

Formation à l'Orfeo ToolBox: Guide des Travaux Pratiques

OTB team

2017

Table des matières

1	Avant-propos	2
1.1	Objectifs de la formation	2
1.1.1	Contenu du kit de formation	2
1.2	Les logiciels nécessaires	2
1.3	Les données	3
2	Généralités	5
2.1	Utiliser Monteverdi et QGIS	5
2.1.1	Description	5
2.1.2	Étapes	5
2.2	Le mécanisme des applications Orfeo ToolBox	6
2.2.1	Description	6
2.2.2	Étapes	6
2.2.3	Pour aller plus loin	7
2.3	Les mécanismes internes de l' Orfeo ToolBox	7
2.3.1	Description	7
2.3.2	Étapes	8
3	Imagerie THR optique, des pré-traitements au SIG	9
3.1	Pré-traitements de l'imagerie THR optique	9
3.1.1	Description	9
3.1.2	Étapes	10
3.2	Segmentation et export vers un SIG	11
3.2.1	Description	11
3.2.2	Étapes	12
4	Classification supervisée pour les séries multi-temporelles	13
4.1	Description	13
4.1.1	Résumé	13
4.1.2	Pré-requis	13
4.1.3	Objectifs	14
4.2	Étapes	14
4.2.1	Présentation des données Sentinel-2	14
4.2.2	Réaliser un apprentissage mono-date	15
4.2.3	Identifier la date la plus performante	15
4.2.4	Réaliser la classification et produire une carte en couleur	16
4.2.5	Évaluer la performance globale	16
4.2.6	Régulariser et mesurer le gain de performance	16
4.2.7	Réaliser une classification multi-date et mesurer le gain de performance	16
4.2.8	Pour aller plus loin	16

5 Traitements SAR pour l'imagerie Sentinel 1	17
5.1 Introduction au traitements des images RSO	17
5.1.1 Description	17
5.1.2 Étapes	17
6 Développer avec l'OTB	19
6.1 Tutoriel pour futur développeur OTB	19
6.1.1 Description	19
6.1.2 Étapes	19
6.1.3 Pour aller plus loin	21

1 Avant-propos

1.1 Objectifs de la formation

L'objectif de cette formation est de présenter l'OTB et mettre en œuvre les outils disponibles dans les OTB applications pour l'exploitation, et la manipulation des images de télédétection.

Les capacités acquises en fin de formation concernent la mise en œuvre de chaînes de traitement d'images en utilisant les applications OTB, incluant notamment :

- Extraction de primitives
- Calibration
- Classification
- Segmentation
- Traitements basiques radar

1.1.1 Contenu du kit de formation

Paquet de données

- Contient les données nécessaires à chaque TP, dans des sous-répertoire séparés
- Le sous-répertoire nécessaire est indiqué au début de l'énoncé du TP
- Téléchargeable ici : www.orfeo-toolbox.org/packages/WorkshopData/WorkshopData.zip

Enoncés

- Guide de TP
- Planches (cette présentation)
- Guide d'installation
- Questionnaire d'évaluation
- Solutions (distribuées à la fin)

Documentation des logiciels

Software Guide Guide de l'API C++ (contient également des descriptions algorithmiques)

www.orfeo-toolbox.org/SoftwareGuide/index.html

CookBook Guide pour les non développeurs (contient l'API complète des applications)

www.orfeo-toolbox.org/CookBook/

QGIS User Guide Manuel d'utilisation de QGIS

docs.qgis.org/2.14/en/docs/user_manual

1.2 Les logiciels nécessaires

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

- **Orfeo ToolBox** 5.4 ou plus récent avec les applications
- **Monteverdi** 3.2 ou plus récent



— **QGIS** 2.4 ou plus récent

Pour ce qui concerne l'installation de l'**Orfeo ToolBox** et de **Monteverdi**, vous pouvez consulter le Cookbook ou le guide d'installation.

Pour installer **QGIS** consultez la documentation du logiciel qui est disponible sur le [site du projet](#).

1.3 Les données

Les données utilisées au cours du TP sont des extraits des données suivantes :

Sentinel-2 plusieurs dates concaténées, dénuagés (gap-filling) avec des données terrains de référence au format ESRI Shapefile (pour le TP classification),

Pléiades image PHR Bundle PRIMARY Niveau 1A issue de la RTU (Recette Thématique Utilisateurs Pléiades - projet CNES) sur le site OSR MiPy (sud-ouest Toulouse), acquise en novembre 2013 (©CNES (2013), distribution Airbus DS/ Spot Image),

Sentinel-1 Produit SLC (complexe) mode Strip Map (80 km swath, 5x5 m résolution), bi-polarisation (HH et HV) sur le sud du lac Constance (Allemagne).

Pour obtenir d'autres données et plus d'informations relatives à ces capteurs :

Sentinel-2 produit de niveau 2A disponible sur le site [THEIA](#)

Pléiades [accès privilégié UIA](#) (institutionnels français)

Sentinel-1 Données gratuites disponibles sur le [Hub ESA](#) ou sur le site CNES [PEPS](#)

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.



2 Généralités

2.1 Utiliser Monteverdi et QGIS

2.1.1 Description

Résumé Ce TP est l'occasion de découvrir et de prendre en main les logiciels que nous utiliserons pour visualiser les données et les résultats produits. Ces logiciels sont, au choix, QGIS ou Monteverdi. Les données utilisées lors de ce TP sont les données qui seront utilisées pour la suite de la formation.

Pré-requis

- Installation des logiciels
- Téléchargement des données

Objectifs Les objectifs de ce TP sont les suivants :

- Savoir visualiser une image et régler son rendu dans Monteverdi,
- Savoir visualiser une pile de données dans Monteverdi
- Savoir visualiser une image et régler son rendu dans QGIS,
- Savoir visualiser une données vecteur dans QGIS,
- Savoir visualiser une pile de données dans QGIS,
- Echanger des trucs et astuces.

2.1.2 Étapes

Pour cet exercice, il est proposé de travailler en binôme. A la fin de l'exercice, chaque binôme présentera au reste du groupe une astuce qu'il a découvert en manipulant les logiciels, ou posera une question qui est restée sans réponse au terme de l'exercice.

Visualiser une image dans Monteverdi Ouvrir l'image `phr_xs_osr_mipy.tif` du répertoire `Data/preprocessing` dans Monteverdi.

L'ensemble des raccourcis clavier sont disponibles dans le menu *Aide*.

Modifiez les bandes visualisées et les bornes de radiométrie utilisées pour la visualisation.

Testez les effets *Contraste local*, *Gradient* et *Angle Spectral*. A quoi servent-ils ?

Modifiez le niveau de zoom en utilisant *CTRL + molette souris*, les raccourcis de la barre d'outil ou en éditant directement l'échelle dans la barre d'état en bas à droite.

