**Traitement d’images pour le microscope**

**Compte rendu de la première semaine**

Notre tâche consiste en l’élaboration d’algorithmes ayant pour but de permettre la calibration d’un microscope à partir d’images. Cette calibration consiste en la connaissance précise du focus, et si possible des déplacements latéraux.

Pour répondre à ce problème nous avons choisi de coder sous Matlab, et avons passé la première semaine à nous créer des outils qui permettront d’effectuer un traitement d’image rapide.

***Ouverture d’images binaires :***

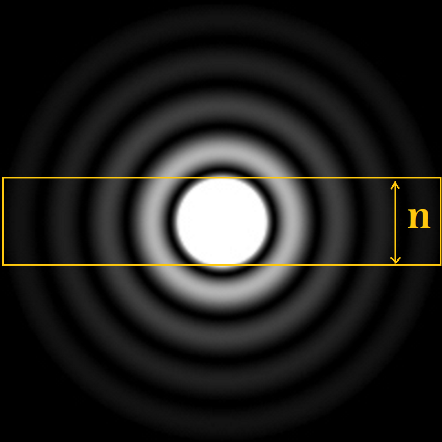
Fonction **bin2mat**

Syntaxe : *A=bin2mat('Calibration\STACK=0001\_IM=00001\_Z=001375.2Ddbl');*

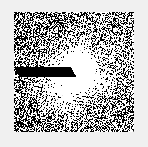
Cette fonction nous permet d’ouvrir rapidement les stacks de mesures qui sont sous forme binaire.

***Mesures de moyennes sur une partie d’image :***

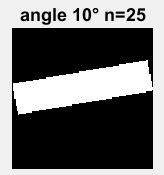
**moyhor** et **moyvert** renvoient les moyennes horizontales et verticales d’une image en partant du centre, et en prenant un écart au centre de n.

*Rôle de moyvert*

**moydegv**, **moydegv2** et **moydegv3**  suivent le même processus que moyvert : ils parcourent l’image horizontalement mais calcule des moyennes sur des segments (longueur n) orientés d’un angle d>0 par rapport à la verticale. moydegv2 effectue une pondération 2 pixels autour de la droite et est donc plus précise, et moydegv3 une pondération sur 4 pixels.

*Parcours d’une image par moydegv, angle 30°*

La fonction **moydeg**, de manière semblable aux fonctions précédentes, renvoie la moyenne sur des segments de taille n faisant un angle de d degré par rapport à la verticale (dans le sens trigonométrique), d étant rentré en paramètre. On peut de plus préciser le nombre d’échantillon pris dans une moyenne (le pas x rentré en paramètre).



***Obtention rapide des filepaths des mesures :***

Les filepaths des données sont tous basés sur le même modèle :

‘*\Calibration\STACK=0001\_Im=00001\_Z=000700.2Ddbl’* ou ‘*\Mesures\STACK=0000\_Im=00199\_Z=000700.2Ddbl’.*

La fonction **name\_Z** nous permet de ne plus avoir à rentrer le filepath en entier, uniquement les facteurs qui changent, le nom de dossier, le numéro de stack ou la valeur de Z par exemple.

***Affichage de l’ensemble des mesures :***

La fonction **myui** a pour rôle d’afficher simultanément toutes les images et graphes d’un ensemble de mesures en créant un slider qui fait varier la profondeur Z et charge les images en conséquence. Elle affiche l'image avec des traits correspondant à l'inclinaison des franges, et utilise les fonctions bin2mat, moyvert et moyhor.

Ces différentes fonctions vont nous servir d’outils pour afficher et exploiter nos différents résultats.

Nous avons par la suite cherché à appliquer un filtre de Wiener aux mesures ; pour cela, il nous a fallu d’abord appliquer un masque hyper gaussien à notre image, puis filtrer l’image résultante à l’aide d’un passe-bas ; ensuite nous avons traité la transformée de Fourier de l’image pour en éliminer la phase. Ces différentes nous assurent que l’image et la calibration qui nous sert de référence sont bien en phase, et qu’il n’y a pas trop de bruit auxiliaire. Au cours de ces différentes étapes, nous avons utilisé les fonctions suivantes :

***Détermination des coefficients de la gaussienne correspondant à notre image :***

La fonction **fit\_gauss** renvoie les coefficients correspondant à l'approximation d'une gaussienne 1D avec offset de la forme y=A+Bexp((x-D)^2)/(2s\*s). On applique cette fonction sur l’horizontale et la verticale de notre image afin d’obtenir les coordonnées du centre de l’image, ainsi que la variance et l’offset de la gaussienne correspondant à l’image. On peut alors se débarrasser de l’offset puisqu’on connait sa valeur.

***Création d’une hyper gaussienne :***

La fonction **hgaussp** fournit la matrice d’une hyper gaussienne 2D en prenant en entrée les caractéristiques de l’hyper gaussienne (coordonnées du centre, écart-type, degré de la gaussienne…). Cette fonction peut aussi donner la matrice d’une gaussienne, en choisissant comme degré 1.

***Application du masque hyper gaussien :***

Connaissant les caractéristiques de l’image, on peut appliquer la fonction **masque\_hg** qui multiplie notre image à la matrice d’une hyper gaussienne obtenue à l’aide de la fonction **hgaussp** : on obtient ainsi notre image masquée par une hyper gaussienne.

***Application du filtre passe-bas :***

La fonction **passebas\_hg2D** applique à notre image, rentrée en paramètre, un filtre passe-bas basé sur une hyper-gaussienne dont les caractéristiques sont aussi rentrées en paramètre.

