150. Affichez l'image et le masque dans le même *viewer*
5. Créez un masque correspondant aux pixels dont la valeur est supérieure à 1000 dans toutes les bandes spectrales. Visualisez le masque et l'image dans le même *viewer*.
6. Utilisez le module *Concatenate* afin de créer une image RVB composite avec le masque des valeurs élevées dans le rouge, le masque des valeurs faibles de PIR dans le bleu et le NDVI dans le vert.
7. Utilisez le module *Color Mapping* afin de créer une image composite RVB du NDVI qui permet une meilleure interprétation visuelle.

Trucs et astuces :

- Les valeurs du NDVI sont comprises entre -1 et +1, mais la vraie dynamique des valeurs observées peut être beaucoup plus faible.

OTB applications : mode graphique et ligne de commande

1. Lancez la commande suivante :

```
$ otbcli_OrthoRectification
```

et ensuite

```
$ otbgui_OrthoRectification
```

Qu'observez-vous?

2. Combien d'applications sont actuellement disponibles dans **OTB Applications**?
3. Comment peut-on obtenir de l'aide et de la documentation à propos des applications?

OTB applications : traitements simples Utilisez les applications afin de reproduire les résultats des étapes 3 à 7 de la section sur Monteverdi 2.1.2.

Pour aller plus loin

1. Comment peut-on lire ou visualiser des images directement depuis la ligne de commande avec **Monteverdi**?
2. Y-a-t'il d'autres façons de calculer des indices comme le NDVI avec les **OTB Applications**?
3. Renseignez-vous sur l'utilisation des **OTB Applications** depuis *Python* et écrivez un script en *Python* qui réalise les mêmes traitements que dans la section 2.1.2

2.1.3 Solutions

Monteverdi: lecture de données et visualisation

1. Point 3 Pour visualiser une image dans **Monteverdi**, vous pouvez :
 - soit faire clic droit sur l'image et sélectionner *Display in viewer*,
 - soit utiliser le menu *Visualization/Viewer*, choisir l'image et cliquer sur *OK*.

La 2^e option permet de visualiser plusieurs images simultanément.
2. Point 4 La zone en bas à gauche affiche des informations sur l'image et sur le pixel qui se trouve sous le pointeur de la souris :
 - la position courante dans l'image,
 - la taille de l'image,
 - la bande affichée,
 - les valeurs des pixels,
 - la résolution (spacing),
 - la position géographique si elle est disponible,
 - le nom de l'endroit s'il est disponible.

Monteverdi : traitements basiques

1. Point 2 Le module **BandMath** permet de faire des calculs avec les bandes des images en utilisant la syntaxe de *muParser*.
2. Point 3 Afin de calculer le NDVI, on peut utiliser l'expression **BandMath** suivante :

$$(im1b4-im1b1) / (im1b4+im1b1)$$

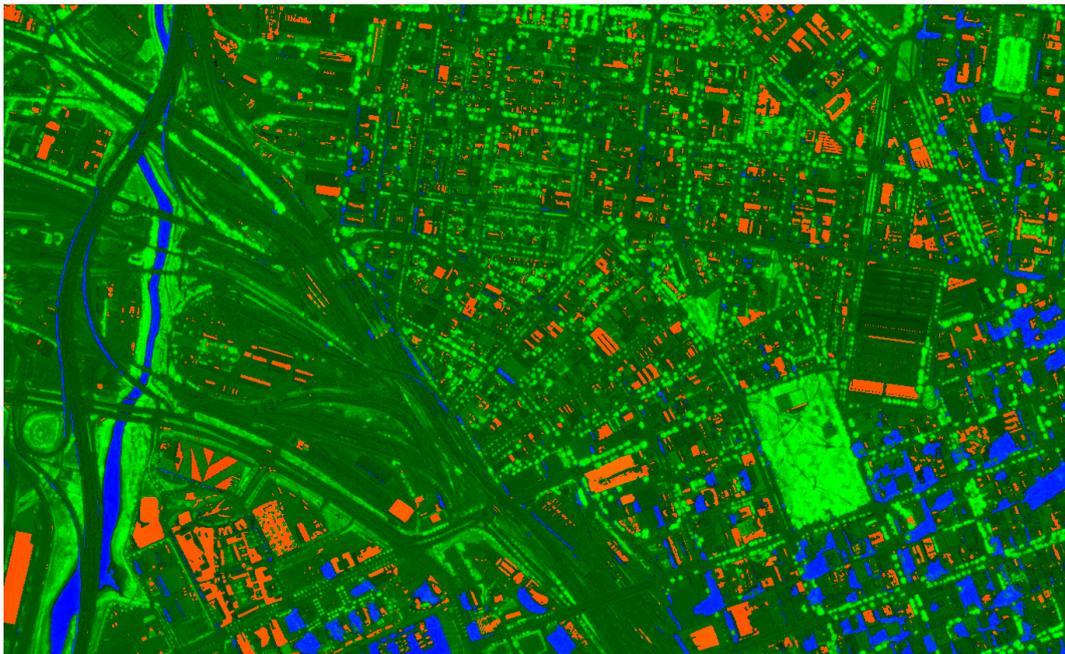
3. Point 4 Afin de construire le masque des pixels dont le compte numérique dans le PIR est inférieur à 150, on peut utiliser l'expression **BandMath** suivante :

$$if(im1b4 < 150, 255, 0)$$

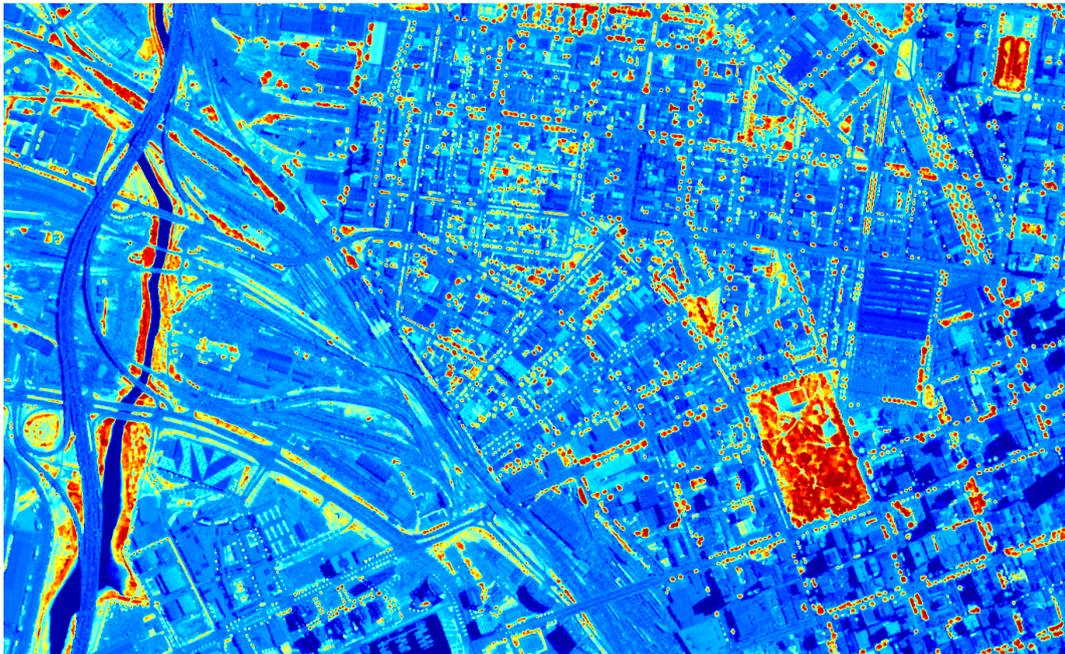

- Point 5 Pour générer le masque des pixels dont le compte numérique est supérieur à 1000 dans toutes les bandes spectrales, on peut utiliser l'expression **BandMath** suivante :

```
if (min(im1b1, im1b2, im1b3, im1b4) > 1000, 255, 0)
```

- Point 6 Dans la barre de menu, choisir *File/Concatenate images* et choisir les sorties des 3 modules **BandMath**. L'image résultante peut être affichée dans le *Viewer* et elle ressemble à ceci :



- Point 7 Dans la barre de menu, choisir *Visualisation/Color Mapping* et charger la sortie du NDVI calculé avec le module **BandMath**. Sélectionner une plage de valeurs entre -0.2 et 0.7 et utiliser la table de couleurs *Jet*. Voici le résultat tel qu'affiché par le *Viewer* :



OTB applications : mode graphique et ligne de commande

1. Point 1 La première commande lance la version en ligne de commande de l'application **Orthorectification**. La deuxième commande en lance la version avec IHM graphique.
2. Point 2 Il y a 59 applications dans la version 3.14.1 de l'OTB.
3. Point 3 Il existe plusieurs façons d'obtenir de l'aide et de la documentation :
 - Le lancement d'une application en ligne de commande sans paramètres affiche une courte description de l'application et de ses paramètres. Un lien vers la documentation complète sur le site de l'OTB est aussi affiché.
 - La version graphique des applications dispose d'un onglet *Documentation* où une description détaillée est donnée.
 - Enfin, une documentation très complète sur les applications est disponible dans l'Orfeo ToolBox Cookbook.

