



## Thème 06 - Bruit et filtres

Ouvrir l'image *plant-noise.tif* à partir du dossier 06 - *noise and filter*. L'image contient un niveau élevé de bruit. Zoomer sur l'image. Le fond devrait être homogène, il contient en fait une distribution aléatoire des intensités. La même distribution aléatoire est ajouté au signal.

### Exercice 6.1 – Bruit

Le bruit peut poser un problème pour l'analyse d'image, par exemple dans la séparation des objets de l'arrière-plan. On devra pré-traiter notre image afin de réduire l'effet perturbateur de bruit. Sur l'image *plant-noise.tif* essayez d'utiliser le seuillage pour séparer la plante de l'arrière-plan. Est-ce possible?

### Exercice 6.2 - filtre de convolution - Mean Filter

Une façon évidente de lisser notre image et de réduire l'impact du bruit est de remplacer chaque pixel dans l'image par une moyenne des intensités des pixels voisins. C'est ce qu'on appelle un filtre moyenneur. Dans ImageJ vous pouvez le trouver dans **Process>Filters>Mean**.

a) Appliquer le *Mean Filter* à l'image *plant-noise.tif*. Comparez l'histogramme avant et après l'application du filtre. Quel rayon faut-il utiliser pour pouvoir séparer les plantes de l'arrière-plan en utilisant le seuillage?

b) Imaginez que votre image est un rectangle blanc (intensité 255). Quel sera le résultat de l'application d'un *Mean Filter* de rayon 1?

Ouvrir *rectangle.tif* pour appliquer le filtre et vérifier son effet.

c) Utiliser *rectangle.tif* pour comparer les résultats de la commande **Process>Smooth** avec l'application d'un *Mean Filter* de rayon 1.

d) Utiliser **Process>Filter>Convolve** pour créer une filtre moyenneur. Comment est le noyau ?

Sur l'image *plant-noise.tif*, est-ce que le résultat du filtre de convolution est exactement le même que le résultat de la commande **Process>Filters>Mean**?

### Exercice 6.3 - filtre de convolution - Flou gaussien

Un filtre de flou gaussien est similaire au filtre moyenneur, mais au lieu d'utiliser des coefficients uniformes dans le noyau, les coefficients suivent une distribution normale, appelée aussi la distribution gaussienne.

- Ouvrir le fichier *Gaussian.txt*. Ici vous voyez les valeurs d'une distribution normale. Pour la voir en image utilisez **File>Import>Text Image**. Appliquer la LUT *Ice*. Maintenant, allez à **Analyze>Surface Plot** ou utiliser le plugin *Interactive 3D surface plot (Plugins> 3D)* afin d'obtenir une impression de la forme 3D.
- Appliquer le filtre Flou gaussien avec un sigma 1.2 dans **Process>Filters>Gaussian blur** sur l'image *rectangle.tif* et comparer le résultat avec un *Mean Filter* de rayon 3.
- Appliquer un filtre de flou gaussien de sigma 6 et un *Mean Filter* de rayon 15 à l'image *roots.tif* et comparer les résultats.
- Ouvrir l'image *plant-noise.tif*. Ouvrez l'outil de convolution et de copier dedans le fichier *Gaussian.txt*. Exécutez le filtre de convolution sur l'image *plant-noise.tif*.

### Exercice 6.4 – Filtre d'amélioration de contour - filtre de Sobel et Prewitt

D'autres noyaux peuvent être utilisés pour créer des filtres améliorant les contours.

- Le filtre de Prewitt est défini par les noyaux:

kx:

-1 0 1

-1 0 1

-1 0 1

et ky:

1 1 1

0 0 0

-1 -1 -1

. Si  $D_x$  et  $D_y$  sont les résultats de la convolution par  $k_x$  et  $k_y$  de l'image d'origine, alors vous devez calculer

$$\sqrt{D_x^2 + D_y^2}$$

pour appliquer le filtre de Prewitt .

Appliquer le à *rectangle.tif*. Créer une image composite de l'image originale et de l'image filtrée (**Image>Color>Merge Channels**)

Conseils:

- convertir d'abord l'image à 32 bits, faire une copie et utiliser l'outil de convolution avec les deux noyaux
  - vous trouverez les opérations carré et racine carrée dans **Process>Math**
  - utilisez le calculateur d'image (**Process>Image Calculator...**) pour ajouter deux images
- Appliquer le filtre de Prewitt à *roots.tif*. Ne pas oublier de le convertir en 32 bits. Créer une superposition de l'image originale et de l'image filtrée.
  - Sobel est un autre filtre améliorant les contours. Il fonctionne de la même manière mais avec le noyaux:

kx:

