

## TP : Convolution Spatiale

### Filtrage Passe Bas : Réduction de Bruit additif

### Filtrage Passe Haut : Rehaussement de contraste

#### **Remarque importante :**

Créer sous Windows un répertoire de travail TP3 au même niveau que le répertoire des images et le répertoire des utilitaires. Se positionner dans ce répertoire de travail dans l'environnement Matlab (taper la commande « pwd » pour vérification).

#### **Travail préliminaire :**

1°) On considère un noyau unidimensionnel continu représenté par la fonction  $h(x)$  ci-dessous. On pourra considérer pour fixer les idées que cela correspond à un filtre continu 1D que l'on applique sur une image ligne après ligne.

$$h(x) = \exp\left\{-\frac{x^2}{2\sigma^2}\right\}.$$

La meilleure approximation au sens des moindres carrés des filtres gaussiens analogiques est donnée par les filtres numériques binomiaux dont les coefficients du noyau de convolution

sont les coefficients du polynôme d'ordre  $n$  :  $(x + y)^n = \sum_{k=0}^n b_{k,n} \cdot x^{n-k} \cdot y^k$ . C'est le triangle de

Pascal :

Ordre n=0	1
Ordre n=1	1 1
Ordre n=2	1 2 1
Ordre n=3	1 3 3 1
Ordre n=4	1 4 6 4 1
etc.	etc.

1°) Calculer la somme en ligne, notée  $s_n$ , des éléments du triangle. Que remarque-t-on ?

2°) On note  $h_n[k]$ , le noyau de convolution du filtre d'ordre  $n$ . Appliquer manuellement l'algorithme de convolution et montrer que l'on a :  $h_n[k] = (h_1)^{*n}[k]$ . Vérifier ensuite ces résultats en utilisant l'instruction « conv2 » (help conv2) de Matlab. Ne pas effectuer de programme spécifique pour cela, entrer directement les instructions dans la fenêtre de commande de Matlab

3°) On interprète maintenant  $p_n[k] = \frac{h_n[k]}{\sum_k h_n[k]}$  comme la probabilité pour une variable

aléatoire  $K_n$  de prendre la valeur  $k$ . Les valeurs de  $k$  possibles sont telles que la pointe du

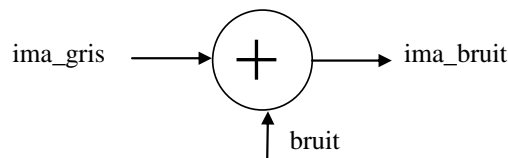
triangle est à la position 0, les valeurs pour n pair sont entières et les valeurs pour n impair sont multiples de 1/2. Voir le tableau ci-dessous. Calculer pour chaque variable aléatoire, la moyenne et la variance. On fera les calculs pour 5 variables aléatoires, n=0...4. Que remarque-t-on ?

n=0	Valeur			0		
	probabilité			1		
n=1	Valeur		-1/2	1/2		
	probabilité		1/2	1/2		
n=2	Valeur		-1	0	1	
	probabilité		1/4	2/4	1/4	
n=3	Valeur	-3/2	-1/2	1/2	3/2	
	probabilité	1/8	3/8	3/8	1/8	
n=4	Valeur	-2	-1	0	1	2
	probabilité	1/16	4/16	6/16	4/16	1/16

### Fichier TP\_reduction\_bruit.m

Laisser vous guider par le fichier. L'objectif est de former des noyaux de convolution de filtres de moyenne et de filtres gaussiens (binomiaux) et comprendre leur impact sur des images bruitées suivant différents niveaux de bruit et différentes tailles de noyau.

1°) La première partie du programme consiste à créer une image de bruit que l'on ajoutera à l'image originale en niveaux de gris.



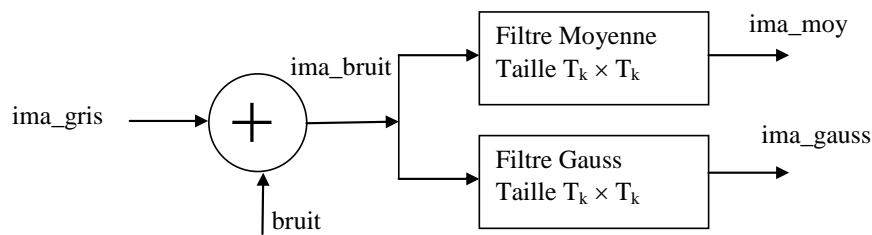
Le bruit sera modélisé par une distribution gaussienne de moyenne nulle et de variance  $\sigma_b^2$  que l'on fixera. En théorie du signal, on assimile la variance à la puissance d'un signal. Le rapport « Signal à Bruit » est ainsi défini :

$$RSB = 10 \cdot \log_{10} \left( \frac{P_{Signal}}{P_{Bruit}} \right) = 10 \cdot \log_{10} \left( \frac{Var(Signal)}{Var(Bruit)} \right).$$

Calculer la variance des niveaux de gris de l'image originale et faire varier la variance du bruit pour explorer les RSB entre 10 et 20dB. Pour cela, utiliser les instructions « randn » et « std » (help randn, help std).

Comment évolue l'aspect de l'image bruitée quand la valeur de RSB diminue ?

2°) Dans la deuxième partie du programme, on va filtrer l'image bruitée par un filtre de moyenne et par un filtre de Gauss (binomial) en faisant varier la taille du noyau de convolution.



Dans un premier temps, il vous faudra générer correctement les noyaux de convolution, puis ensuite effectuer la convolution. Faire en sorte que l'image convoluée ait la même taille que l'image d'entrée. Les instructions pour les différents affichages sont déjà écrites dans le programme, mais libre à vous de les modifier si nécessaire. Vous utiliserez les résultats du travail préliminaire, pour générer automatiquement et récursivement les noyaux gaussiens (binomiaux) aux tailles mentionnées à savoir  $3 \times 3$ ,  $5 \times 5$  et  $7 \times 7$ .

3°) Analyser les résultats obtenus en comparant d'une part, les filtres « moyenne » et « gaussien » pour une taille donnée et d'autre part, pour chaque famille de filtre, les différentes tailles. Quelles sont les limites de ce traitement ?

### ***Fichier TP\_rehaussement.m***

Laisser vous guider par le fichier. L'objectif est de former des noyaux de convolution de filtres de rehaussement à partir de filtres Passe Bas. La difficulté réside à afficher correctement les images car les filtre Passe Haut modifient largement la dynamique des images de sortie dont la valeur des pixels peut être positive ou négative.

1°) Implémenter un premier filtre de rehaussement en prenant comme filtre Passe Bas de départ, le filtre binomial  $3 \times 3$ . Observer les résultats obtenus et proposer des solutions pour résoudre les problèmes d'affichage.

2°) Expérimenter plusieurs filtres en changeant le filtre binomial en filtre de moyenne et en faisant évoluer le paramètre  $\alpha$  de construction du filtre rehausseur. Vous pouvez également faire varier la taille du noyau de convolution. Interpréter les résultats obtenus.