

	Section	2ème année Génie des Télécommunications	Année Universitaire :
	Matière	TP Traitement et Codage Image	2023 - 2024

## TP N°2 : Filtrage d'images

### Introduction:

Nous entendons par filtrage la réduction et l'élimination des distorsions introduites (bruit) par le système ayant servi à acquérir l'image. Ceci est afin d'obtenir une image qui soit la plus proche possible de l'image idéale qui aurait été obtenue si le système d'acquisition était parfait.

Durant ce Tp, vous allez survoler certains filtres d'images allant des filtres linéaires (passe-bas et passe-haut) jusqu'aux filtres non linéaires tels que le filtre médian. L'environnement Jupyter Notebook riche en fonctions, nous a permis de visualiser les images filtrées, observer les différences par rapport à l'image d'origine et les interpréter.

### Partie I : filtrage passe-bas

#### 1. Bruitage d'une image

- Appliquer un bruit de type «poivre-et-sel» («salt-and-pepper») sur l'image de cameraman en utilisant la commande **random\_noise** avec un pourcentage des pixels remplacés égale 10%.
- Appliquer un bruit de type «gaussien» sur l'image de cameraman en utilisant la commande `random_noise`.
- Afficher l'image bruitée et dire comment peut-on créer ce bruit?

#### 2. Filtrage moyennneur

##### a. Principe :

Le filtre moyennneur est un filtre passe-bas. En l'appliquant sur l'image, celle-ci devient floue et plus particulièrement les contours. Ce filtre sert à éliminer les dégradations locales de faibles dimensions. Son principe est de remplacer chaque pixel par la moyenne des valeurs des pixels adjacentes et du pixel central. Il pourrait y avoir plusieurs tailles de masque 3×3, 5×5 et 7×7 qui définissent des filtres moyennneurs différents.

##### b. Questions:

- Utiliser la fonction `ndimage.convolve` pour effectuer un filtrage moyennneur de l'image cameraman bruitée.
- Utiliser respectivement des noyaux de dimension 3x3, 5x5 et 7x7
- Comparer les résultats et interpréter l'effet du filtre moyennneur.

### 3. Filtrage médian

#### a. Principe:

Le filtre médian est un filtre non linéaire. Il permet d'atténuer certains types de bruits en dégradant très peu les contours. Il est efficace pour éliminer les bruits qui affectent seulement un petit nombre de pixels. Son principe est d'affecter à un pixel la valeur médiane des intensités dans son voisinage.

#### b. Questions:

Utiliser la fonction `ndimage.median_filter` pour effectuer ce filtrage.

- Appliquer un filtre médian de 3x3 et de 7x7.
- Comparer et interpréter l'effet du filtre médian
- Comparer les résultats du filtre moyenneur et du filtre médian.

### 4. Filtrage gaussien

#### a. Principe:

Le filtre gaussien est un filtre passe-bas dont la réponse impulsionnelle est une fonction gaussienne. Mathématiquement, ce filtre modifie le signal entrant par une convolution avec une fonction gaussienne. Comme pour le filtre moyenneur, nous distinguons entre le filtre gaussien 3 x 3 et le filtre gaussien 5 x 5...

#### b. Questions:

Le filtrage gaussien peut être effectué en utilisant la commande `filters.gaussian`

- Appliquer un filtre gaussien à l'image cameraman bruitée pour **sigma = 1**, **sigma = 3** et **sigma = 5**.
- Comparer et interpréter l'effet du filtre gaussien.

## Partie II : Filtrage passe-haut

#### a. Principe:

Un filtre passe-haut réalise l'opération complémentaire de celle obtenue avec les filtres passe-bas.

Il permet alors d'atténuer les basses fréquences et faire passer les hautes fréquences. Son utilisation principale est la détection de contours.

## b. Filtrage par convolution

Filtrer l'image cameraman par un filtre passe-haut ayant comme noyau:

$$F_{PH} = \frac{1}{9} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

- Appelons le résultat *IPH*
- Filtrer la même image par un filtre passe-bas (moyenne 3x3). Appelons le résultat *IPB*
- Comparer les images *IPH* et (*originale* - *IPB*)
- Pour  $A = 1.1, 1.2$  et  $1.5$ , comparer *IPH* avec l'image *IHB* calculée de la manière suivante:

$$IHB = (A - 1)(originale) + IPH$$

## c. Filtres passe-haut prédéfinis

### Détecteurs de Sobel, Prewitt et de Canny

Cette partie est consacrée à la détection de caractéristiques pour la métrologie, en appliquant les méthodes vues en cours à l'image **L.png**.

Sous le module **skimage.filters**, ils existent certains types de filtres prédéfinis, comme **sobel**, **prewitt**.

1. Tester les filtres mentionnés ci-dessus afin de détecter les contours de l'image L.png.
2. Appliquez un seuil sur l'image obtenue pour effectuer une première détection de contours. Comment évolue cette détection lorsque le seuil varie ? (tester les trois valeurs de seuils suivants {0.02, 0.1, 0.4}).

**Remarque :** la méthode de seuillage se fait par la commande suivante :

**z=img>seuil\*img.max()**

3. Trouvez le seuil optimal, c'est-à-dire celui qui permettra d'obtenir le plus de contours de l'objet tout en conservant des contours les plus précis possibles
4. Appliquer le filtre **canny** prédéfini sous le module **skimage.feature** sur l'image étudiée en testant des valeurs de sigma égale à {0.1, 1 et 5}. Commenter les résultats.
5. Comparer ces résultats avec ceux obtenus dans la question précédente.