-1 0 1

-2 0 2

-1 0 1

et k y:

1 2 1

0 0 0

-1 -2 -1

. Appliquer le filtre Sobel à l'image *rectangle.tif* et comparer le résultat avec le résultat de la commande **Process>Find Edges**.

d) Un filtre passe-haut peut être utilisé pour obtenir un effet de netteté à une image. Convoluer l'image de *ophrys.tif* avec le noyau

-1 -1 -1

-1 12 -1

-1 -1 -1

et comparer le résultat avec le résultat de la commande **Process>Sharpen**.

e) Un laplacien de gaussienne (LoG) aussi appelé « Chapeau Méxicain » applique un filtre de lissage gaussien et un filtre d'amélioration de contour en une seule étape. Il peut être utilisé pour détecter les taches.

Des approximations possibles pour le noyau Laplacien sont:

0 -1 0

-1 4 -1

0 -1 0

ou

-1 -1 -1

-1 8 -1

-1 -1 -1

Ouvrir l'image *cell.tif* à partir du dossier 06 - noise and filter

. Vous pouvez appliquer un filtre Log en appliquant d'abord un filtre gaussien puis une convolution avec l'un des noyaux ci-dessus.

Cependant, la gaussienne et le laplacien peuvent être combinés en un seul noyau. Installez le plugin LoG à partir de <http://bigwww.epfl.ch/sage/soft/LoG3D/>. Essayez de seuiller les points lumineux sur l'image.

Exécutez le plugin LoG sur l'image. Regarder le noyau en tant qu'image . Essayez de seuiller les taches après l'application du filtre.

## **Exercice 6.4 – Filtre non linéaire - filtre médian**

Les filtres non linéaires (*Ranking filters*) sont un autre type de filtres. Encore une fois nous regardons, pour chaque pixel, les pixels voisins d'un rayon r. Mais cette fois les valeurs d'intensité sont triées et le pixel est remplacé par la valeur qui est à une position spécifique. Si nous utilisons la position du milieu nous obtenons un filtre médian.

a) Ouvrir *plant-sp-noise.tif*. Zoomer sur l'image. Vous voyez que dans ce cas le bruit consiste en l'ajout de pixels noirs et blancs à l'image. Ce type de bruit est appelé «poivre et sel». Appliquer un filtre médian de rayon 3 (**Process>Filters>Median**). Comparer le résultat avec le résultat du filtre moyennneur.

b) Comparer l'effet du filtre médian de rayon 1 et de la commande **Process>Noise>Despeckle**.

c) Dans **Process>Noise>Remove Outliers**, vous trouverez un filtre médian sélectif, qui remplace un pixel par la valeur médiane si la valeur du pixel diffère de plus d'un seuil avec la cette valeur médiane. Utilisez-le pour supprimer séparément les pixels sombres et les pixels clairs du bruit poivre et sel.

d) Le rapport signal sur bruit est une mesure de la distribution du bruit. Il peut être calculé de la manière suivante:

$$\frac{IntDen(I^2)}{IntDen[(I-M)^2]}$$

où **I** est l'image idéale sans bruit, **M** la mesure de l'image bruitée et **IntDen** la somme de toutes les valeurs d'intensité de l'image. A l'aide de l'enregistreur de macro, créer une macro qui calcule le *SNR (signal noise ratio)* pour **File>Open Samples>Boats**. Créer l'image M mesurée avec **Process>Noise>Add Specified Noise...** avec sigma = 15 . Utilisez la macro pour calculer le SNR après l'application d'un filtre gaussien-flou avec Sigma 1.2, un filtre moyenneur de rayon 3 et un filtre médian de rayon 3. Comparez les résultats. Faites de même avec d'autres images.

### **Exercice 6.5 - Filtre non linéaire - filtre Min et Max –morphologie en Niveaux de Gris**

Si au lieu de la médiane, on prend la valeur minimale ou la valeur maximale on obtient les filtres min ou max. Le filtre min augmente les régions sombres, le filtre max augmente les régions lumineuses. Nous pouvons les utiliser pour définir des opérations morphologiques en niveaux de gris, avec la seule restriction que l'élément structurant est fixé à un carré de rayon r. Certaines des opérations sont les suivantes:

min - éroder - agrandir les zones sombres

max - dilater - agrandir les zones brillantes

ouverture – érosion puis dilatation - supprime les zones brillantes < r

fermeture - érosion puis dilatation - supprime les zones sombres < r

filtre séquentiel – fermeture puis ouverture - supprime les zones brillantes et sombres < r

chapeau haut de forme clair – Image-ouverture(Image) - extrait les zones brillantes

chapeau haut de forme sombre – Image-fermeture(Image) - extrait les zones sombres

a) Appliquer les filtres min et max sur *Lena.tif* essayer les valeurs 1, 10 et 20 pour le rayon.

b) Créer une boîte à outils *simple O* avec un bouton pour chacune des opération :

*erode,dilate, open, close, sequential filter, white top hat and black top hat*

Un clic droit sur le premier bouton devrait ouvrir une boîte de dialogue dans laquelle l'utilisateur peut définir le rayon qui sera utilisé par toutes les opérations. Exécuter les opérations sur le *lena.tif* et *clown.jpg* avec des valeurs différentes de rayon.