OTB applications : traitements simples

1. Point 1 Voici l'ensemble des commandes à utiliser :

D'abord, nous calculons le NDVI avec l'application **BandMath** :

```
$ otbcli_BandMath -il phr_xs_melbourne.tif
  -out ndvi.tif float -exp "(im1b4-im1b1)/(im1b4+im1b1) "
```

Ensuite, nous calculons le masque pour PIR < 150 :

```
$ otbcli_BandMath -il phr_xs_melbourne.tif
  -out lownir.tif uint8 -exp "if(im1b4<150,255,0) "
```

Puis, nous calculons le masque des pixels dont toutes les bandes sont supérieures à 1000 :

```
$ otbcli_BandMath -il phr_xs_melbourne.tif
-out high.tif uint8
-exp "if(min(im1b1,im1b2,im1b3,im1b4)>1000,255,0) "
```

Notez que pour générer des masques, un type de pixel *uint8* (entier codé en 8 bits) est suffisant, tandis que pour le calcul du NDVI, un codage en flottant est nécessaire.

Nous pouvons maintenant concaténer tous ces résultats dans une seule image :

```
$ otbcli_ConcatenateImages -il high.tif ndvi.tif lownir.tif
-out map1.tif float
```

Enfin, nous appliquons la palette de couleurs au NDVI comme ceci :

```
$ otbcli_ColorMapping -in ndvi.tif -out map2.png uint8
-method continuous -method.continuous.min -0.2
-method.continuous.max 0.7 -method.continuous.lut jet
```

Pour aller plus loin

1. Point 1 A partir de la ligne de commande, lancer

```
$ monteverdi -in phr_xs_melbourne.tif
```

ouvrira l'image dans **Monteverdi** et l'affichera dans le module de visualisation, et

```
$ monteverdi -il phr_xs_melbourne.tif ndvi.tif
```

permet d'ouvrir plusieurs images dans **Monteverdi**.

2. Point 2 L'application **RadiometricVegetationIndices** permet de calculer plusieurs indices de végétation dont le NDVI.
3. Point 3 Vous pouvez consulter ce chapitre du **Cookbook** afin de vous documenter sur l'interface *Python*.

2.2 Pré-traitements : corrections géométriques et radiométriques

2.2.1 Description

Résumé Cet exercice présente les corrections radiométriques et géométriques des images à l'aide d'**OTB applications**.

Pré-requis

- Connaissances de base en traitement des images de télédétection.
- Notions d'utilisation d'applications en ligne de commande.

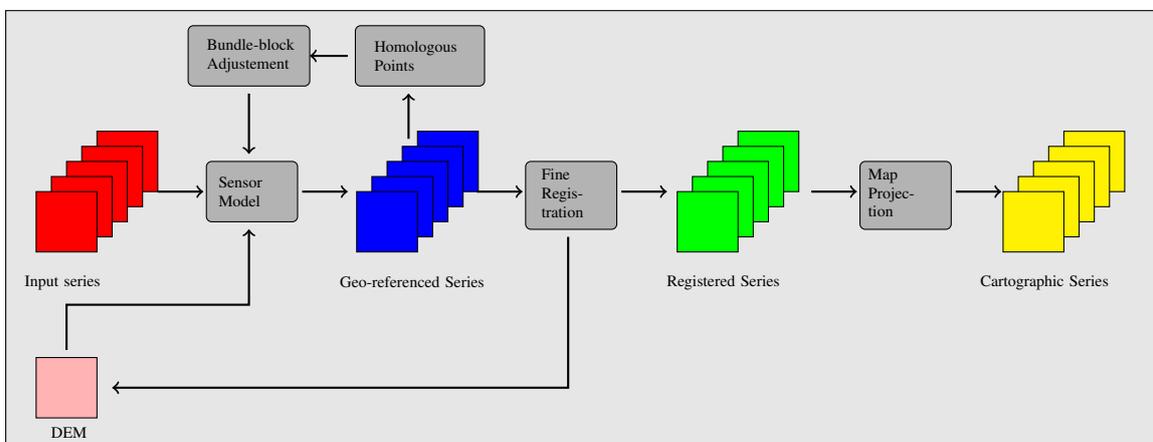
Objectifs

- Ortho-rectifications d'images de télédétection avec **OTB Applications**.
- Corrections radiométriques d'images optiques avec **OTB Applications**.

2.2.2 Étapes

Dans cette partie de l'exercice, vous utiliserez l'image `phr_xs_melbourne.tif`

Principe des corrections géométriques Cette opération permet d'associer des coordonnées au sol aux pixels de l'image.



1. Lancez les versions en ligne de commande et graphique de l'application **Orthorectification**.
2. Quels paramètres permettent de gérer les modèles numériques de terrain (DEM)? Quels modes sont-ils disponibles?
3. Ouvrez l'image avec **Monteverdi**. Quelle est la résolution de l'image?

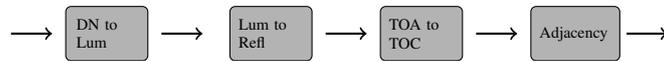
Orthorectification

1. Dans quelle zone UTM se trouve Melbourne?
2. Utilisez Quantum GIS afin de déterminer le code EPSG pour cette projection.
3. Réalisez l'orthorectification de l'image et ouvrez-la dans Quantum GIS.

Trucs et astuces :

- Utilisez la résolution estimée dans **Monteverdi** comme paramètre de résolution de sortie dans l'**Orthorectification**.
- Utilisez une élévation moyenne de 20 m.

Corrections radiométriques d'images optiques Cette opération consiste à transformer les comptes numériques en réflectances (valeurs physiques). Elle comprend le passage de réflectance au sommet de l'atmosphère vers la réflectance de surface, dont l'objectif est de corriger les effets atmosphériques.



1. Utilisez l'application **OpticalCalibration** pour calculer la réflectance au sommet de l'atmosphère.
2. Utilisez l'application **OpticalCalibration** pour calculer la réflectance de surface (top of canopy).
3. Comparez les 2 images en utilisant Monteverdi ou les applications en ligne de commande.

Trucs et astuces :

- Activez l'option '-milli' afin de générer les images de sortie en millièmes de réflectance (entier 16 bits). Par défaut, les images de réflectance sont sauvegardées en flottant (entre 0 et 1).

2.2.3 Solutions

Principe des corrections géométriques

1. Point 2 Le groupe de paramètres *elev* sert à configurer la gestion des modèles numériques de terrain.
2. Point 3

La résolution estimée dans **Monteverdi** est de (1.95;2.03).

Orthorectification

1. Point 1 La zone UTM de Melbourne est la 55. Voir par exemple ici.
2. Point 2 Le code EPSG est le 32755. Cette information est disponible dans la fenêtre de propriétés de projet dans Quantum GIS.
3. Point 3 Voici la commande pour exécuter l'orthorectification :

```

$ otbcli_Orthorectification -io.in phr_xs_melbourne.tif
  -io.out solution/phr_xs_melbourne_ortho.tif uint16
  -map utm -map.utm.zone 55
  -outputs.spacingx 2. -outputs.spacingy -2.
  -interpolator bco
  -elev average -elev.average.value 20
  
```

Corrections radiométriques d'images optiques

1. Point 1 Voici la commande pour calculer la réflectance au sommet de l'atmosphère :

```
$ otbcli_OpticalCalibration -io.in phr_xs_melbourne.tif  
-io.out solution/phr_xs_melbourne_toa.tif uint16  
-milli 1
```

2. Point 2 Voici la commande pour calculer la réflectance de surface (TOC) :

```
$ otbcli_OpticalCalibration -io.in phr_xs_melbourne.tif  
-io.out solution/phr_xs_melbourne_toc.tif uint16  
-milli 1 -level toc
```

3. Point 3 On peut utiliser le module **BandMath** pour calculer la différence entre les 2 images.

2.3 Segmentation

2.3.1 Description

Résumé Cet exercice présente la segmentation d'images à l'aide de l'application **Segmentation**. Vous apprendrez à produire une segmentation raster avec différents algorithmes. L'utilisation du format de sortie vecteur est aussi présenté, car il permet le traitement d'images de taille très grande.

Pré-requis

- Connaissances de base sur les **OTB Applications** et Quantum GIS.
- Notions en segmentation d'images.
- Notions de base en SIG et formats de données vecteur.

Objectifs

- Utilisation de l'application **Segmentation**.
- Segmentation d'images de grande taille et importation des résultats dans un SIG.

2.3.2 \ 'Etapes

Introduction à l'application Segmentation

1. Lancez les versions en ligne de commande et graphique de l'application
2. Lisez la documentation. Quelles sont les 3 méthodes de segmentation disponibles?
3. Quels sont les 2 modes (type de format) des données en sortie?



Segmentation simple en mode raster Dans cette partie de l'exercice vous utiliserez l'image `segmentation_small_xt_phr.tif`

1. Lancez la segmentation en mode *raster* avec la méthode des composantes connexes et une condition de seuil sur la distance spectrale.
2. Visualisez le résultat de la segmentation dans **Monteverdi**. Que voyez-vous?
3. Utilisez l'application **ColorMapping** afin d'améliorer la visualisation :
 - (a) Essayez la méthode *optimal*.
 - (b) Essayez la méthode *image*.
4. Essayez différentes conditions pour les composantes connexes et analysez leur influence sur les résultats. Vous pouvez essayer de changer le seuil de distance ou regarder la documentation pour connaître d'autres mots-clés.