Fermez Monteverdi.

Visualiser une pile d'images dans Monteverdi Ouvrez les 5 images Sentinel-2 présentes dans le dossier `Data/classification/images/`.

Modifiez les bandes visualisées pour afficher les couleurs naturelles (rouge : bande 3, vert : bande 2, bleu : bande 1). Utilisez le bouton *Appliquer tous les paramètres aux autres couches* pour avoir un rendu uniforme des images.

Utilisez la molette de la souris pour faire défiler les différentes dates.

Testez les effets *Transparence local*, *Echiquier* et *Balayage*. A quoi servent-ils ?

Fermez Monteverdi.

Visualiser une image dans QGIS Ouvrir la première date Sentinel-2 dans QGIS. Utilisez le menu contextuel de la couche pour aller dans *Propriétés* puis dans l'onglet *Style* et régler l'affichage (dynamique et bandes) pour le rendre équivalent à celui de Monteverdi. Toujours avec le menu contextuel, ajouter le style ainsi créé pour pouvoir l'appliquer à d'autres images.

Naviguez en résolution.

Chargez les autres images Landsat-8, et appliquez-leur le style ainsi créé.

Quelles sont les différences entre les fonctionnalités de visualisation d'images offertes par Monteverdi et QGIS ?

Visualiser une donnée vecteur dans QGIS Ouvrez la donnée vecteur `training.shp` située dans le dossier `Data/classification/reference/training/`.

En utilisant le menu contextuel, ouvrez les *Propriétés* puis l'onglet *Style*. En utilisant un rendu *Catégorisé* en fonction de la colonne *LC*. Combien y a-t-il de classes différentes dans le fichier ? Modifiez la table des couleurs pour qu'elles correspondent aux classes.

Ouvrir la table des attributs à l'aide du menu contextuel. Combien y a-t-il de polygones dans la classe *pelouse* ?

A l'aide de l'outil de sélection conditionnel, sélectionnez uniquement les polygones correspondants à cette classe.

2.2 Le mécanisme des applications Orfeo Toolbox

2.2.1 Description

Résumé Au cours de cet exercice, nous allons apprendre à utiliser les applications de l'Orfeo Toolbox. Des images intégrant des messages codés par stéganographie vous sont proposées. A vous d'utiliser les applications de l'Orfeo Toolbox afin de révéler le message contenu dans les images.

Toutes les images d'origine utilisées dans cet exercice sont des extraits d'images Pléiades.

Pré-requis

- Logiciels installés (Monteverdi et Orfeo Toolbox)
- Données téléchargées

Objectifs

- Savoir rechercher une application dans la liste des applications disponibles
- Savoir paramétrer une application
- Savoir trouver la documentation d'une application
- Savoir utiliser plusieurs applications classiques

2.2.2 Étapes

Les données se trouvent dans le répertoire `Data/stegano`.

Pour chacun des messages, on observera d'abord l'image pour essayer de déceler le message, puis on cherchera à utiliser les applications suggérées pour révéler le message.

Message 1 Dans l'image `image1.tif`, une phrase a été encodée dans une zone de forte réflectance, en utilisant une valeur de pixel qui n'est pas admissible pour Pléiades (rappel : les images Pléiades sont codées sur des entiers de 12 bits non signés).

Utilisez l'application **BandMath** pour détecter ces valeurs aberrantes et ainsi révéler le message.

Message 2 Dans l'image `image2.tif`, une phrase a été encodée dans une zone homogène de faible luminance. Les pixels ainsi modifiés ne sont pas détectables à l'oeil nu, mais pourraient être révélés par l'application d'un gradient ou d'un détecteur de contours.

Utilisez l'application **EdgeExtraction** pour révéler le message.

Message 3 Dans l'image `image3.tif`, une phrase a été encodée en modifiant légèrement les valeurs des pixels des bandes rouge et proche infra-rouge. Cette modification est invisible à l'oeil nu, mais peut être révélée en calculant l'indice de végétation NDVI.

Utilisez l'application **RadiometricIndices** pour révéler le message.

Alternativement, utilisez l'application **BandMath** pour calculer l'indice NDVI à l'aide de la formule suivante :

$$NDVI = \frac{NIR-RED}{NIR+RED}$$



Pour rappel, pour l'imagerie Pléiades la bande rouge est la première bande, et la bande proche infra-rouge la dernière.

Message 4 Dans l'image `image4.tif`, un message a été dissimulé dans les 2 bits de poids faible de l'image. Cette transformation est non détectable à l'oeil nu, mais pourrait être révélé en isolant la valeur de ces 2 bits.

Utilisez l'application **BandMath** pour isoler les 2 bits de poids faible dans l'image (encodée sur 12 bits pour rappel), afin de faire apparaître le message.

Notes : La fonction `rint()` permet d'arrondir à l'entier le plus proche dans une expression de l'application **BandMath**.

Message 5 Dans l'image `image5.tif`, un message a été dissimulé en modifiant légèrement et localement le bruit de l'image. Il pourrait être révélé par une transformation permettant d'isoler le bruit.

Utiliser l'application **DimensionalityReduction** pour isoler le bruit de l'image et ainsi révéler le message.

Éventuellement, imaginer d'autres procédés utilisant les applications pour mettre en évidence cette modification locale du bruit.

Message 6 Dans l'image `image6.tif`, un message a été dissimulé en utilisant localement un opérateur morphologique en niveaux de gris (ouverture de rayon 1). Il pourrait être révélé en utilisant la propriété d'idempotence de cette transformation. Pour rappel, une fonction f est dite idempotente si :

$$f(f(x)) = f(x)$$

Utilisez les applications **GrayscaleMorphologicalOperation** et **BandMath** pour révéler le message en vous appuyant sur la propriété d'idempotence.

2.2.3 Pour aller plus loin

Quels messages étaient détectables en analysant finement l'image avec Monteverdi ? Quels messages ne l'étaient pas ?

Pouvez vous imaginer d'autres procédés pour incruster des messages indétectables dans les images ? Une image (`image.tif`) et un message sous la forme d'une image binaire (`message.tif`) vous sont fournies dans le répertoire `Data/stegano/diy` pour vous exercer.

2.3 Les mécanismes internes de l'Orfeo ToolBox

2.3.1 Description

Résumé Cet exercice permet de se familiariser avec les mécanismes internes de l'Orfeo ToolBox :

- Les noms de fichier étendus
- Le streaming
- Le multi-threading
- Les variables d'environnement
- Les fichiers geom

Pré-requis

- Logiciels installés (Monteverdi et Orfeo ToolBox)
- Données téléchargées
- Connaissance du mécanisme des applications de l'Orfeo ToolBox (voir exercice correspondant)

Objectifs

- Comprendre les mécanismes transparents de l'Orfeo ToolBox
- Savoir influencer la manière dont l'Orfeo ToolBox exécute un traitement
- Savoir ou trouver des informations complémentaires

2.3.2 Étapes

Les données se trouvent dans le répertoire `Data/internals/`.

