

Traitement d'images

TP N°3 : Filtrage spatial

Manipulation 1 : Réduction du bruit

- I. Lire et afficher l'image « isabelle.bmp ».
- II. Créer un bruit gaussien de moyenne nulle et d'écart-type $\sigma = 0.1$ à l'aide de la fonction « `np.random.normal` », le rajouter à l'image et afficher l'image bruitée.
 - A. Filtre moyennneur
 1. Réaliser un filtrage moyen 3x3 à l'aide des fonctions « `cv.blur` ». Afficher le résultat et commenter. Quelle est la taille de l'image obtenue ? Comment les bords ont été traités ? Changer de méthode et commenter.
 2. Réaliser la soustraction entre l'image originale et l'image filtrée. Afficher le résultat et commenter.
 3. Refaire les étapes 2-4 pour un masque 5x5, 6x6, 7x7. Conclure
 - B. Filtre gaussien
 1. Réaliser un filtrage gaussien (3x3 ; $\sigma = 1$) à l'aide de la fonction « `cv.GaussianBlur` ». Afficher le résultat et commenter.
 2. Etudier l'influence de la taille du filtre et de la valeur de l'écart-type.
 3. Sélectionner le masque qui vous semble le meilleur. Réaliser une soustraction entre l'image originale et l'image filtrée. Commenter
 - C. Filtre médian
 1. Réaliser un filtrage médian à l'aide de la fonction « `cv.medianBlur` » en considérant plusieurs voisinages 3x3, 5x5, 7x7 et 9x9. Comparer et conclure.
 2. Sélectionner la taille du filtre qui vous semble la meilleure. Réaliser une soustraction entre l'image originale et l'image filtrée. Commenter
- III. Comparer les 3 filtres. Quel est le filtre le plus adapté au bruit gaussien ?
- IV. Refaire l'étude des 3 filtres pour un bruit impulsionnel de 0.1 à l'aide de la fonction « `impulse` » (voir page 2). Quel est le filtre le plus adapté au bruit impulsionnel ?
- V. Reprendre le filtre médian. Augmenter le pourcentage de pixels remplacés. Le filtre médian est-il toujours performant ? Si non, à partir de quel pourcentage il ne l'est plus dans le cas du bruit impulsionnel.

Manipulation 2 : Détection de contours

1. Lire et afficher l'image « lena.png ».
2. Transformer l'image en niveaux de gris.
3. Pour chacun des opérateurs « classique, Prewitt et Sobel », calculer et afficher :
 - a. le gradient selon les lignes,
 - b. le gradient selon les colonnes,
 - c. le module du gradient,
 - d. la phase du gradient, que représente-t-elle ?
 - e. Comparer les 3 opérateurs
(Utiliser la fonction « `cv.filter2D` » pour le produit de convolution)

Traitement d'images

4. Reprendre l'image originale, calculer et afficher le Laplacien 4-connexités et 8-connexités. Comparer
5. Afin d'améliorer le résultat obtenu, appliquer un filtrage gaussien avant de calculer le Laplacien 8-connexité. Afficher le résultat.
6. Comparer entre les différents filtres appliqués pour la détection de contours.

```
def impulse(img, amount):  
    row, col = img.shape  
    s_vs_p = 0.5  
    out = np.copy(img)  
    # Salt mode  
    num_salt = np.ceil(amount * img.size * s_vs_p)  
    coords = [np.random.randint(0, i - 1, int(num_salt))  
               for i in img.shape]  
    out[coords] = 255  
  
    # Pepper mode  
    num_pepper = np.ceil(amount * img.size * (1. - s_vs_p))  
    coords = [np.random.randint(0, i - 1, int(num_pepper))  
               for i in img.shape]  
    out[coords] = 0  
    return out
```