Trucs et astuces :

- Utilisez le mot-clé **distance** dans l'expression pour indiquer la distance spectrale.
- Faites attention au type d'image en sortie.

Autres algorithmes de segmentation Dans cette partie de l'exercice, vous utiliserez l'image `segmentation_small_xt_phr.tif`

1. Lancez la segmentation en mode *raster* à nouveau, mais utilisez la méthode *Mean-Shift*. Utilisez **ColorMapping** pour visualiser les résultats.
 - (a) Commencez par utiliser les paramètres par défaut.
 - (b) Modifiez ensuite les paramètres et analysez les résultats. Les paramètres les plus déterminants sont les rayons spatial et spectral (*range*).
2. Lancez la segmentation en mode *raster* avec la méthode de la ligne de partage des eaux (*Watershed*). Servez-vous de **ColorMapping** pour la visualisation.
 - (a) Commencez par utiliser les paramètres par défaut.
 - (b) Modifiez ensuite les paramètres et analysez les résultats.
3. Comparez les résultats des 3 algorithmes. Conservez la meilleure segmentation pour l'exercice suivant.

Trucs et astuces :

- Il existe 2 mises en oeuvre de la méthode *Mean-Shift*. Edison est la version originale des auteurs de l'algorithme.

Passage à l'échelle : mode vecteur Dans cette partie de l'exercice, vous utiliserez l'image `segmentation_large_xt_phr.tif`

1. Lancez la segmentation en mode *raster* en utilisant les meilleurs paramètres de l'exercice précédent, mais sur la grande image. Regardez la mémoire et l'utilisation du processeur. Qu'est-ce qui se passe?
2. Lancez à nouveau la segmentation mais en mode *vecteur* en désactivant l'option **stitching**. Comparez les ressources utilisées avec le point précédent.
3. Ouvrez l'image et le résultat de la segmentation dans Quantum GIS. Ajustez les paramètres de visualisation (voir [Trucs et astuces](#) plus bas). Qu'observez-vous?
4. Lancez à nouveau la segmentation en mode *vecteur* en activant l'option **stitching**. Sauvegardez les résultats dans un nouveau fichier et ouvrez-le dans le même projet Quantum GIS. Quel est l'effet de l'option **stitching**?

Trucs et astuces :

- Sous Linux ou Mac OS, les ressources peuvent être consultées avec la commande `top` dans un terminal. Sous Windows, on peut utiliser le gestionnaire de tâches.
- Appuyez sur `Ctrl C` pour arrêter l'application.
- Utilisez le format *sqlite* pour sauvegarder les fichiers vecteur (extension `.sqlite`).
- Quantum GIS peut importer des couches raster et vecteur.
- La visualisation de couches raster sous Quantum GIS peut être ajustée comme ceci :
 - click droit sur la couche, puis *Properties*
 - allez dans l'onglet *Style*
 - choisir *Use standard deviation*
 - dans *Contrast enhancement*, choisir *Stretch to MinMax*
- La visualisation de couches vecteur sous Quantum GIS peut être ajustée comme ceci :
 - click droit sur la couche, puis *Properties*
 - allez dans l'onglet *Style*, choisissez *Change*
 - dans *Symbol layer type*, choisissez *Outline: Simple line*
 - on peut aussi changer la couleur
- Dans Quantum GIS on peut sauvegarder le projet dans un fichier de façon à conserver tous ces paramètres.

Pour aller plus loin

1. En mode *vecteur*, étudiez les effets des paramètres *tilesize*, *simplify* et *minsize*.
2. Comment segmenter toutes les régions de l'image sauf la végétation à l'aide de l'application **Segmentation** et éventuellement d'autres applications?
3. Avec les applications disponibles, comment gérer la segmentation de structures fortement réfléchissantes?



2.3.3 Solutions

Introduction à l'application Segmentation

1. Point 1 Pour obtenir l'aide en ligne de commande, lancez

```
$ otbcli_Segmentation
```

Pour la version graphique, lancez

```
$ otbgui_Segmentation
```

2. Point 2 Il y a 3 méthodes de segmentation dans l'application :

- Mean-Shift (deux version différentes)
- Ligne de partage des eaux ou Watershed
- Composantes connexes

3. Point 3 Il y a 2 modes pour la génération de la segmentation :

- Le mode *raster* qui est utile pour des images de petite taille et qui produit une image où chaque région de la segmentation est étiquetée avec un identifiant unique.
- Le mode *vecteur* qui permet de segmenter des images de grande taille et qui produit un fichier vecteur où chaque région de la segmentation est représentée par un polygone.

Segmentation simple en mode raster

1. Point 1 Voici la ligne de commande avec un seuil de 30 sur la distance spectrale :

```
$ otbcli_Segmentation -in segmentation_small_xt_phr.tif  
-filter cc -filter.cc.expr "distance < 30"  
-mode raster -mode.raster.out first_cc.tif uint32
```

Notez que nous utilisons `uint32` pour le codage des valeurs de sortie afin d'avoir assez d'étiquettes uniques pour toutes les régions.

2. Point 2 Le résultat de la segmentation est difficile à visualiser car les segments adjacents sont souvent étiquetés de façon consécutive. On peut noter le gradient d'intensité entre le haut et le bas de l'image, ce qui correspond aux valeurs croissantes des étiquettes.
3. Point 3 La ligne de commande suivante utilise **ColorMapping** en mode *optimal* :

```
$ otbcli_ColorMapping -in first_cc.tif  
-out first_cc_color_optimal.png uint8  
-method optimal
```

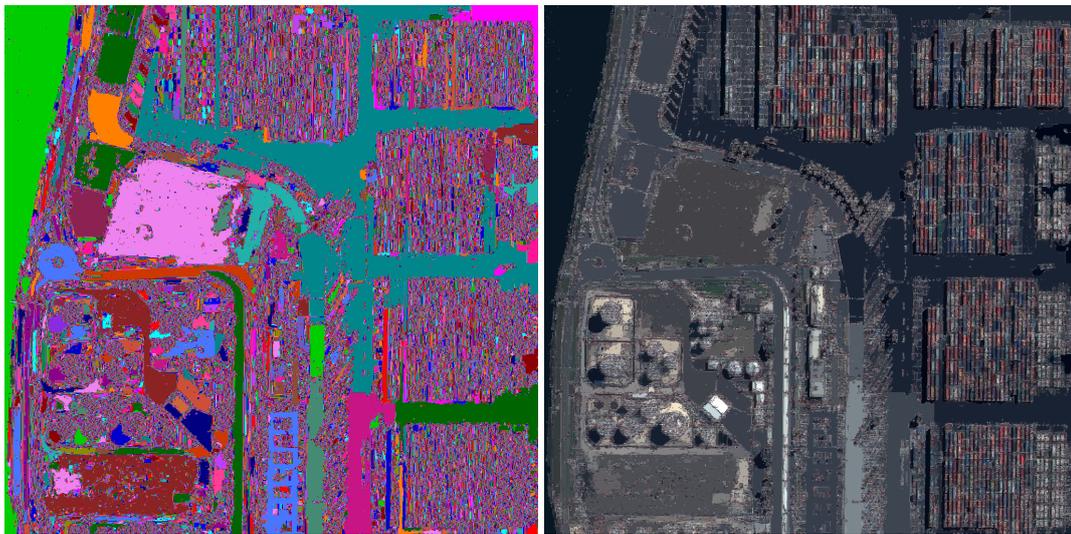
La table de couleurs `/optimal/e` associe une couleur à chaque région de façon à maximiser le contraste avec les régions voisines. Nous utilisons `uint8` pour le codage de sortie, car **ColorMapping** produit des images en 8 bits qui peuvent être visualisées avec n'importe quel outil de visualisation.

Avec cette table de couleurs, on observe que l'image est sur-segmentée.

```
$ otbcli_ColorMapping -in first_cc.tif
-out first_cc_color_image.png uint8
-method image -method.image.in segmentation_small_xt_phr.tif
```

La table de couleurs *image* associe à chaque segment la couleur moyenne dans l'image originale. Ceci permet d'avoir une visualisation plus réaliste. Étant donné que l'image est sur-segmentée, l'application affichera beaucoup de messages dans le terminal.