Encodage des images Dans l'Orfeo ToolBox, c'est l'utilisateur qui choisit le type d'encodage des images en fonction de ce que l'image représente. Ainsi, on choisira en général :

- Un encodage entier sur 8 bits non-signés (plage de valeurs [0,255]) pour des sorties destinées à un affichage ou des documents (rapports, articles),
- Un encodage entier sur 16 bits non signés pour représenter des images satellite,
- Un encodage flottant (32 ou 64 bits) quand on représente le résultat d'un calcul produisant des nombres réels (NDVI, calibration radiométrique ...).

Dans les applications de l'Orfeo ToolBox, l'utilisateur peut choisir simplement son encodage (menu déroulant dans les interfaces graphiques ou paramètre additionnel en ligne de commande).

type	plage de valeurs	entier ou flottant	nombre de bits
uint8	[0,255]	entier non signé	8 bits
int16	[-32 767, +32 767]	entier signé	16 bits
uint16	[0, 65 535]	entier non signé	16 bits
int32	[-2 147 483 647, +2 147 483 647]	entier signé	32 bits
uint32	[0, 4 294 967 294]	entier non signé	32 bits
float	[-3.402823 × 10 ³⁸ , 3.402823 × 10 ³⁸]	flottant	32 bits
double	[-10 ³⁰⁸ , 10 ³⁰⁸]	flottant (plus précis)	64 bits

Utilisez **gdalinfo** pour connaître l'encodage des pixels de l'image `image1.tif` (vous pouvez également y accéder dans QGIS). Analysez les valeurs des pixels de l'image dans **monteverdi**. Que pouvez vous conclure ?

Utilisez l'application **Convert** pour convertir l'image `image1.tif` dans un encodage entier sur 16 bits. Comparez la taille de fichier des deux images. Utilisez l'application **CompareImages** pour comparer le contenu des deux images. Que pouvez vous en conclure ? Peut-on réduire encore la taille du fichier image (voir paragraphe sur les noms de fichier étendus) ?

Utilisez l'application **RadiometricIndices** pour calculer un indice de végétation NDVI à partir de l'image en entier sur 16 bits ainsi créée. Conservez l'encodage de la sortie en entier sur 16 bits. Visualisez le résultat. Que constatez vous ? Quel encodage faut-il utiliser pour stocker correctement cette image ?

Les fichiers .geom Examinez le contenu du fichier `image1.geom`. Que pouvez vous en conclure quand à son rôle ? Quelles sont les opérations nécessitant ce type d'informations ?

Les noms de fichiers étendus Les noms de fichiers étendus permettent d'influencer la manière dont l'Orfeo ToolBox lit ou écrit une image. Ils ne sont pas spécifiques aux applications, mais peuvent être utilisés pour n'importe quel logiciel construit autour de l'Orfeo ToolBox.

L'ensemble des options disponibles sont listées dans la section *Extended filename for reader and writer* du Software Guide. Nous allons illustrer leur fonctionnement en nous concentrant sur quelques options.

Les options de lecture Comparez les sorties des deux commandes suivantes :

```
$ otbcli_ReadImageInfo -in "image1.tif"
$ otbcli_ReadImageInfo -in "image1.tif?&skipgeom=true"
```

Quel est l'effet du paramètre *skipgeom* ? Notez son équivalent *skipcarto*, qui permet de s'affranchir de la projection cartographique grossière dans le cas des produits de type *Ortho Ready* (produit projeté à une altitude moyenne par défaut).



Comparez les sorties des deux commandes suivantes :

```
$ otbcli_ReadImageInfo -in "image2.tif"
$ otbcli_ReadImageInfo -in "image2.tif?&geom=image1.geom"
```

Quel est l'effet du paramètre *geom* ? Quelle peut être son utilité ?

Les options d'écriture Parmi les options d'écriture accessibles à partir des noms de fichier étendus, l'option *gdal:co* permet de passer des options de créations du fichier directement à *gdal*.

A l'aide de cette option, et en consultant les options de création de fichier disponibles pour le format Tif dans *gdal*, ré-encodez l'image *image1.tif* en entier signé sur 12 bits, avec une compression LZW. Comparez la taille des fichiers et le contenu des image à l'aide de l'application **CompareImages**.

Une seconde option de nom de fichier étendu qui peut être utile est l'option *box*. Utilisez l'option *box* pour n'écrire qu'un carré de 100x100 pixels au centre de l'image d'entrée.

La gestion du streaming Par défaut, l'Orfeo ToolBox choisit le mode de découpage (tuile ou bandeaux) le plus adapté à l'image d'entrée, et la taille de bloc qui maximise l'occupation mémoire spécifiée dans l'application (ou dans la variable d'environnement *OTB_MAX_RAM_HINT*). Il est cependant possible de modifier ce comportement en utilisant les noms de fichiers étendus à l'écriture.

1. Utilisez l'application **LocalStatisticsExtraction** pour filtrer la première bande de l'image *image1.tif* avec un rayon de 9.

1. Exécutez une seconde fois cette opération, cette fois en désactivant complètement le streaming à l'aide de l'option de nom de fichier étendu *streaming:type*. Que constatez vous ?

1. Exécutez à nouveau cette opération, en spécifiant cette fois un découpage en 1000 bandeaux, en utilisant les options *streaming:type*, *streaming:sizemode* et *streaming:sizevalue*. Que constatez vous ?

Pour mieux observer les phénomènes, il est conseillé d'ouvrir un moniteur système permettant de surveiller l'occupation des processeurs et le débit instantané vers les disques durs.

Le multi-threading Par défaut, tous les filtres de l'Orfeo ToolBox qui en ont la capacité vont automatiquement utiliser l'ensemble des coeurs disponibles sur le processeur de la machine. Ce comportement peut être modifié en utilisant la variable d'environnement *ITK_GLOBAL_DEFAULT_NUMBER_OF_THREADS*.

Reprenez l'exemple de l'exercice précédent utilisant l'application **LocalStatisticsExtraction** en désactivant le streaming. Utilisez la variable d'environnement *ITK* pour diminuer puis augmenter le nombre de threads utilisés. Que constatez vous ?

Vous pouvez utiliser la fonction *time* pour mesurer le temps d'exécution.

