IMA201 - TP02
Élève : OLIVEIRA MACHADO DE SOUSA, Lucas

Questions

2. Transformation géométrique

- Utiliser la fonction (rotation) pour transformer une image de votre choix. Quelle différence y a-t-il entre la méthode à plus proche voisin et la méthode bilinéaire ?

On voit qu'avec la méthode PPV l'image perd un peu de qualité ; on peut noter des gros pixels sur les bords. Avec la méthode bilinéaire, l'image est un peu plus floue.

- Que constatez-vous sur une image qui aurait subi huit rotations de 45 degrés (en bilinéaire et en plus proche voisin) ?

L'interpolation bilinéaire préserve mieux les caractéristiques de l'image avec un peu plus de flou, pendant que l'interpolation par la méthode PPV augmente les morceaux vraiment constants dans l'image.

- Que constatez-vous si vous appliquez la rotation avec un facteur de zoom inférieur à 1 (par exemple 1/2) ? Qu'aurait-il fallu faire pour atténuer l'effet constaté ?

On voit que l'image perd beaucoup de qualité, on peut percevoir des gros pixels en spécial sur les bords. Pour améliorer ça, il faut faire une interpolation avec filtrage (appliquer un filtre passe-bas avec l'interpolation).

3. Filtrage linéaire et médian

- Expliquer le rapport entre la taille du noyau (size) renvoyé par *get_gau_ker* et le paramètre de cette commande.

Le vecteur retourne par *get_gau_ker* est une matrice MxM, où M est environ 5 fois le paramètre de la fonction (ça dépend, mais en général c'est proche de cette valeur). Il y a quelques opérations d'arrondi, alors la valeur peut être un peu différent (M est 17 pour le paramètre 3, par exemple).

- Après avoir ajouté du bruit à une image simple telle que pyramide.tif ou carre orig.tif et avoir filtré le résultat avec des filtres linéaires, expliquez comment on peut évaluer (sur des images aussi simples) la quantité de bruit résiduel (la commande var image donne la variance d'une partie d'une image).

Pour une image simple comme celui-là, on peut choisir une région où il n'y a pas de variance de couleur (un morceau constant) et on peut utiliser la fonction *var_image* pour calculer la variance dans cette partie de l'image. Comme la partie choisie est constant, on n'espère pas de variance pour l'image originale, mais on l'espère pour l'image bruitée.

- Appliquer un filtrage médian à une image bruitée et comparer le résultat avec un filtrage linéaire.

Qualitativement, on peut percevoir qu'il y a un changement de contraste quand on utilise le filtre médian. Les nuances de gris sont plus claires dans l'image filtrée avec le filtre médian qu'avec le filtre linéaire de noyau gaussien. En utilisant la méthode proposée par la dernière question, on peut calculer le bruit résiduel pour les deux cas.

En choisissant une partie grise à gauche de l'image, j'ai obtenu une variance de :
0 pour l'image originale ;
95.16 pour l'image bruitée ;
1.67 pour l'image filtrée avec le filtre linéaire ;
2.18 pour l'image filtrée avec le filtre médian.

La variance est proche entre les deux filtres, mais on peut clairement percevoir la différence entre eux.

- Faites une comparaison linéaire/médian sur l'image pyra-impulse.tif. Que constatez-vous ?

Pour ce cas, on voit que l'image filtrée par le filtre médian ressemble mieux l'image originale. Le filtre médian envoie "la valeur telle qu'il y ait autant de pixels plus brillants que de pixels plus sombres que cette valeur dans le voisinage", alors il est mieux adapté au filtrage du bruit impulsionnel.

- Expliquer la différence de comportement entre filtrage linéaire et médian sur le point lumineux situé en haut à droite de l'image carre orig.tif.

Avec le filtre linéaire on ne perd pas l'information du point lumineux, au contraire du cas avec le filtre médian.

4. Restauration

- Appliquer un filtre linéaire à une image puis utilisez la fonction filtre inverse. Que constatez-vous ? Que se passe-t-il si vous ajoutez très peu de bruit à l'image floutée avant de la restaurer par la commande précédente ?

On constate qu'on peut récupérer l'image originale facilement avec cette méthode s'il n'y a pas de bruit, mais si on ajoute du bruit (même un petit bruit de valeur 0.1) ça ne marche plus.

- Comment pouvez-vous déterminer le noyau de convolution qu'a subi l'image carre flou.tif ?

  On peut voir que le noyau utilisé est la région 3x3 où était le point lumineux en miroir, puisque le point blanc est une impulsion, alors si on convolue cette impulsion avec un noyau défini on va obtenir le propre noyau en miroir.

- Après avoir ajouté du bruit à cette image utilisez la fonction Wiener pour restaurer cette image. Faites varier le paramètre λ et commentez les résultats.

Avec le filtre de Wiener on peut percevoir que c'est difficile de retrouver l'image originale parfaite. Le paramètre lambda est proportionnel au bruit. Si on sous-estime le bruit, on récupère une image avec des dégâts de texture. Si on surestime le bruit, on récupère une image très flou. Alors, il faut rencontrer le point optimal entre ces deux cas.

5. Applications

5.1. Comparaison filtrage linéaire et médian

- Pour une image simple telle que carré_orig.tif et un bruit d'écart-type 5, trouver la taille du noyau constant qui réduit le bruit dans les mêmes proportions qu'un filtre médian circulaire de rayon 4. Explicitez l'algorithme utilisé.

On peut mesurer le bruit en sélectionnant une région et en mesurant le bruit dans cette région. Donc, on peut tester par force brute (avec des noyaux constants de tailles différentes) quel est la meilleure taille pour ça.

```
im=skio.imread('images/carre_flou.tif')
imbr=noise(im, 5)
viewimage(imbr)

imFMC = median_filter(imbr, r=4)
viewimage(imFMC)
bruitMedian = var_image(imFMC, 10, 10, 90, 90)
bruitMedian


differenceMin = pow(10,9)
for i in range(100):
    imFC = filtre_lineaire(imbr, get_cst_ker(i+1))
    bruitConstant = var_image(imFC, 10, 10, 90, 90)
    difference = abs(bruitConstant - bruitMedian)
    if difference < differenceMin:
        differenceMin = difference
        bruitPlusProche = bruitConstant
        taille = i+1
```

J'ai trouvé la taille 7x7.