Voici les résultats obtenus avec les 2 méthodes, *optimal* à gauche et *image* à droite :

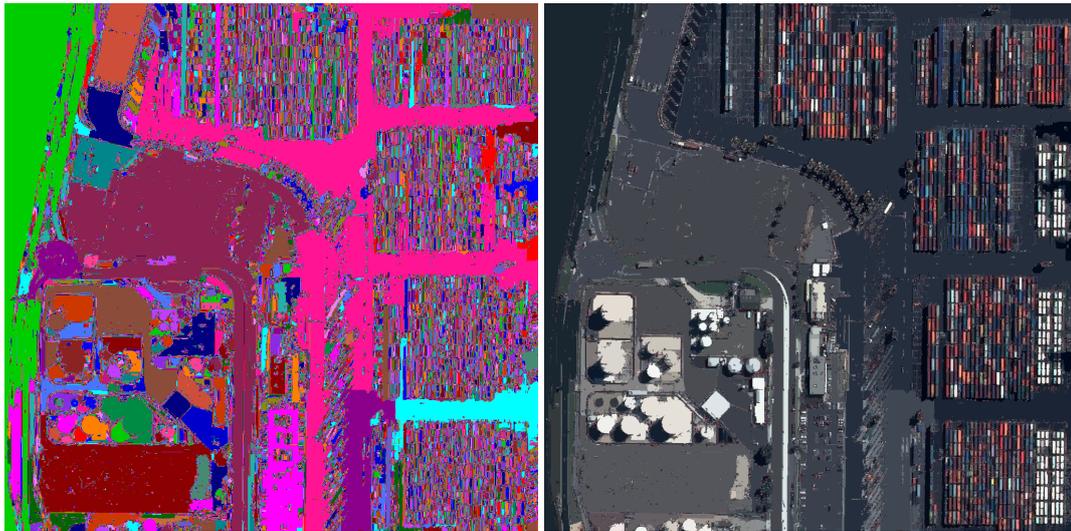


4. Point 4 Voici un autre exemple. La commande suivante regroupera les pixels qui

- ont une distance spectrale inférieure à 30
- ou qui ont simultanément une intensité supérieure à 400 et une distance spectrale inférieure à 50
- ou qui ont tous les deux une intensité supérieure à 1000
- ou qui ont tous les deux une valeur dans le PIR inférieure à 150.

```
$ otbcli_Segmentation -in segmentation_small_xt_phr.tif
-filter cc -filter.cc.expr "distance<30
or (intensity_p1>400 and intensity_p2 > 400 and distance<50)
or(intensity_p1 >1000 and intensity_p2>1000
or (p1b4 <150 and p2b4<150))"
-mode raster -mode.raster.out second_cc.tif uint32
```

Voici le résultat obtenu :



Autres algorithmes de segmentation

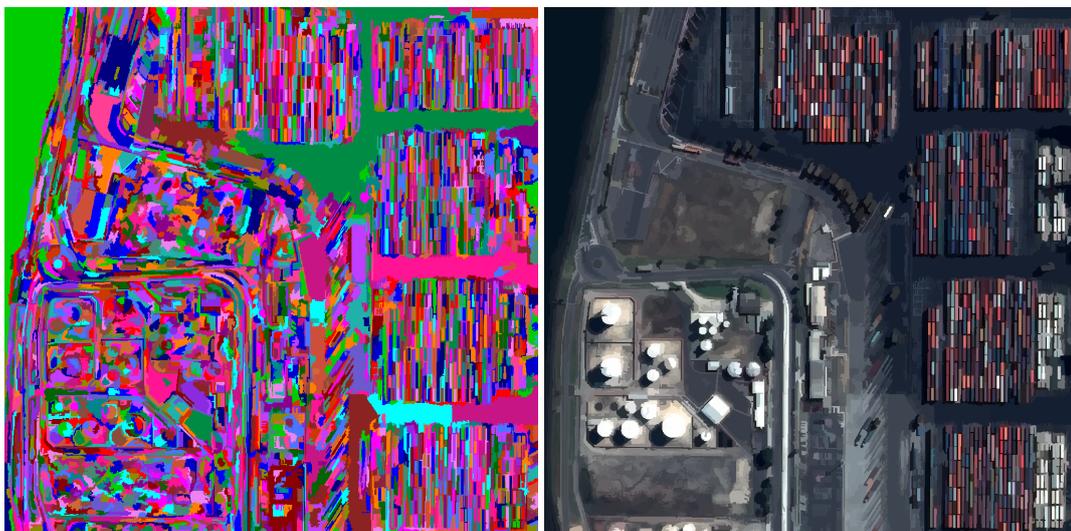
1. Point 1 Voici la ligne de commande pour lancer la segmentation *Mean-Shift* avec les paramètres par défaut :

```
$ otbcli_Segmentation -in segmentation_small_xt_phr.tif  
-filter meanshift -mode raster  
-mode.raster.out meanshift.tif uint32
```

Un meilleur résultat est obtenu en utilisant un rayon spectral de 30 :

```
$ otbcli_Segmentation -in segmentation_small_xt_phr.tif  
-filter meanshift -filter.meanshift.ranger 30 -mode raster  
-mode.raster.out meanshift.tif uint32
```

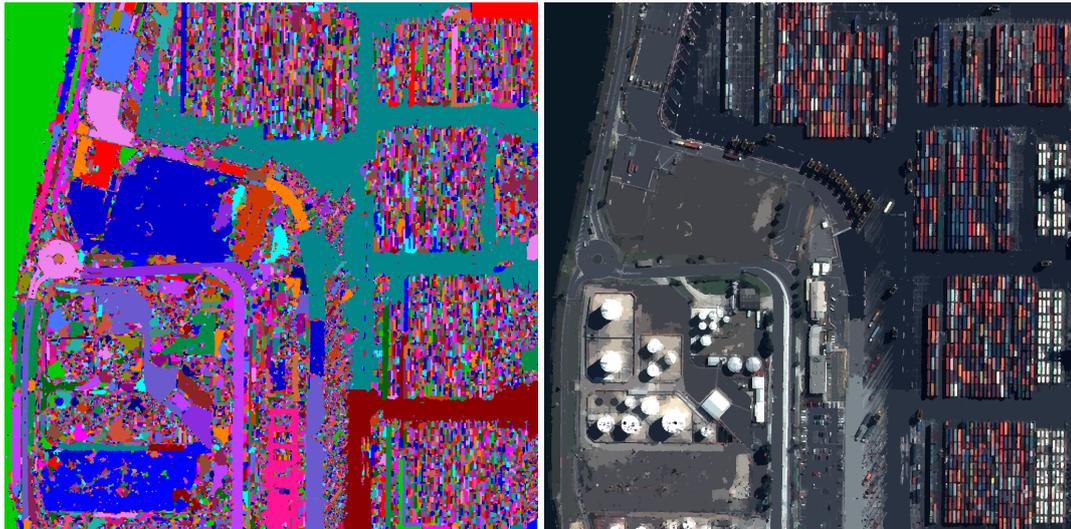
Voici le résultat :



Voici la ligne de commande pour l'algorithme du Watershed avec les paramètres par défaut :

```
$ otbcli_Segmentation -in segmentation_small_xt_phr.tif
  -filter watershed -mode raster
  -mode.raster.out watershed.tif uint32
```

Et enfin les résultats :



Passage à l'échelle : mode vecteur

1. Point 1 La commande suivante lance la segmentation sur la grande image :

```
$ otbcli_Segmentation -in segmentation_large_xt_phr.tif
  -filter meanshift -filter.meanshift.ranger 30 -mode raster
  -mode.raster.out meanshift.tif uint32
```

Étant donné que l'image est de grande taille, (8192 par 8192 pixels), il est probable que, en fonction de la machine :

- l'application termine en erreur par manque de mémoire
- l'application ne s'arrête pas, mais consomme toute la mémoire disponible.

2. Point 2 La commande suivante lance la segmentation en mode *vecteur* sans l'option *stitch* :

```
$ otbcli_Segmentation -in segmentation_large_xt_phr.tif
  -filter meanshift -filter.meanshift.ranger 30 -mode vector
  -mode.vector.out meanshift.sqlite -mode.vector.stitch 0
```

En mode vecteur, la consommation de mémoire est stable, car la segmentation se fait en appliquant un tuilage sur l'image.

3. Point 3 Dans le SIG on observe l'effet du tuilage : les bords des tuiles sont visibles sur le résultat de la segmentation. On peut aussi voir que la segmentation produit un très grand nombre de polygones.
4. Point 4 La commande suivante lance l'application en mode *vecteur* avec l'option *stitch* :

```
$ otbcli_Segmentation -in segmentation_large_xt_phr.tif  
-filter meanshift -filter.meanshift.ranger 30 -mode vector  
-mode.vector.out meanshift.sqlite -mode.vector.stitch 1
```

La visualisation des résultats dans Quantum GIS permet de voir que la plupart des effets du tuilage ont disparu. Les résultats sont donc proches de ce que nous obtiendrions sans tuilage.

Voici les résultats tels que visualisés dans le SIG. En bleu, les résultats sans stitching et en rouge, avec.



Pour aller plus loin

1. Point 1
 - L'option *tilesize* permet de régler la taille des tuiles utilisées pour la segmentation.
 - L'option *simplify* permet de simplifier les polygones générés en utilisant une valeur de tolérance exprimée en pixels. Le fichier résultat sera de taille inférieure.
 - L'option *minsize* permet d'éliminer les segments de taille inférieure à un seuil (en pixels).

2. Point 2 Afin d'éviter de segmenter la végétation, on peut construire un masque de végétation avec **BandMath** par seuillage du NDVI. Ce masque peut être utilisé par la segmentation en utilisant l'option *mode.vector.inmask*. Notez que cette option est seulement disponible en mode *vecteur*.
3. Point 3 Les objets de forte réflectance sont souvent difficiles à segmenter. A cause des réflexions spéculaires, leur variance est très forte. En conséquence, les méthodes basées sur la comparaison de pixels voisins ne marchent pas correctement. Les 3 méthodes présentées ici utilisent cette approche.