3 Imagerie THR optique, des pré-traitements au SIG

3.1 Pré-traitements de l'imagerie THR optique

3.1.1 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

Savoir réaliser une calibration optique

- Pouvoir inter-comparer les valeurs entre images pour le même capteur (multi-temp) et multi-capteurs
- Conversion en réflectance TOA (Top of Atmosphere) et TOC (Top of Canopy)

Savoir réaliser une fusion (pan-sharpening)

- Produit Pan (mieux résolu spatialement) et XS (multi-spectral)
- Tester différentes techniques de fusion
- Voir les spécificités de la fusion en fonction du capteur (mode *Pléiades*)
- Visualisation des images et les effets par la fusion (objets mobiles, indices radiométriques)

Savoir réaliser une orthorectification

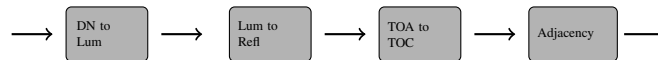
- Spatialiser l'image
- Associer une coordonnée géographique au pixel
- Connaître les pré-requis, les biais, les approximations réalisées...

3.1.2 Étapes

Les données se trouvent dans le répertoire `Data/preprocessing/`. Les sous-répertoires `SRTM` et `Geoid` sont également utilisés.

Dans cette partie de l'exercice, vous utiliserez les images `phr_xs_osr_mipy.tif` et `phr_pan_osr_mipy.tif`. Ce sont des extraits Pan et XS d'un produit Bundle Pléiades Primary (Niveau 1) sur le sud-ouest de Toulouse.

Corrections atmosphériques 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.



Avec l'image `phr_xs_osr_mipy.tif` :

1. Utilisez l'application **OpticalCalibration** pour calculer la réflectance au sommet de l'atmosphère (top of atmosphere).
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 (TOA-TOC). Comparez cette différence pour les bandes rouge, verte et bleue (B0,B1,B2). Quelle bande est la plus *impactée* par la correction TOC ?
4. Appliquez les opérations 1, 2 et 3 à l'image panchromatique `phr_pan_osr_mipy.tif`.

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).

Fusion P+XS L'objectif de cet exercice est de créer une image fusionnée (*Pan-Sharpning*). En raison des contraintes physiques sur la conception du capteur, il est difficile d'obtenir la meilleure résolution spatiale et spectrale en même temps. Pour cette raison la plupart des capteurs spatiaux THR délivrent 2 types d'images :

- Multi-spectrale (XS) : composée de plusieurs bandes, chacune sur une plage de résolution spectrale donnée. Pour Pléiades, 4 bandes (B, V, R, PIR) à une résolution spatiale de 2,8m (rééchantillonnées à 2m)

- Panchromatique (PAN) : image en niveaux de gris avec un détecteur couvrant une plus large partie du spectre ce qui permet d'augmenter le flux et d'améliorer la résolution par rapport à l'image multi-spectrale. 0,7m dans le cas de Pléiades (rééchantillonnée à 0,5m)

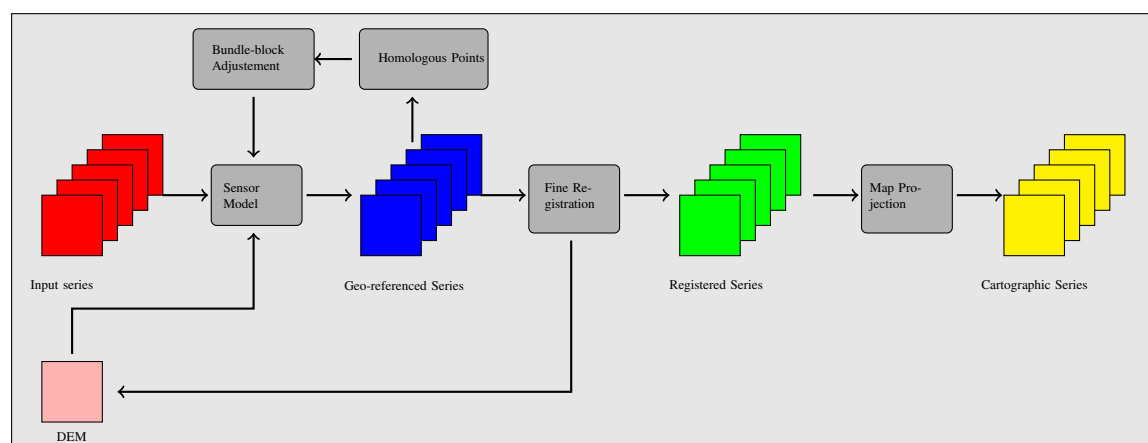
Nous allons réaliser cette fusion à partir des extraits PAN et XS en réflectance au sommet de l'atmosphère `phr_pan_osr_mipy_toa.tif` et `phr_xs_osr_mipy_toa.tif` :

Pour cela :

1. Utilisez l'application **BundleToPerfectSensor** pour superposer et fusionner l'image PAN et XS. Notez que l'application possède un mode *phr* qui permet d'effectuer la superposition des 2 images sans utiliser les modèles de capteurs respectifs des images PAN et XS (mode par défaut). En effet les produits bundle Pléiades ont la particularité d'être sur 2 grilles colocalisées.
2. Quel algorithme de fusion est utilisé dans l'application **BundleToPerfectSensor** ?
3. (optionnel) Utilisez les applications **Superimpose** et **Pansharpening** pour réaliser la même opération en utilisant d'autres méthodes de fusion.

Ortho-rectification Cette opération permet d'associer des coordonnées au sol aux pixels de l'image.

Le schéma ci-dessous décrit l'ensemble des étapes susceptibles d'être réalisées pour passer d'un ensemble de produits de niveau 1 à une pile d'images co-registrées et géo-localisées.



Les modalités de ces étapes peuvent varier en fonction du type de capteurs en entrée.

Utilisez l'application **Orthorectification** pour réaliser l'orthorectification du produit fusionné :

1. Sans DEM
2. Avec un DEM et un geoid (utilisez le sous-répertoire SRTM)
3. Comparez les 2 images en sortie dans Monteverdi. Que constatez vous ?
4. Quel est le système de projection par défaut dans l'application **Orthorectification** ?
5. Dans quelle zone UTM se trouve l'extrait PHR Pléiades ?
6. Testez plusieurs types de projection en sortie (WGS84, Lambert 93)

3.2 Segmentation et export vers un SIG

3.2.1 Description

Résumé Cet exercice permet de se familiariser avec la segmentation large échelle en utilisant l'algorithme MeanShift. On y parcourt les différentes étapes jusqu'à l'export des polygones segmentés dans un SIG.

Pré-requis

- Logiciels installés (Monteverdi et QGIS)
- Données issues du TP de pré-traitements pour l'imagerie THR optique
- Connaissance du mécanisme des applications de l'Orfeo ToolBox (voir exercice correspondant)

Objectifs

- Connaître les étapes pour réaliser une segmentation
- Savoir optimiser les paramètres de la segmentation
- Savoir exporter la segmentation vers un logiciel SIG

3.2.2 Étapes

Les données pour cet exercice se trouvent dans le répertoire `Data/segmentation`.

