IMA201-TP02

Leying ZHANG

July 31, 2022

1 Transformation géométrique

1. 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 ?

   En comparant la Figure 1 et la Figure 2, on trouve que la Figure 2 est plus claire, avec une lumière légèrement plus vive.

Figure 1: Rotation 45° de méthode à plus proche voisin

Figure 2: Rotation 45° de méthode bilinéaire

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

   En comparant la Figure 3 et la Figure 4, on trouve que la différence avant devient plus grande: L'image tournée par la méthode bilinéaire (la Figure 4) est plus nette et plus lumineuse.

   De plus, quand on fait la rotation, il ne faut pas presenter image complet, car notre image tourné va devient plus grande après 8 fois.

Figure 3: Rotation 45° 8 fois de méthode à plus proche voisin

Figure 4: Rotation 45° 8 fois de méthode bilinéaire

3. 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 ?

   En comparant la Figure 5 et la Figure 6, on trouve que la différence entre les deux images est encore plus grande: L'image tournée par la méthode bilinéaire (la Figure 6) est plus nette, surtout sur les bords, par exemple au bord du chapeau. En outre, lorsque on compare la Figure 6 à un agrandissement de 400 % avec la Figure 2 à un agrandissement de 200 %, on constate que la Figure 6 est moins nette en raison du repliement à cause de la facteur de zoom inférieur à 1.

Figure 5: Rotation 45° avec zoom 0.5 de méthode à plus proche voisin

Figure 6: Rotation 45° avec zoom 0.5 de méthode bilinéaire

2 Filtrage linéaire et médian

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

   Le paramètre de cette commande est s, et la taille de le noyau gaussien est

   $$\text{taille de noyau} = \text{int}(\max(3, 2 \times \text{np.round}(2.5 \times s) + 1))$$

   Donc, si s = 0, alors taille de noyau est 3×3 (comme le noyau est un carré 2D). Si s = 1, alors, la taille est 5×5, si $s \geq 2$, la taille devient $(2\text{np.round}(2.5 \times s) + 1) \times (2\text{np.round}(2.5 \times s) + 1)$. Presque tous les noyaux de convolution ont une taille impaire, donc on peut trouve une pixel au milieu.

2. 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 evaluer (sur des images aussi simples) la quantité de bruit résiduel (la commande var image donne la variance d'une partie d'une image).

   On ajout des bruit (br = 30) dans l'image pyramid.tif. Pour savoir la variance, on choisit une petit region (pixel de taille 5×5, l'axe x est de 50 à 55, et l'axe x est de 50 à 55). Cette region doit être tout noir si on n'ajout pas de bruit. Donc, après les filtres différents, on a les résultats dans la Table 1:

Table 1: Variance des images après filtres

Filtre	**Variance**
Image Originale	0
Image avec bruit	576.87
Filtre constante	28.58
Filtre gaussien	61.61
Filtre median 1	34.41
Filtre median 2	64.60
Filtre median 3	279.97

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

   On applique les filtres en pyramide.tif après avoir ajouté des bruit dans cet image. On trouve que les filtres linéaires est plus efficace, comme la Figure 9 et la Figure 10 sont plus similaire en comparant avec l'image originale dans la Figure 7. Ceci est également cohérent avec ce que on a obtenu dans la question précédente, où la variance obtenue après le filtrage linéaire est plus petite. C'est probablement parce que nous ajoutons du bruit gaussien et que le filtre médian n'est pas aussi performant.

Figure 7: pyramide.tif

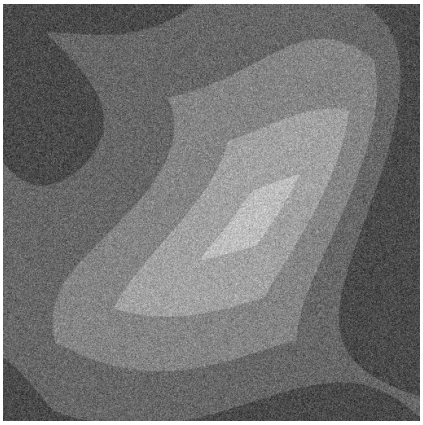

Figure 8: pyramide.tif avec br=30

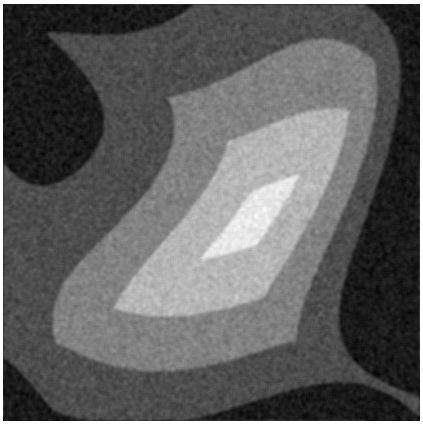

Figure 9: Après le filtre constante

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

   Dans ce cas, on trouve que les filtre medians sont plus performant que les filtre linéaire en comparant avec la Figure 14, cela montre que le filtre médian est plus efficace pour le bruit impulsif que pour le bruit gaussien.

5. 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

   En comparant avec l'image originale dans la Figure 20, on trouve que Le filtre linéaire n'est pas assez bon pour les bords, car les bords de ce carré noir deviennent flous. Les bords du filtre médian, en revanche, sont toujours nets, même s'il manque quelques pixels dans les quatre coins.

   En outre, sur l'image originale, il y a un point lumineux en haut à droite, mais il n'est pas visible dans l'image après le filtre constante (Figure 21) et les filtres medians (Figure 23,24,25), alors que ce point lumineux est encore faiblement visible dans l'image du filtre gaussien 22.