Une façon de contourner le problème est d'associer ensemble tous les pixels voisins de forte réflectance, ce qui peut s'obtenir avec la segmentation par composantes connexes comme cela a été montré plus haut.

2.4 Extraction de primitives

2.4.1 Description

Résumé Cet atelier présente les modules d'extraction de primitives de **Monteverdi**. Vous apprendrez à utiliser ces modules et à évaluer visuellement l'utilité de différents types de primitives pour l'extraction d'information des images.

Pré-requis

- Utilisation de **Monteverdi**

Objectifs Être capable d'utiliser les modules d'extraction de primitives de **Monteverdi**.

2.4.2 Étapes

Familiarisation avec les modules d'extraction de primitives L'extraction de primitives est un terme générique qui fait référence aux techniques de traitement des images qui permettent de produire de nouvelles images ou des ensembles d'objets afin de représenter l'information contenue par les images d'origine avec un niveau d'abstraction plus élevé.

D'autres termes que *primitives* peuvent être utilisés, comme par exemple *indices* ou *descripteurs*. Même s'il existe des différences légères entre ces termes, nous les utiliserons comme des synonymes.

Dans **Monteverdi** il y a 5 modules pour l'extraction de primitives et ils sont regroupés sous le menu *Filtering/\$→\$/Feature extraction*.

Tous les modules d'extraction de primitives utilisent la même approche : ils prennent une image en entrée (avec un nombre quelconque de bandes) et produisent une image multi-canal où chaque composante est une des primitives calculées.

Dans cette partie de l'exercice, vous utiliserez l'image `phr_xs_melbourne.tif`

1. Ouvrez l'image dans **Monteverdi**.
2. Ouvrez le module **Smoothing** à partir du menu *Filtering/\$→\$/Feature extraction* et choisissez l'image d'entrée.
3. A quoi correspondent les 3 images qui s'affichent dans l'interface?

4. L'onglet *Action* est sélectionné par défaut. Que peut-on faire dans cette partie du module?
5. Allez dans l'onglet *Output* et décrivez ce que l'on peut y faire.

Le module Smoothing Vous pouvez garder le module ouvert dans l'exercice précédant ou le fermer et en ouvrir un de nouveau.

1. Une des primitives disponibles est *Original data*. De quoi s'agit-il et quelle en est l'utilité?
2. Choisissez la primitive *Mean* et calculez-la seulement sur l'intensité de l'image. A quoi correspondent les paramètres *Radius along X* et *Radius along Y* et quel est leur effet sur le résultat?
3. Choisissez la primitive *Meanshift smooth*. Essayez d'interpréter les paramètres et leur effet sur les résultats.
4. Sélectionnez la primitive *Meanshift clusters*. Les mêmes paramètres que pour *Meanshift smooth* sont disponibles. Quelle est la différence entre ces 2 primitives?

Le module Edge extraction Ouvrez le module *Edge extraction* et donnez-lui une image d'entrée. Pour rester simple, on ne travaillera qu'avec une seule bande. Aussi les primitives *Harris detector* – points d'intérêt – et *Touzi* – pour des images SAR – seront ignorées.

Répondez aux questions suivantes en utilisant le texte affiché dans la fenêtre *Feature Information*, la liste de paramètres pour chaque primitive et leurs effets sur les résultats.

1. Quelle est l'information produite par la primitive *Meanshift boundaries*?
2. Comment peut-on détecter des contours avec un filtre de *Variance*?
3. Quel est l'effet du paramètre *sigma* du filtre *Recursive gradient*?
4. Dans la primitive *Sobel*, le résultat est une densité de contours calculée après l'application d'un filtre de rehaussement de contours de Sobel. On peut utiliser 2 seuils différents. Quel est leur effet sur le résultat?

Le module Radiometric index extraction Les indices radiométriques sont des combinaisons de bandes spectrales qui permettent de mettre en évidence la présence de certains matériaux ou types de couverture de la surface. Il existe des indices de végétation, de sols, d'eau, de surfaces bâties, etc. Vous pouvez trouver des descriptions détaillées et des références bibliographiques à partir de la page sur les indices radiométrique du wiki de l'Orfeo Toolbox.

1. Ouvrez le module *Radiometric index extraction* et choisissez comme entrée l'image à 4 bandes. Choisissez *NDVI* dans la liste *Vegetation*. Générez des résultats en utilisant différents choix pour les canaux (*Channels Selection*). Pourquoi le résultat est-il toujours le même indépendamment de ce choix?
2. Quelles sont les bandes spectrales les plus souvent utilisées pour les indices de végétation?
3. Trouvez une valeur pour le paramètre *s* de *MSAVI* qui donne des résultats similaires à ceux du *RVI*. Quel est l'intérêt d'indices avec des paramètres par rapport à des indices qui n'en ont pas, comme le *RVI* ou le *NDVI*?
4. Quels indices d'eau ne peuvent pas être calculés pour une image Pléiades?

Le module Texture extraction

1. SFS L'approche *Structural Feature Set* calcule des textures en appliquant une analyse directionnelle à partir du pixel central de la fenêtre d'analyse. Les directions sont calculées en utilisant un pas constant.

Une direction est définie comme :

$$d_i = \sqrt{(m^{e1} - m^{e2})^2 + (n^{e1} - n^{e2})^2}$$

A partir de d_i , on définit des histogrammes :

$$H(c) : \{c \in I \mid [d_1(c), \dots, d_i(c), \dots, d_D(c)]\}$$

Enfin, 6 textures sont calculées :

$$length = \max_{i \in [1;D]} (d_i(c))$$

$$width = \min_{i \in [1;D]} (d_i(c))$$

$$PSI = \frac{1}{D} \sum_{l=1}^D d_l(c)$$

$$\omega - mean = \frac{1}{D} \sum_{l=1}^D \frac{\alpha \cdot (k_l - 1)}{st_l} d_l(c)$$

$$ratio = \arctan \frac{\sum_{j=1}^n sort_{min}^j(H(c))}{\sum_{j=1}^n sort_{max}^j(H(c))}$$

$$SD = \frac{1}{D-1} \sqrt{\sum_{l=1}^D (d_l(c) - PSI)^2}$$

- (a) En utilisant les paramètres par défaut, calculez les primitives *width* et *length*. Expliquez les différences observées.
 - (b) Quel est l'effet du seuil spectral sur le résultat? Même question pour le seuil spatial.
2. Haralick Les textures de Haralick sont un ensemble d'indices calculés à partir des matrices de co-occurrence en niveaux de gris. Ces matrices sont calculées pour chaque pixel de l'image en niveaux de gris. Un voisinage est défini par une fenêtre rectangulaire et son homologue décalé d'un certain offset est aussi utilisé.

La matrice de co-occurrence C définie sur un voisinage de $n \times m$ pixels d'une image I et avec un offset $(\Delta x, \Delta y)$ est définie comme ceci :

$$C_{\Delta x, \Delta y}(i, j) = \sum_{p=1}^n \sum_{q=1}^m \begin{cases} 1, & \text{if } I(p, q) = i \text{ and } I(p + \Delta x, q + \Delta y) = j \\ 0, & \text{otherwise} \end{cases}$$

Les valeurs des niveaux de gris sont quantifiées avec une certaine précision (nombre de bins) de façon à ce que l'égalité entre les valeurs soit probable.

Dans **Monteverdi**, il existe 2 versions des textures de Haralick. Nous utiliserons seulement la 1ère.



- (a) Quel est le sens des paramètres *radius*, *offset*, *min/max* et *quant. levels*?
- (b) Calculez l'entropie de l'intensité de l'image pour des rayons différents (2, 3, etc.). Quel est l'effet de ce paramètre?
- (c) Calculez les indices *Energy*, *Entropy*, *Correlation* et *Inertia* avec les mêmes paramètres. Comparez-les et dites si certains vous paraissent redondants.

2.4.3 Solutions

Familiarisation avec les modules d'extraction de primitives

1. Point 3 L'interface graphique affiche une version sous-échantillonnée de l'image à gauche, la pleine résolution au centre et la primitive calculée à droite.
2. Point 4 L'onglet *Action* permet de sélectionner la primitive à calculer, décider sur quels canaux elle sera calculée, choisir les paramètres de la primitive et voir la liste des primitives calculées. Le bouton *Add* ajoute une primitive à la liste de droite.
3. Point 5 L'onglet *Output* permet de choisir, parmi les primitives calculées, quelles seront gardées en sortie et dans quel ordre elles apparaîtront sur l'image.