Lissage de l'image par l'algorithme MeanShift L'algorithme du MeanShift réalise un lissage de l'image tout en préservant ses contours. Il rend ainsi l'image plus homogène tout en conservant des frontières franches entre les zones. Il facilite ainsi la segmentation de l'image par des méthodes simples, comme l'extraction des composantes connexes. L'algorithme moyenne itérativement les pixels dans un certain voisinage spatial et spectral.

Réalisez un lissage de l'image `phr_orthopxs_osr_mipy_xt.tif` en utilisant l'application **MeanShiftSmoothing**. Faites varier les paramètres *ranger* et *spatialr* pour observer leur influence.

Notes :

- L'option *modesearch* permet d'accélérer le calcul au détriment de la stabilité, et sera donc désactivée,
- Le rayon spatial (*spatialr*) et spectral (*ranger*) sont les paramètres déterminants pour l'algorithme. Le premier s'exprime en nombre de pixels, tandis que le second comme une distance entre signatures spectrales des pixels.
- La sortie *foutpos* code les position moyennes des pixels après lissage, et sera utilisée dans la suite du TP.
- Le paramètre de nom de fichier étendu *box* peut être utilisé pour trouver la valeur de *ranger* et *spatialr* optimale sans traiter toute l'image.

Pour la suite de l'exercice, utilisez une valeur de *ranger* autour de 25 et une valeur de *spatialr* autour de 3 (pour limiter le temps de calcul).

Segmentation Cette étape va produire une segmentation initiale à partir des images écrites précédemment (sorties *fout* et *foutpos*). Les pixels adjacents dont les valeurs dans l'image *fout* sont distantes de moins de *ranger* et dont les positions estimées dans l'image *foutpos* sont distantes de moins de *spatialr* seront groupés au sein de la même composante connexe.

Le calcul est fait par tuile, dont on peut fixer la taille au moyen des paramètres *tilsizex* et *tilsizey*.

Enfin, le paramètre *minsize* permet d'éliminer toutes les régions dont la taille produite est trop petite (seuil exprimé en nombre de pixels).

Utilisez l'application **LSMSSegmentation** pour réaliser cette étape. A noter :

- Utilisez un *minsize* de 0 dans cette première étape,
- L'image de sortie sera une image étiquetée, dans laquelle chaque pixel porte un label entier unique correspondant au segment auquel il appartient. Il est recommandé d'encoder cette sortie en *uint32* afin de disposer de suffisamment de labels uniques. En effet la segmentation génère potentiellement un grand nombre de segments.
- Les valeurs pour les paramètres *ranger* et *spatialr* doivent être inférieures aux valeurs utilisées pour le lissage (étape précédente). Des valeurs d'environ la moitié de celles utilisées lors de l'étape précédente sont un bon point de départ.

Utilisez la méthode *optimal* de l'application **ColorMapping** afin de créer une carte contrastée des régions segmentées, qui sera plus facile à analyser.



Traitement des petites régions Une fois la segmentation initiale réalisée, il reste en général beaucoup de régions de petite taille, qui ne correspondent à aucun objet d'intérêt. L'application **LSMSSegmentation** permet de les filtrer (paramètre *minsize*), mais il est également possible de réaliser un post-traitement pour fusionner itérativement les régions de taille trop faible avec les régions adjacentes dont la radiométrie est la plus proche. C'est l'objet de l'application **LSMSSmallRegionsMerging**.

Utilisez cette application pour retraiter la segmentation initiale, en fusionnant toutes les régions dont la taille est inférieure à 50 pixels.

Utilisez à nouveau la méthode *optimal* de l'application **ColorMapping** afin de créer une carte contrastée des régions segmentées, et comparer avec la carte de segmentation précédente.

Vectorisation La dernière étape consiste à exporter les polygones ainsi créés vers un fichier vectoriel. L'application **LSMSVectorization** permet d'effectuer cette étape, en calculant au passage la moyenne et la variance de chaque polygone en fonction d'une image support.

Commencez par calculer une image de NDVI à l'aide de l'image initiale (l'ordre des bandes est Rouge, Vert, Bleu, Proche Infra-Rouge).

Utilisez ensuite l'application **ConcatenateImages** pour former une image contenant les radiométries initiales ainsi que la bande de NDVI. Prenez garde à l'encodage du fichier de sortie.

Utilisez l'application **LSMSVectorization** en utilisant l'image ainsi créée et l'image de segmentation en entrée. Choisissez un format *ESRI Shapefile* (extension shp) pour le fichier de sortie.

Ouvrez l'image initiale ainsi que le fichier vecteur ainsi créé dans QGIS. Changez le style d'affichage de la couche vecteur pour afficher la bordure des polygones en rouge et le corps transparent. Analysez plus finement la segmentation ainsi affichée.

Ouvrez la table des attributs de la donnée vectorielle. Que constatez vous ?

Filtrage des polygones dans QGIS Utilisez l'outil de sélection par une expression de la table des attributs dans QGIS pour sélectionner les segments qui ne sont pas des ombres, et copiez les dans une nouvelle couche vecteur.

Utilisez la calculatrice de champs de la table des attributs pour créer un nouvel attribut représentant la compacité :

$$compactness = \frac{\sqrt{area}}{perimeter}$$

Utilisez l'outil de sélection par une expression de la table des attributs dans QGIS pour sélectionner les petits objets compacts dont la valeur moyenne du NDVI est forte (arbres ?). Copiez-les dans une nouvelle couche.

4 Classification supervisée pour les séries multi-temporelles

4.1 Description

4.1.1 Résumé

Cet exercice permet de se familiariser avec les applications de classification supervisée pixellique de l'Orfeo ToolBox, en utilisant une série multi-temporelle Sentinel-2 et un jeu de données de référence pour la supervision.

4.1.2 Pré-requis

- Logiciels installés (Monteverdi et Orfeo ToolBox)
- Données téléchargées
- Connaissance du mécanisme des applications de l'Orfeo ToolBox (voir exercice correspondant)
- Notions de classification supervisée

4.1.3 Objectifs

Les objectifs sont les suivants :

- Connaître les différentes applications constituant la procédure de classification supervisée
- Utiliser différents algorithmes pour l'apprentissage
- Savoir mesurer les performances de la classification
- Connaître les post-traitements applicables à une classification

4.2 Étapes

Les données sont disponibles dans le répertoire `Data/classification`, avec les sous-répertoires suivants :

- `images` contient la série multi-temporelle Sentinel-2,
- `references/training` contient la donnée d'apprentissage au format *shp*,
- `references/testing` contient la donnée de validation au format *shp*.

