**Traitement d’image TP N° 1 : filtrage linéaire**

**A/ Problème**

On désire comparer les performances de deux filtres linéaires : le filtre de la moyenne et le filtre gaussien. L’image utilisée pour les différents tests est une image bruitée synthétique dont les paramètres (variance du bruit ...) sont totalement maîtrisés.

Les performances sont évaluées à l’aide du critère des moindres carrés suivant :

2=

*NM 1*

*∑ N*

*∑ M*

i=1

j=1

*I i,j−I*

*f*

*i, j2*

(1)

où I et I

*f*

sont respectivement les images non bruitée et bruitée puis filtrée.

1/ A quelle grandeur correspond le paramètre s2 ?

**B/ Manipulation**

1/ Chargement et affichage de l'image non bruitée. A partir de la fenêtre console de Matlab, charger l'image stockée sur le DD :

**>>im = imread('photophore.tif') ;**

Cette commande permet de créer la variable im. Pour l’afficher deux commandes peuvent être utilisées, la commande imshow et la commande image. La commande imshow s’utilise simplement :

**>>imshow(im)**

Cette commande tient compte du type de l’image afficher pour ajuster la dynamique. Une image de type UINT8 (entier) sera afficher avec une dynamique entre 0 et 255, une image de type DOUBLE ou FLOAT sera affichée avec une dynamique entre 0 et 1.

La commande image n’ajuste pas la dynamique, il est donc nécessaire de celle de l'image à afficher de telle sorte qu’elle soit entre 0 et 64. D'autre part il est également nécessaire de préciser quelle la table des couleurs utilisée (colormap).

Nous avons donc dans le cas de l’affichage d’une image de type UINT8 avec cette fonction les instructions suivantes :

**>>image(im/4) >>colormap(gray)**

D’autre part, afin de pouvoir manipuler numériquement cette image, il est nécessaire dans un premier temps de la convertir en double :

**>>imf = double(im) ;**

2/ Réalisation de l’image synthétique.

L’image imf constitue la référence non bruitée à partir de laquelle nous allons construire l’image imb en ajoutant à cette image un bruit gaussien d'écart type et = 5.

**>>et = 5 ; >>imb = imf + et\*randn(size(imf)) ;**

Visualiser les deux images et constater les effets du bruit pour différentes valeurs de et.

n.b. : Pour afficher les deux images simultanément, il est possible de faire apparaître une nouvelle fenêtre dans laquelle sera affichée la nouvelle image à l'aide de la commande figure :

*(affichage première image) figure (affichage deuxième image)*

3/ Filtre moyen

On se propose de comparer deux filtres de la moyenne de tailles respectives 3x3 et 5x5. Réaliser les deux masques des filtres. S’assurer que le filtre sont bien normalisés (de poids 1) de telle sorte que le niveau moyen de l’image ne soit pas affecté.

**>>flt1 = [1 1 1 ;1 1 1 ;1 1 1]/9 ; >>flt2 = [1 1 1 1 1 ;1 1 1 1 1 ;... ;1 1 1 1 1]/25 ;**

En utilisant l’opérateur de convolution, réaliser le filtrage par les deux filtres ainsi créés :

**>>imf1 = conv2(imb,flt1,’same’) ; >>imf2 = conv2(imb,flt2,’same’) ;**

Visualiser les différentes images filtrées. Que constate-t-on ?

4/ Filtre gaussien

Sur le même principe on réalise le filtrage à l’aide de deux filtres gaussiens de taille 3x3 et 5x5. Rappel : L'expression de la fonction de Gauss (non normalisée) est donnée par la relation :

f x,y=exp

 −

x2y2 2 2



Pour réaliser un filtre Gaussien de taille nxn il est nécessaire de procéder comme suit :

Construire à l'aide de l'instruction meshgrid deux masques X et Y qui contiennent respectivement les coordonnées x et y du pixel considéré.

Exemple : l'instruction

**>>[X,Y] = meshgrid(-1:1,-1:1) ;**

construit les masques X et Y qui contiennent respectivement les coordonnées x et y d'un filtre 3x3 :

**X = [ -1 0 1 Y = [-1 -1 -1**

**-1 0 1 0 0 0 -1 0 1] 1 1 1]**

A partir de ces masques on construit le filtre normalisé :

**>>fgauss = exp(-(X.^2 + Y.^2)/(2\*sigma\*sigma) ; >>fgauss = fgauss/sum(sum(fgauss)) ;**

Réaliser les 2 filtres (normalisés) en s’assurant que la taille n du filtre soit impaire et soit égal à 6 s. Filtrer l’image test, visualiser les résultats.

5/ Comparaison des performances.

Calculer les performances des filtres en utilisant l'équation (1). Classer les 4 filtres en fonction de leur performance respective.