Le moule Smoothing

1. Point 1 *Original data* copie les canaux de l'image d'entrée choisis comme des canaux des l'image de sortie. Ceci peut être utile dans le cas où l'on souhaite créer une image pour réaliser des classifications et l'on souhaite avoir des bandes d'origine en plus des primitives calculées.
2. Point 2 Les *radius* définissent la taille de la fenêtre glissante qui sera utilisée pour de calcul de la moyenne autour de chaque pixel. Il s'agit d'une fenêtre rectangulaire centrée sur le pixel à traiter dont la taille est de $2 \times \text{Radius}_x + 1$ pixels dans la direction horizontale et $2 \times \text{Radius}_y + 1$ pixels dans la direction verticale. Plus la taille de la fenêtre est grande, plus l'effet de lissage est important.
3. Point 3 La primitive *Meanshift smooth* utilise l'algorithme du mean-shift afin de lisser l'image. Il y a 2 intérêts principaux à utiliser cet algorithme par rapport à la moyenne glissante :
 - (a) les contours sont préservés;
 - (b) on peut l'appliquer à des images multi-canal et utiliser la corrélation entre canaux.

Cet algorithme effectue le lissage de façon simultanée dans l'espace image (lignes, colonnes) et dans l'espace spectral (celui défini par exemple par les bandes spectrales).

Voici la description des paramètres disponibles :

- (a) Spatial radius : le rayon de la fenêtre spatiale pour le lissage.
- (b) Range radius : le rayon de la fenêtre de lissage dans l'espace spectral.
- (c) Min. region size : la taille minimale pour qu'une région soit conservée dans le clustering (n'est pas utilisé dans le lissage).
- (d) Scale : facteur multiplicatif pour modifier les valeurs des pixels, ce qui est nécessaire quand la dynamique de l'image est faible.

4. Point 4 La différence entre le lissage et le clustering réside sur le fait que ce dernier produit une image qui est constante par morceaux, c'est à dire que les pixels qui appartiennent à une même région ont la même valeur.

Ces régions sont définies à la fin du lissage en attribuant à chaque pixel la valeur du mode de de l'histogramme auquel il appartient.

Quand des clusters (les régions associées au même mode) ont une taille inférieure au minimum spécifié dans les paramètres, ils sont fusionnés avec la région adjacente la plus similaire.

Module Edge extraction

- Point 1 Il s'agit des contours des régions générées par la primitive *Meanshift clusters* du module *Smoothing*.
- Point 2 Ce filtre associe à chaque pixel la valeur de la variance locale à l'intérieur d'une fenêtre centrée sur celui-ci :

$$var(i, j) = \frac{1}{(2Radius_x + 1) \times (2Radius_y + 1)} \sum_{i-Radius_x}^{i+Radius_x} \sum_{j-Radius_y}^{j+Radius_y} (pix(i, j) - \mu(i, j))^2$$

où $pix(i, j)$ est la valeur du pixel d'entrée et $\mu(i, j)$ est la moyenne locale calculée en utilisant la même fenêtre.

Les valeurs de variance seront élevées quand les pixels à l'intérieur de la fenêtre s'éloignent de la moyenne locale. Ceci peut arriver dans 2 cas différents :

- Quand il y a une texture très prononcée.
 - Quand il y a plusieurs régions dans la fenêtre avec des valeurs moyennes différentes. Ceci correspond au cas où un contour est présent.
- Point 3 Le gradient récursif utilise un lissage gaussien (filtrage passe-bas) préalable au calcul du gradient utilisé pour détecter les contours. Le paramètre *sigma* règle la largeur de la gaussienne utilisée pour le lissage, c'est à dire le degré de flou.
L'effet de ce paramètre est le suivant : plus sa valeur est élevée, plus les contours détectés seront larges et moins il y aura de fausses détections dues au bruit.
La valeur de *sigma* choisie dépendra donc du niveau de bruit de l'image et du type de contours que l'on souhaite détecter.
 - Point 4 Les seuils haut et bas définissent les intervalles de valeurs qui seront mis à 1 (à l'extérieur des seuils) ou à 0 (entre les 2 seuils) après le filtrage de Sobel et avant le calcul de densité de contours. Ces seuils déterminent donc comment le résultat du filtrage de Sobel est binarisé.

Le module Radiometric index extraction

- Point 1 Pour les indices radiométriques, la sélection des canaux à gauche n'a pas d'influence, car chaque indice demande à ce que les bandes spectrales utilisées soient sélectionnées dans la partie centrale de l'interface (*Feature Parameters*).



2. Point 2 La plupart des indices de végétation utilisent la bande rouge (R) et la bande proche infra-rouge (NIR), car la végétation a une faible réponse dans le rouge et très forte dans le PIR. Beaucoup d'indices utilisent donc des différences ou des ratios entre ces 2 bandes.

Parfois, la bande verte est aussi utilisée.

3. Point 3 Des valeurs supérieures à 6 devraient bien marcher.

L'intérêt d'avoir des paramètres est que l'on peut prendre en compte la réflectance du sol pour le cas de végétations peu denses. La paramètre L du SAVI est proche de 0 pour des végétations très éparées et proche de 1 pour des végétations très denses. Le paramètre s du MSAVI est la pente de la droite des dols, c'est à dire, la réflectance PIR en fonctions de la réflectance dans le rouge pour des pixels de sol nu.

4. Point 4 Le NDTI et le NDWI ne peuvent pas être calculés avec des images Pléiades (ou avec des images Quickbird ou Ikonos d'ailleurs) car le moyen infra-rouge (MIR ou SWIR) n'est pas disponible. Dans ce cas, le NDWI2 peut être utilisé.

Le module Texture extraction

1. SFS

- (a) Point 1 Il peut paraître paradoxal, mais *width* donne des valeurs élevées aux pixels qui appartiennent à des régions longues, tandis que *length* attribue des valeurs élevées à toute région (longue ou pas) qui a une grande superficie. Les formules de chaque primitive permettent de comprendre pourquoi.
- (b) Point 2 Le seuil spectral règle la valeur de la différence entre 2 pixels adjacents le long d'une ligne qui peut être acceptée pour continuer à calculer la longueur dans cette direction. Un faible valeur de ce seuil produira donc des lignes plus courtes et donc moins de pixels avec des valeurs élevées.

Le seuil spatial arrête l'analyse dans une direction indépendamment de la valeur des pixels une fois qu'une certaine longueur est atteinte. Une valeur faible produira aussi des lignes plus courtes.

2. Haralick

(a) Point 1

- Le paramètre *radius* définit la taille de la fenêtre locale utilisée pour le calcul de la matrice de co-occurrence.
- Le paramètre *offset* fixe Δx et Δy .
- Les valeurs *min*/*max* peuvent être utilisées pour définir la plage de valeurs sur laquelle la quantification sera réalisée.
- Le paramètre *quant. levels* détermine le nombre de valeurs qui seront utilisées pour la quantification.

- (b) Point 2 Plus le rayon est grand, plus les zones détectées seront grandes, car une sorte de lissage est introduit quand on utilise des fenêtres plus grandes.

- (c) Point 3 Visuellement, l'énergie et l'entropie semblent très corrélées. Cependant, à y regarder de plus près, on voit que les 4 textures donnent le même type d'information en général. Même si la corrélation et l'énergie paraissent très différentes à cause de l'inversion de contraste, elles mettent en évidence le même type de zones.

En fait, les textures de Haralick sont utiles pour les zones avec des motifs pseudo-périodiques et que l'on choisit les paramètres correctement. Dans les autres cas, il est souvent plus judicieux d'utiliser des statistiques de 1er ordre (comme la variance locale) qui sont plus simples et plus rapides à calculer et qui donnent des informations similaires.

2.5 Apprentissage et classification au niveau pixel

2.5.1 Description

Résumé Cet exercice présente l'approche pixel des outils de classification de l'OTB. On apprendra à faire l'apprentissage d'un SVM avec des images Pléiades en utilisant des régions d'entraînement. On appliquera ensuite le modèle appris afin de générer des cartes d'occupation des sols à partir d'images Pléiades.

Pré-requis

- Utilisation des applications OTB et savoir de débrouiller avec QGIS
- Notions sur la classification supervisée
- Connaissance des formats vecteur utilisés dans les SIG.

Objectifs

- Utilisation des applications de classification de l'OTB
- Classification d'images de grande taille
- Import de résultats dans un SIG

2.5.2 Étapes

Dans cet exercice, vous utiliserez l'image `melbourne_ms_toa_ortho_extract_small.tif`

Génération et analyse d'exemples d'apprentissage

- Utilisez QGIS pour générer des polygones pour 5 classes (végétation, routes, sols nus, bâtiments et eau).
- Exportez cette couche vecteur en format shapefile.
- Quelle est l'étiquette de la classe **eau** dans le shapefile? Un exemple de fichier avec des échantillons d'apprentissage est disponible pour cet exercice dans le fichier `training.shp`

Trucs et astuces :

- Notez le nom du champ du shapefile qui contient l'étiquette. Il sera nécessaire pour lancer l'application qui réalise l'apprentissage.



Estimation des statistiques de l'image Afin de rendre les primitives comparables entre images, il est utile d'en analyser les statistiques. Ces statistiques seront utilisées pour normaliser les données (centrer et réduire).

- Utilisez l'application **ComputeImagesStatistics** afin d'obtenir les statistiques de l'image.
- Quelle est la moyenne de la bande rouge?
- L'extrait fourni a été converti en millièmes de réflectance à partir des comptes numériques. Quel est l'utilité de cette transformation pour la classification d'images?