4.2.1 Présentation des données Sentinel-2

Dans l'archive de données, le dossier `Data/classification/images` contient 5 images Sentinel-2, extraites de la tuile T31TCJ, aux dates suivantes :

 2016-06-07
 2016-07-07
 2016-08-06
 2016-09-05
 2016-10-05

Ces images sont toutes multispectrales avec 10 bandes ré-échantillonnées à 20 m :

#	Band name	S2 band id	Wavelength	Initial resolution
0	Blue	B2	490 nm	10 m
1	Green	B3	560 nm	10 m
2	Red	B4	665 nm	10 m
3	NIR - Narrow 1	B5	705 nm	20 m
4	NIR - Narrow 2	B6	740 nm	20 m
5	NIR - Narrow 3	B7	783 nm	20 m
6	NIR - Wide	B8	842 nm	10 m
7	NIR - Narrow 4	B8A	865 nm	20 m
8	SWIR 1	B11	1610 nm	20 m
9	SWIR 2	B12	2190 nm	20 m

Au total, c'est donc 50 bandes qui représentent chaque pixel. Les images sont encodés sur 16 bits.

Ouvrez une image dans `monteverdi` et réglez les bandes pour un affichage en vrais couleurs (rouge, vert, bleu).

Ouvrez les cinq images et remarquez les changements.

Note : Le fichier de style `support/images.qml` peut être chargé dans `QGis` pour régler la dynamique et la composition colorée de chaque image à l'identique.

Les fichiers `references/training/training.shp` et `references/testing/testing.shp` contiennent des polygones qui définissent 13 classes sur l'ensemble de la scène :



Code	Nom	#polygones training	#polygones testing
10	Cultures annuelles	3129	3078
31	Forêt feuilles caduques	176	292
32	Forêt feuilles persistantes	23	29
34	Pelouses	2	2
36	Lande ligneuse	63	38
41	Bâti dense	30	33
42	Bâti diffus	326	239
43	Zones industrielles	154	212
44	Routes	162	114
51	Eau	243	332
211	Prairie	320	311
221	Verger	227	254
222	Vigne	129	97

Ouvrez un des fichiers de polygones dans QGIS. La table d'attributs est accessible depuis clic-droit sur la couche -> *Ouvrir la table des attributs*. Chaque label est visible et la liste est filtrable par expression SQL.

Note : Le fichier de style `support/polygons.qml` peut être chargé dans QGIS pour coloriser les polygones en fonction de leur classe.

Les polygones sont répartis en deux ensembles : apprentissage (training) et validation (testing).

4.2.2 Réaliser un apprentissage mono-date

Nous allons maintenant utiliser l'application **TrainImagesClassifier** afin de réaliser l'apprentissage supervisé à partir des données d'entraînement disponibles dans `references/training/training.shp`. Pour commencer, nous allons réaliser cet apprentissage avec uniquement l'image du 07.06.2016.

L'application **TrainImageClassifier** va échantillonner certains pixels de l'image contenus dans les polygones de la vérité terrain, afin de constituer un ensemble d'apprentissage équilibré. Celui-ci est ensuite transmis à l'algorithme d'apprentissage.

Cette application prend en paramètres obligatoires :

- L'image dont les bandes seront utilisées comme descripteurs pour l'algorithme de classification,
- La couche vecteur contenant les polygones de référence,
- Le nom du champ correspondant à la classe d'occupation du sol dans cette couche vecteur,
- Le fichier sortie ou stocker le modèle appris (on peut l'appeler `model.rf`).

Régler certains paramètres optionnels comme suit :

- Le classifieur "Random Forests" pour l'algorithme d'apprentissage,
- Le nombre d'arbres à 50,
- La profondeur maximale de l'arbre à 20,
- Le nombre minimum d'échantillons pour chaque noeud à 70,
- Le nombre de clusters à 13 (équivalent au nombre de classes)

Examiner les logs de l'application, en particulier la matrice de confusion, la valeur du coefficient Kappa et les scores par classe. Que constatez-vous ? En l'absence de polygones dédiés à la validation, l'application utilise une partie des échantillons générés pour la validation. Que peut-on en déduire quand aux performances affichées ?

Refaire l'apprentissage, cette fois en utilisant les données de validation `reference/testing/testing.shp` comme vecteur de validation (vous donnez donc à l'application deux fichiers shp différents). Que constatez-vous ?

Refaire l'apprentissage, en désactivant l'option `cleanup`. Regardez les données intermédiaires qui ont été générées. A quoi correspondent elles ?

4.2.3 Identifier la date la plus performante

Réitérer l'apprentissage pour chacune des dates. Quelle date fournit la meilleure performance ? Le coefficient Kappa change-t-il beaucoup ?

Rejouer l'apprentissage correspondant à la meilleure date afin de conserver le fichier de modèle `model.rf`.

4.2.4 Réaliser la classification et produire une carte en couleur

Utiliser l'application **ImageClassifier** pour produire la carte de classification correspondant à la meilleure date (celle du 05.09.2016). Attention à bien utiliser le fichier de modèle correspondant entraîné à partir de cette date.

La sortie de l'étape précédente est une image .tif qui associe à chaque pixel une classe. Pour visualiser cette image, l'application **ColorMapping** permet d'associer à chaque label une couleur RGB et de générer une image de visualisation.

Utilisez le mode **custom** l'application **ColorMapping** avec la table de couleur fournie `support/color_map.txt` pour produire une carte colorisée.

Note : Il se peut que l'image ne s'affiche pas correctement dans Qgis, du fait d'une valeur non renseignée (no data) par défaut enregistrée dans le fichier. La prise en compte du nodata peut être désactivé dans les propriétés de la couche dans Qgis.

4.2.5 Évaluer la performance globale

Nous allons maintenant utiliser l'application **ComputeConfusionMatrix** afin de calculer la performance globale de la classification. Par rapport à l'évaluation des performances réalisée lors de l'apprentissage, cette application permet de :

- Prendre en compte l'ensemble des pixels disponibles dans la donnée de référence,
- Évaluer la performance d'une carte de classification qui a été

retraitée (par exemple avec une régularisation).

Le paramètre `ref.vector.field CODE` est nécessaire. Il indique le nom du champ contenant le numéro de label.

Calculer la performance globale de la classification. Que constatez vous par rapport à la performance évaluée lors de la phase d'entraînement ? Comment expliquer ce phénomène ?

4.2.6 Régulariser et mesurer le gain de performance

Nous allons utiliser l'application **ClassificationMapRegularization**. Elle filtre une image classifiée en utilisant un vote majoritaire local.