Conseils:

- Utilisez l'éditeur de macro pour écrire une nouvelle boîte à outils (**Plugins>New>Macro**).

- Vous pouvez ouvrir une boîte à outils existante, pour regarder un exemple, en tenant la touche MAJ enfoncée et cliquant sur la boîte à outils par le bouton>>.

- Une macro a la forme

```
macro "Erode Action Tool- C000T4b12e" {
...
}
```

La chaîne de C000T4b12e définit l'image sur le bouton. Dans ce cas, c'est la lettre e en noir.

- Déclarez au début une variable pour le rayon des filtres min et max

```
var radius = 1;
```

Pour ajouter une option qui permet de définir le rayon à un bouton, vous pouvez utiliser par exemple:

```
macro 'Erode Action Tool Options' {
radius = getNumber("radius:",radius);
```

}

- Vous pouvez déclarer des fonctions que vous pouvez appeler à partir d'une macro:

```
function erode() {  
run("Minimum...", "radius=" + radius);  
}
```

- Vous pouvez utiliser l'enregistreur de macros, si vous ne savez pas comment écrire une commande spécifique dans le langage de macro.

## Exercice 6.6 - Filtrage dans le domaine fréquentiel

La transformée de Fourier change le système des coordonnées spatiales en coordonnées d'angle et de fréquence. Tout signal peut être approximativement la somme de fonctions harmoniques (sin et cos) à différentes amplitude et phase. Le résultat est dans le domaine des nombres complexes, nous transformons une image spatiale en deux images résultantes: l'image de phase (partie imaginaire) et le *PowerSpectrum* (partie réelle). La partie la plus intéressante est le *PowerSpectrum*.

Selon le théorème de convolution, une multiplication point par point dans le domaine fréquentiel correspond à une convolution dans le domaine spatial. La transformée et la transformée inverse peuvent être calculées dans ImageJ. Les filtres utilisant les transformées de Fourier peuvent être plus efficaces que des filtres de convolution qui n'ont qu'un noyau dans le domaine spatial.

Les applications possibles sont:

- filtre passe-bas - lissage - la suppression du bruit
- filtre passe-haut - la détection de contour
- filtre bande-passe - détecter des structures avec une fréquence donnée
- corrélation, modèle d'appariement, assemblage
- déconvolution (filtrage inverse)

ImageJ permet d'appliquer une transformée de Fourier dans **Process > FFT > FFT**, pour afficher le *PowerSpectrum*. C'est une image spéciale sur laquelle vous pouvez définir un filtre en mettant des régions en blanc ou en noir. Lorsque vous calculez la transformation inverse (**Process > FFT > Inverse FFT**), avec du blanc seules les fréquences sous le masque sont conservées, avec du noir les fréquences sous le masque sont supprimées,

a) Ouvrir *pierre.tif*. Calculer la transformée de Fourier. Voyez-vous les deux taches une dans la moitié supérieure gauche et l'autre en bas à droite ? Dessinez un masque blanc dessus et calculez la transformée inverse. Y a-t-il un objet dans l'image qui correspond aux fréquences conservées?

b) Utiliser la transformée de Fourier pour supprimer le bruit dans l'image *plant-noise.tif*. Quelles fréquences vous avez à supprimer?

c) Utiliser la transformée de Fourier pour détecter les bords de la *ophrys.tif* image. Quelles les fréquences à supprimer?

d) *lines.tif* se compose de lignes blanches verticales qui ont une largeur d'un pixel et une distance de 4 pixels. Comment sera le *PowerSpectrum* de la transformée de Fourier ? Verifier votre réponse en le créant.

e) Charger *text.tif* et *template.tif* . Utilisez la commande de *correlate* dans **Proces>FFT>FD Math** pour calculer la corrélation entre le texte et le modèle.

Transformer le résultat en 8 bits, seuiller pour obtenir juste des points.

Utilisez **Process>Find Maxima** et créer une sélection de points que vous pouvez transférer sur l'image *text.tif* pour contrôler les résultats.