Estimation du modèle de classification avec les SVM L'application **TrainSVMImagesClassifier** réalise l'apprentissage d'un classifieur SVM à partir de plusieurs paires image/donnée vecteur (exemples). Les échantillons sont composés des valeurs des pixels dans chaque bande, éventuellement centrées-réduites en utilisant les statistiques fournies par un fichier XML qui peut être produit avec **ComputeImagesStatistics**.

Nous réaliserons l'apprentissage avec une seule image ici.

- Utilisez **TrainSVMImagesClassifier** afin de générer le modèle SVM.
- Quel noyau est utilisé par défaut par l'application?
- Quelle est la précision (*accuracy*) mesurée?

Utilisation du modèle de classification

- Utilisez **ImageSVMClassifier** afin d'appliquer le modèle de classification à l'image.
- Quelle est la sortie de l'application?
- Bonus : utilisez le même modèle sur un autre extrait `melbourne_ms_toa_ortho_extract_large.tif`

Génération d'une carte de classification imprimable Afin de pouvoir imprimer ou visualiser plus aisément le résultat de la classification, nous utiliserons **ColorMapping**. Cet outil remplace chaque étiquette avec une couleur RVB codée en 8 bits donnée dans un fichier de correspondances. Ce fichier ressemble à ceci :

```
$ # Les lignes commençant par # sont ignorées
1 255 0 0
```

- Produisez votre table de correspondances (LUT, pour *look-up table*).
- Utilisez cette LUT pour générer la classification *imprimable* (format PNG).
- Superposez ce résultat sur l'image originale dans QGIS. Commentez les résultats de la classification.

Pour aller plus loin

- Générez un modèle de classification avec un type différent de noyau SVM.
- Passage à l'échelle : appliquez ce type de classification à une image Pléiades fusionnée.

2.5.3 Solutions

Génération et analyse d'exemples d'apprentissage L'étiquette de la classe **eau** (water) dans le fichier est **2**.

Estimation des statistiques de l'image Voici la commande pour produire les statistiques de l'image :

```
$ otbcli_ComputeImagesStatistics
  -il melbourne_ms_toa_ortho_extract_small.tif
  -out melbourne_ms_toa_ortho_extract_small_stats.xml
```

La valeur de la moyenne de la bande rouge est **111.625**. L'utilisation des réflectances permet de comparer les valeurs radiométriques obtenues par des capteurs différents ou par le même capteur à des instants différents.

Estimation du modèle de classification avec les SVM Voici la commande pour générer le modèle SVM :

```
$ otbcli_TrainSVMImagesClassifier
  -io.il melbourne_ms_toa_ortho_extract_small.tif
  -io.imstat melbourne_ms_toa_ortho_extract_small_stats.xml
  -io.vd training.shp
  -io.out melbourne_ms_toa_ortho_extract_small_model.svm
```

Le noyau linéaire est utilisé par défaut dans l'application.

La précision mesurée est d'environ 0.988. Cette valeur est très élevée à cause du faible nombre d'échantillons d'apprentissage et de la faible variabilité.

Utilisation du modèle de classification Voici la commande pour générer la classification :

```
$ otbcli_ImageSVMClassifier
  -in melbourne_ms_toa_ortho_extract_small.tif
  -svm melbourne_ms_toa_ortho_extract_small_model.svm
  -imstat melbourne_ms_toa_ortho_extract_small_stats.xml
  -out melbourne_extract_small_classification_5classes.tif
  uint8
```

Les pixels de l'image de sortie contiendront l'étiquette de la classe choisie par le classifieur.

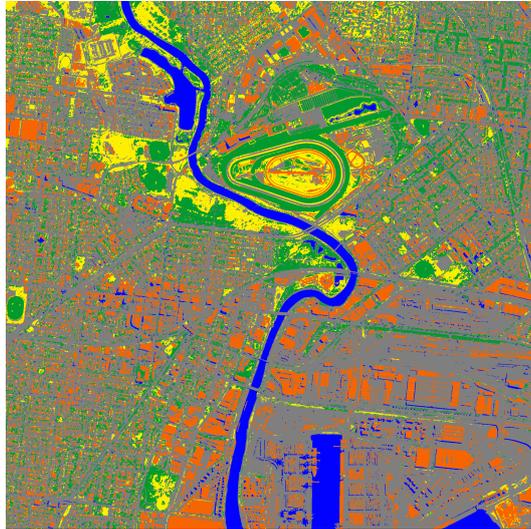
Génération d'une carte de classification imprimable Voici un exemple de LUT :

```
#Vegetation (green) 1 1 154 46 #Water (blue) 2 1 1 254 #Soil (yellow) 3 254 239 1 #Roads (grey)
4 127 127 127 #Buildings (orange) 5 246 101 1
```

Et voici la commande pour l'appliquer à la classification :

```
$ ColorMapping
  -in melbourne_extract_small_classification_5classes.tif
  -out melbourne_extract_small_classification_color.png uint8
  -method custom -method.custom.lut ColorTable.txt
```

Et le résultat de cette commande :



2.6 Apprentissage et classification sur des objets

2.6.1 Description

Résumé Cet atelier présente le module **Object Labeling** de **Monteverdi**. On apprendra à utiliser le module et à observer l'influence des différentes primitives sur la classification. On utilisera aussi une approche simple d'apprentissage actif.

Pré-requis

- Notions d'analyse d'image sur des objets (OBIA, object based image analysis).
- Notions en apprentissage automatique pour la classification.

Objectifs Apprendre à utiliser le module **Object Labeling** de **Monteverdi**.

2.6.2 Étapes

La segmentation préalable Dans cette partie de l'exercice, nous utiliserons les données suivantes :

```
phr_pxs_melbourne_xt_small.tif  
phr_pxs_melbourne_xt_small_segmentation.tif
```

1. Utilisez l'application **ColorMapping** afin de mieux visualiser l'image segmentée (vous pouvez utiliser les modes *optimal* ou *image*).
2. Analysez cette image couleur. Pour quel type d'objets la classification sur les objets risque de bien marcher? Et pour lesquels elle risque de ne pas bien marcher?

Introduction au module Object Labeling

1. Ouvrez l'image originale et la segmentation dans **Monteverdi**.
2. Lancez le module **Object Labeling** à partir du menu *Learning*.
3. A quoi sert chaque onglet à gauche de l'interface?
4. Créez une nouvelle classe dans l'onglet *Objects*. On peut lui changer le nom et la couleur.
5. Cliquez avec le bouton droit sur un objet de l'image. Que se passe-t'il?
6. Cliquez une 2^e fois avec le bouton droit sur le même objet. Décrivez ce que vous observez.
7. Ajoutez quelques objets de plus à la classe courante.
8. Créez une nouvelle classe et ajoutez-y quelques objets.
9. Allez dans l'onglet *Features* et dé-sélectionnez toutes les primitives sauf les moyennes des valeurs radiométriques.
10. Allez dans l'onglet *Learning* et cliquez sur *classify*. Que se passe-t'il?
11. Cliquez sur *Save/Quit*. Quel type de sorties sont produites par le module?

Trucs et astuces :

- Choisissez deux classes simples pour cette partie de l'exercice, par exemple *eau* et *sol*.
- Utilisez la fenêtre de navigation pour changer la zone affichée.
- On peut modifier l'opacité de la couche de classification ainsi que celle des objets sélectionnés de façon à mieux analyser les résultats.
- Vous pouvez aussi effacer la couche de classification.

Utilisation avancée du module Object Labeling Dans cette partie de l'exercice, nous utiliserons ces fichiers supplémentaires : *samples.xml* et *parameters.xml*.

1. Chargez à nouveau dans le module l'image et la segmentation.
2. Chargez le fichier *samples.xml* à l'aide du menu *File/Load Samples*. Quelles sont les classes importées? Combien d'exemples par classe sont utilisés?
3. Dé-sélectionnez toutes les primitives sauf les moyennes radiométriques :
 - Band1::Mean
 - Band2::Mean
 - Band3::Mean
 - Band4::Mean
4. Réalisez la classification. Quels sont les objets de l'image mal classés du fait que leur classe n'existe pas?



5. Quels sont les objets mal classés à cause d'une mauvaise segmentation ou à cause de leur complexité?
6. Essayez d'améliorer la classification en ajoutant des classes manquantes.
7. Essayez d'améliorer la classification en utilisant d'autres primitives.

Trucs et astuces :

- Ce module est très gourmand en mémoire. En fonction de la mémoire disponible sur votre machine, il peut être utile de réinitialiser **Monteverdi**.

Apprentissage actif avec le module Object Labeling

1. Dans l'onglet *Objects*, cliquez sur *Sample* en bas à gauche. Ceci permet d'afficher des objets considérés comme étant difficiles à classer. On utilise ici la technique dite du *margin sampling*.
2. Quels sont les types de régions qui apparaissent comme difficiles à classer?
3. Essayez de créer une classe de rejet pour y ranger des régions de "bruit".
4. Réalisez quelques itérations de plus d'apprentissage actif. Qu'observez-vous?