Les paramètres à régler sont :

ip.radius 1 Rayon de la zone participant au vote

ip.suvbool 0 Que faire lors d'une égalité. 0 pour utiliser la valeur existante.

Filtrez le résultat de la classification précédente. Évaluer la performance globale de la carte de classification filtrée. Que constatez-vous ?

4.2.7 Réaliser une classification multi-date et mesurer le gain de performance

Nous allons maintenant utiliser l'ensemble des dates pour la classification. A cet effet, vous pouvez utiliser le fichier `images/all.vrt` qui contient l'ensemble des bandes de chaque dates concaténées (c'est donc une image à 50 bandes).

Rejouer l'ensemble du TP avec cette image de 50 bandes. Quel est l'apport de la série multi-temporelle pour la performance de classification ?

Comparez dans Qgis les deux cartes de classification régularisées.

4.2.8 Pour aller plus loin

1. Peut on obtenir de meilleure performance avec d'autres algorithmes de classification ?
2. A l'aide de Qgis, fusionner dans la donnée de référence les classes pelouse et lande ligneuse. Quelle performance obtenez vous ?



3. L'application `TrainImagesClassifier` contient également un algorithme de classification non-supervisée (Shark KMeans). Comparer le résultat d'une classification supervisée et non-supervisée avec la même image.

5 Traitements SAR pour l'imagerie Sentinel 1

5.1 Introduction au traitements des images RSO

5.1.1 Description

Résumé Cet exercice constitue une introduction au traitement et à l'analyse des images RSO (radar à synthèse d'ouverture) en utilisant l'Orfeo ToolBox.

Il existe par ailleurs de nombreuses sources bibliographiques et de cours en ligne disponibles sur la télédétection radar. Par exemple :

- Support de cours ENSG : <http://fad.ensg.eu/moodle/course/view.php?id=519>

Pré-requis

- Logiciels installés (Monteverdi et Orfeo ToolBox)
- Données téléchargées
- Connaissance du mécanisme des applications de l'Orfeo ToolBox (voir exercice correspondant)

Objectifs Les objectifs de cet exercice sont les suivants :

- Savoir manipuler et visualiser des images
- Savoir réaliser une calibration radiométrique
- Savoir réaliser une correction géométrique
- Savoir utiliser le filtrage du speckle
- Savoir réaliser une analyse polarimétrique simple
- Savoir réaliser une extraction de primitives simple

5.1.2 Étapes

Les données pour cet exercice se trouvent dans le répertoire `Data/sar`.

Introduction à l'imagerie RSO Dans cet exercice nous utiliserons les extraits de l'image Sentinel 1 SLC : `s1_hh.tif` et `s1_hv.tif`. L'image se situe en Allemagne au sud du lac Constance (47.456276, 9.638616).

Les extraits sont en géométrie capteur et sont inversées dans le sens EST/OUEST.

1. Ouvrir les images dans Monteverdi. Combien de bandes a chaque image ?
2. A quoi correspondent ces bandes ?
3. Calculer l'image d'intensité à partir des produits complexes `s1_hh.tif` et `s1_hv.tif`. Calculer aussi l'intensité en décibels.

Calibration radiométrique Les valeurs des pixels dans les produits SAR comme Sentinel-1 sont sans unité (radar reflectivity ou radar brightness).

Le produit est accompagné de métadonnées permettant de transformer ces valeurs en grandeurs physiques qui permettent de comparer des images radars provenant de capteurs différents ou de capteurs identiques dans le cas multi-temporel.

On veut de l'image d'entrée les caractéristiques physiques originales des valeurs de rétrodiffusion. Les valeurs suivantes peuvent être calculées :

- β_0 : luminosité radar (slant range)
- σ_0 : rétrodiffusion radar (directement relié au propriété du sol)
- γ_0 : rétrodiffusion radar normalisée par l'angle d'incidence (à privilégier généralement)

Pour Sentinel-1 est fourni une table de valeurs permettant de faire la conversion depuis les comptes numériques vers ces différentes grandeurs.

1. Quelle application permet d'effectuer cette opération dans l'OTB ?
2. Effectuer la calibration des extraits complexes HH et HV et calculer les grandeurs γ_0 (gamma naught en anglais).
3. Convertir les images calibrées en décibel (dB).

Corrections géométriques La géométrie des images est primordiale pour établir des liens entre les valeurs observées et les surfaces terrestres.

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

Nous allons utiliser ici la sortie de la partie calibration optique (γ_0).

Utilisez l'application **Orthorectification** pour réaliser l'orthorectification des extraits des produits SLC Sentinel-1 :

1. Sans DEM
2. Avec un DEM et un geoid (utilisez le sous-répertoire SRTM)
3. Comparez les 2 images en sortie dans Monteverdi. Que constatez vous ?
4. Quel est le système de projection par défaut dans l'application **Orthorectification** ?

Filtrage du speckle Les images SAR sont fortement affectées par le chatoiement (speckle) qui constitue un type particulier de bruit présent dans tous les systèmes d'acquisition cohérents (sonar, laser, etc.). Ce bruit est très fort et il a un effet multiplicatif.

Il existe plusieurs méthodes pour réduire ce bruit. Dans la suite nous allons utiliser le filtre de *Frost* qui possède 2 paramètres : rayon : taille de la fenêtre et *deramp* qui contrôle la décroissance d'une fonction exponentielle qui est utilisée pour pondérer la distance entre le pixel central et son voisinage.

Plus de détail sur le speckle et le filtre de Frost [ici](#).

1. Quelles sont les méthodes de réduction du speckle disponibles dans l'OTB ?
2. Utilisez le filtre de Frost avec différents rayons (3, 5 et 10) et commentez l'effet sur l'image de sortie
3. Commentez la forme de l'histogramme des images filtrées comparées à l'image d'intensité ?
4. Utilisez le filtre de Frost avec un rayon de 5 et différentes valeurs du paramètre *deramp* (0.5, 0.2 et 0.8). Commentez l'effet de ce paramètre.

Polarimétrie Nous allons comparer de manière qualitative la différence entre les observations des images en polarisations HH et HV pour détecter certains éléments dans l'image.

1. Calculez la différence entre l'image d'intensité HH et HV. On peut prendre 2 fois HV car la valeur rétrodiffusée en HV est généralement inférieure. Sur cette, on peut utiliser la différence simple HH-HV
2. Effectuez une composition colorée avec les bandes HH, HV et HH-HV. On peut prendre 2 fois HV car la valeur rétrodiffusée en HV est généralement inférieure.
3. Convertissez la composition colorée en Décibel (Db)
4. Affichez le résultat avec Monteverdi et commentez les différences entre HH et HV (zone de layover, zone de végétation, sol nu, zone en eau...)
5. Jouons un peu pour finir à *Où est Charlie ?*. A partir de la composition colorée en Décibel :
 - Indiquer les coordonnées de 2 lignes électriques parallèles visible dans l'image ?
 - Plus dur, chercher dans l'image une zone réagissant comme un coin réflecteur ("croix").
 - A quoi correspondent les "points" en violet dans le lac autour des coordonnées (930,1170) ?
 Vous pouvez pour faciliter l'analyse : visualisez la zone d'étude dans [Google Maps](#).