***Lissage de l’image en gaussienne :*** on lisse l’image en gaussienne 2D une fois que l’on a appliqué le filtre passe-bas.

* ***Détermination de la jacobienne :*** La fonction **jaco\_der** calcule les dérivées par rapport à chaque variable de : f(x,y) =offset + A\*exp( -( ( x-mu1).^2+(y-mu2).^2 ) /(2\*s^2) ), où offset, A, mu1, mu2 et s sont des variables d’entrée de la fonction, et en déduit la jacobienne.
* ***Fonction de la gaussienne :*** La fonction **Rgauss2D** calcule la gaussienne en deux dimensions en utilisant des vecteurs 1D, elle est conçue pour être utilisée avec lsqnonlin, elle fait appel à **jaco\_der.**
* ***Détermination des coefficients de la gaussienne 2D correspondant à l’image :*** La fonction **approxgauss2D** renvoie, de manière similaire à la fonction fit\_gauss, les caractéristiques de la gaussienne 2D. Elle fait appel à **Rgauss2D** (pour l’utilisation de lsqnonlin).
* ***Création de l’image approximée en gaussienne :*** La fonction **fit\_gauss2D** approxime notre image en gaussienne 2D. Elle fait appel à **approxgauss2D**.

***Traitement de la transformée de Fourier de l’image :***

La fonction **traitement\_fft\_gaussienne** calcule la transformée de Fourier de notre image, puis en annule la phase, avant de ré-exprimer la matrice ainsi obtenue dans le domaine temporel. Ce processus permet de centrer notre image.

***Filtrage par filtre de Wiener :***

* ***Détermination du paramètre du filtre :*** La fonction **minEQM** détermine la valeur optimale du paramètre dans l’expression : .
* ***Application du filtre de Wiener :*** La fonction FiltreWiener applique le filtre de Wiener associé à une perturbation rentrée en paramètre à notre image, elle aussi rentrée en paramètre.

Image de départ

Image de départ

+

(x0,y0), variance, offset

Image masquée par le masque hyper gaussien

Image approximée avec phase annulée

Image approximée par une gaussienne

Image masquée sans haute fréquence

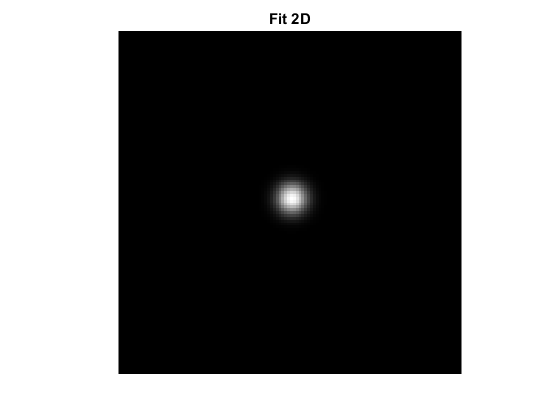
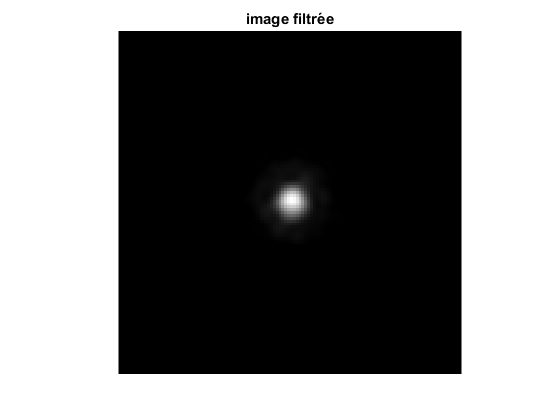
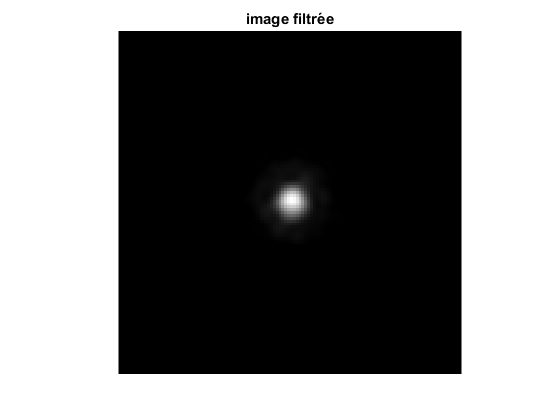
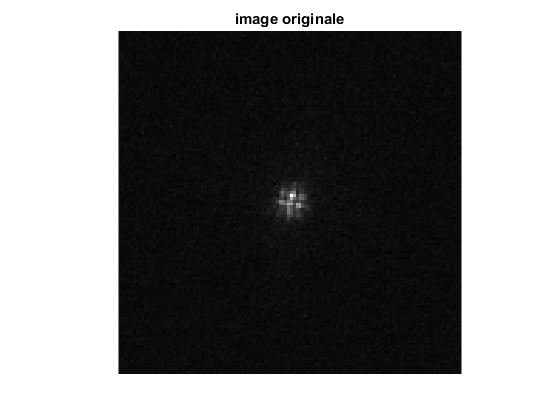


Image filtrée par filtre de Wiener

**fit\_gauss**

**FiltreWiener**

**Traitement\_fft\_gaussienne**

**fit\_gauss2D**

**Passebas\_hg2D**

**masque\_hg**

**Schéma du filtrage de l’image**