2.6.3 Solutions

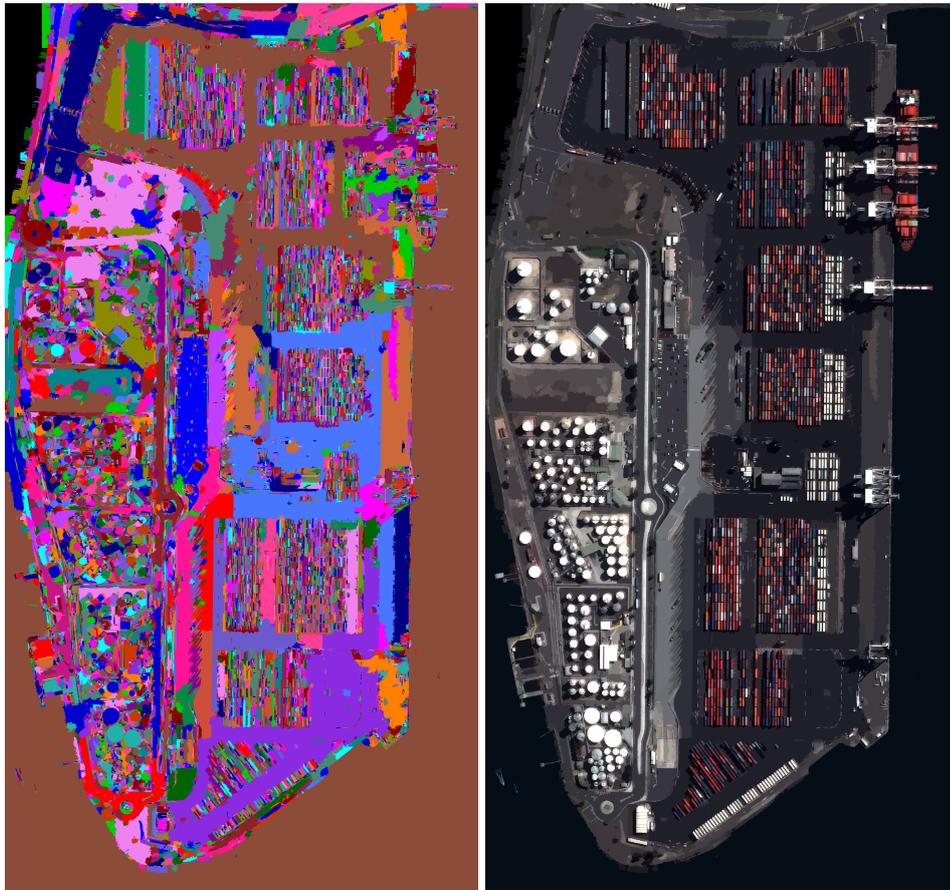
La segmentation préalable

1. Point 1 Voici les lignes de commande pour générer les images couleur :

```
$ otbcli_ColorMapping
  -in phr_pxs_melbourne_xt_small_segmentation.tif
  -out obc_segmentation_optimal.png uint8
  -method optimal
```

```
$ otbcli_ColorMapping
  -in phr_pxs_melbourne_xt_small_segmentation.tif
  -out obc_segmentation_image.png uint8
  -method image
  -method.image.in phr_pxs_melbourne_xt_small.tif
```

Et voici le résultat :



2. Point 2 A partir des résultats de segmentation, on peut supposer que la classification objet marchera bien sur :

- la plupart des cuves circulaires,
- la plupart des containers rectangulaires,
- les classes simples comme l'eau ou les routes.

En revanche, elle marchera plutôt mal sur :

- certains objets circulaires ou rectangulaires qui sont fragmentés par la segmentation,
- des objets complexes comme les bateaux ou les grues,
- des petits objets comme les voitures et les camions.

Introduction au module Object Labeling

1. Point 3 L'onglet *Objects* permet de créer des classes et de leur ajouter des objets pour l'apprentissage. L'onglet *Features* permet de sélectionner les primitives calculées sur les objets qui seront utilisées pour la classification. Enfin, l'onglet *Learning* sert à régler les paramètres de la classification et à exécuter celle-ci.
2. Point 5 La région de la segmentation qui se trouve sous la souris est sélectionné lors du premier click.

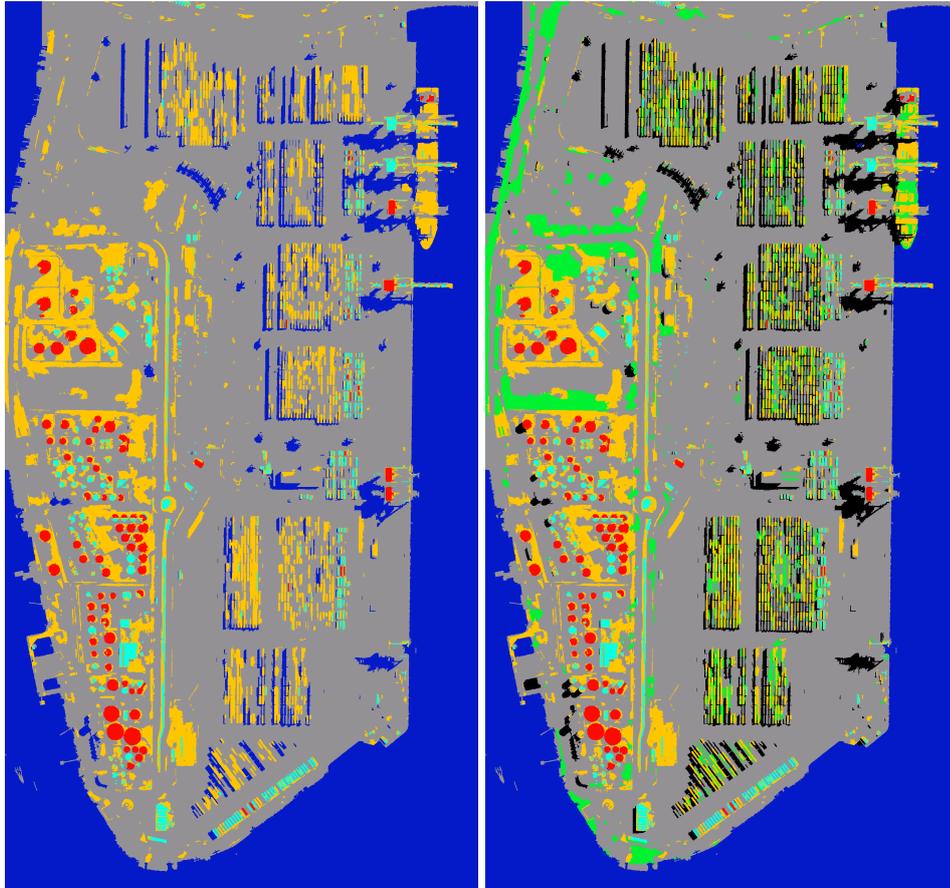
3. Point 6 La région sélectionnée est ajoutée à la classe courante lors du 2^e click.
4. Point 10 Un classifieur SVM est généré par apprentissage sur les objets sélectionnées. Les autres régions de l'image sont classés en utilisant le classifieur ainsi obtenu.
5. Point 11 Le bouton *Save/Quit* permet de fermer le module et de produire les sorties suivantes :
 - une image d'étiquettes correspondant aux classes,
 - une image couleur avec les couleurs des classes utilisées dans le module,
 - une sortie vecteur contenant des polygones étiquetés avec les classes décidées par le classifieur.

Utilisation avancée du module Object Labeling

1. Point 2 Les classes contenues dans le fichier sont :
 - Circular containers
 - Rectangular colored containers
 - Rectangular white containers
 - Water
 - Asphalt
2. Point 4 Les échantillons fournis et les paramètres choisis produisent un résultat où quelques classes de base sont détectées au prix de mauvaises classifications sur des objets difficiles (voir la partie gauche de la figure à la fin de cette section).

On peut noter quelques classes manquantes dans le jeu d'apprentissage et qui induisent des erreurs :

 - Les ombres qui sont classées comme eau. Même si la classe ombre ne fait pas partie des classes d'intérêt, il peut être utile de l'ajouter pour réduire les erreurs.
 - Les zones de végétation, même si elles sont peu nombreuses, seront mal classées, car cette classe n'existe pas dans la nomenclature utilisée.
3. Point 5 Comme prédit dans la section sur la segmentation, quelques objets d'intérêt sont mal segmentés ou trop complexes pour être bien classés :
 - les bateaux et les grues sont trop complexes,
 - certains containers (circulaires ou rectangulaires) sont mal segmentés, ce qui conduit à une mauvaise classification.
4. Point 6 - 7 En ajoutant quelques classes de plus et quelques exemples supplémentaires, nous pouvons obtenir ce genre de résultat :



Apprentissage actif avec le module Object Labeling

1. Point 2 La stratégie mise en oeuvre pour l'apprentissage actif propose souvent des objets difficiles à étiqueter manuellement, car ils correspondent à des parties d'objets mal segmentés ou à du bruit de segmentation.
2. Point 4 Nous pouvons remarquer que, de façon occasionnelle, l'apprentissage actif permet de découvrir des nouveaux objets pour lesquels il n'y a pas de classe encore définie. Parfois, on tombe aussi sur les mêmes objets de façon répétée, car ils restent difficiles à classer après plusieurs itérations.