Extraction d'information Pour aller plus loin on pourra ensuite explorer les primitives et les algorithmes disponibles pour l'extraction de données dans les images SAR.

Voir filtre de Touzi dans l'application **EdgeExtraction** par exemple.

6 Développer avec l'OTB

6.1 Tutoriel pour futur développeur OTB

6.1.1 Description

Résumé Au cours de ces exercices, nous allons apprendre à développer en utilisant l'API en C++ de la librairie Orfeo ToolBox. Après avoir configuré et compilé un programme informatique simple "Hello World" dont le but est de faire la démonstration rapide de la configuration et de l'utilisation de l'OTB, nous allons apprendre les mécanismes permettant de chaîner des traitements (filtres dans le langage OTB), de lire une image, d'effectuer un traitement sur l'image et d'écrire le résultat dans un nouveau fichier. Ensuite nous apprendrons à créer notre propre brique de traitement et nous verrons enfin comment la packager dans une application OTB et enfin un module externe.

Pré-requis

- SDK OTB
- Environnement de développement (CMake, compilateur, éditeur)
- Données téléchargées

Objectifs

- Savoir configurer et compiler un programme en C++ qui utilise la librairie OTB avec l'outil CMake
- Savoir créer un programme C++ câblant plusieurs étapes d'une chaîne de traitements d'images en utilisant l'API C++ de l'OTB. Les étapes à enchaîner :
 - Lecture de l'image à partir d'un fichier en entrée
 - Traitement de l'image (calcul de gradient)
 - Écriture de l'image de gradient dans un fichier en sortie
- Savoir coder son propre traitement unitaire dans un filtre OTB
- Savoir créer une application OTB qui embarque sa chaîne de traitements
- Savoir packager son traitement et son application *custom* dans un module externe OTB

6.1.2 Étapes

Les exercices de ce TP sont fortement inspirés du cours ITK ci-dessous :

<http://www.na-mic.org/svn/NAMICSandBox/trunk/ITKAdvancedCourse/>

Les données se trouvent dans le répertoire `Data/otb_developers`.

Chacun des exercices suivants se présente sous la forme d'un projet CMake contenant le code source de l'exercice. Ce code source est à trous, c'est à dire qu'il contient un certain nombre d'instructions manquantes identifiées dans le code par des commentaires `TODO`. A noter que tous les exercices compilent sans erreurs même avec ces instructions manquantes.

Pour chaque exercice, il s'agit :

- D'implémenter les fonctionnalités listées
- Vérifier que le code compile
- Vérifier que le programme implémente la fonctionnalité attendue

Pour compiler un exercice, on procède de la manière suivante :

Sous Windows :

```
$ cd OTB-5.10.1-xdk-Linux64
$ . ./otbenv.profile
$ cd ..
$ mkdir build_ex1
```

```
$ cd build_ex1
$ cmake $DATA/dev_exercises/Exercises/ex1>HelloWorld/ -G"Make Makefiles"
$ nmake
$ ./HelloWorld.exe
OTB Hello World !
```

Sous Linux :

```
$ cd OTB-5.10.1-xdk-Linux64
$ . ./otbenv.profile
$ cd ..
$ mkdir build_ex1
$ cd build_ex1
$ cmake $DATA/dev_exercises/Exercises/ex1>HelloWorld/
$ make
$ ./HelloWorld
OTB Hello World !
```

Sous Mac OS X :

```
$ cd OTB-5.10.1-xdk-Linux64
$ . ./otbenv.profile
$ cd ..
$ mkdir build_ex1
$ cd build_ex1
$ cmake $DATA/dev_exercises/Exercises/ex1>HelloWorld/ -DCMAKE_CXX_FLAGS='-std=c++11'
$ make
$ ./HelloWorld
OTB Hello World !
```

Exercice 1 : Hello World Il s'agit de modifier le programme `HelloWorld.cxx` pour :

- Déclarer un alias à l'aide du mot clé `typedef` pour le type `Image` de dimension 2 pour des pixels de type entier non signé (`unsigned int`).
- Créer un objet image avec ce type

Documentation utile : section 5.1.1 du Software Guide.

Exercice 2 : Pipeline OTB Il s'agit de modifier le programme `Pipeline.cxx` pour :

- Modifier le programme pour effectuer le calcul de la norme du gradient de l'image en entrée et écrire le résultat dans une autre image en sortie

Documentation utile : section 4.3 du Software Guide

Exercice 3 : Écrire un filtre OTB Il s'agit de modifier le programme `DivideByTwoImageFilter.h` pour :

- Implémenter le foncteur permettant de réaliser la division par 2 de tous les pixels de l'image
- Valider l'implémentation à partir de l'exécutable en utilisant une des images utilisées pendant la formation

Modifier ensuite le fichier `DivideImageFilter.h` pour implémenter un foncteur réalisant la division de chaque pixel par un diviseur passé en paramètre du filtre

Documentation utile : <http://www.na-mic.org/svn/NAMICSandBox/trunk/ITKAdvancedCourse/doc/presentations/Insight-Writing-a-New-Filter.ppt> , planches 11 à 16



Exercice 4 : Écrire une application OTB Il s'agit de modifier le programme `MyApp.cxx` pour :

- Déclarer un nouveau paramètre de type `float` permettant de modifier la valeur
- Modifier la méthode `DoExecute` pour que l'application réalise la division de tous les pixels de l'image par la valeur passée en paramètre de l'application

Documentation utile : Chapitre 30 du Software Guide

Exercice 5 : Écrire un module externe OTB Il s'agit de modifier le modèle de remote module fournit pour :

- Modifier les fichiers CMake pour changer le nom du remote module
- Insérer les fichiers sources correspondant au filtre `DividerImageFilter.h` développés dans l'exercice 3
- Insérer l'application développée à l'exercice dans le répertoire `app`
- Ajouter un test de non régression de l'application développée

Documentation utile : Chapitre 31 du Software Guide

6.1.3 Pour aller plus loin

OTB Software Guide <https://www.orfeo-toolbox.org/SoftwareGuide/>

OTB module template <https://github.com/orfeotoolbox/otbExternalModuleTemplate>

ITK Software Guide <http://www.itk.org/ItkSoftwareGuide.pdf>

ITK CourseWare https://itk.org/Wiki/ITK/Course_Ware