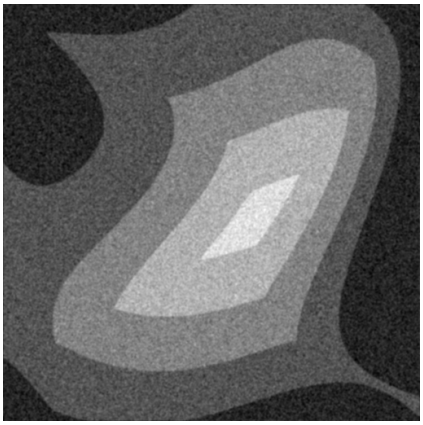

Figure 10: Après le filtre gausienne

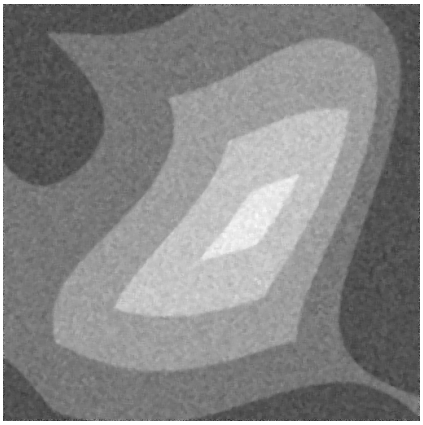

Figure 11: Après le filtre median de type 1

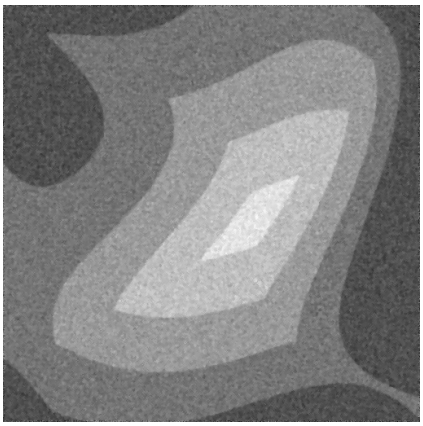

Figure 12: Après le filtre median de type 2

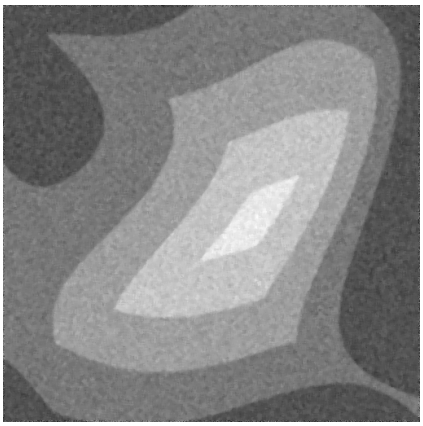

Figure 13: Après le filtre median de type 3

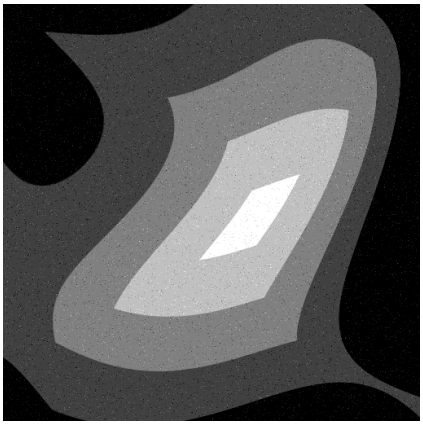

Figure 14: pyramide-impulse.tif

Figure 15: Après le filtre constante

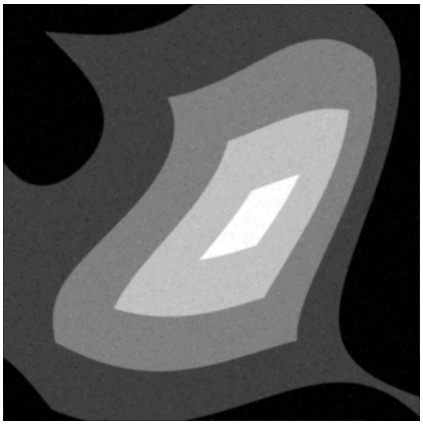

Figure 16: Après le filtre gausienne

Figure 17: Après le filtre median de type 1

Figure 18: Après le filtre median de type 2

Figure 19: Après le filtre median de type 3

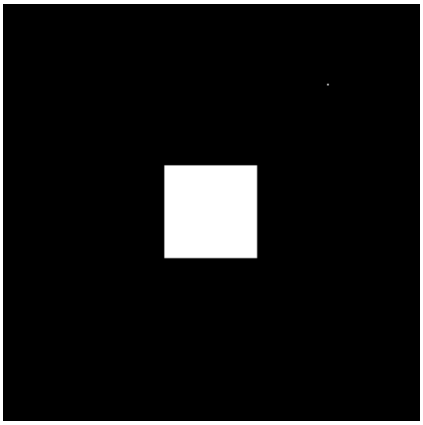

Figure 20: carre_orig.tif

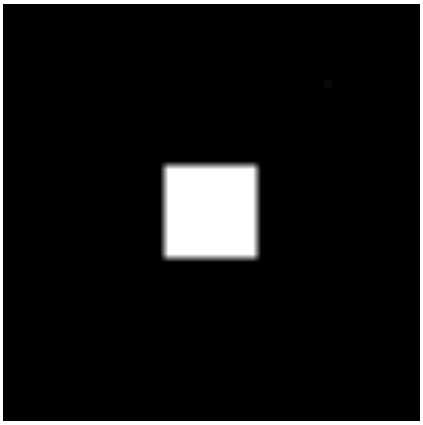

Figure 21: Après le filtre constante

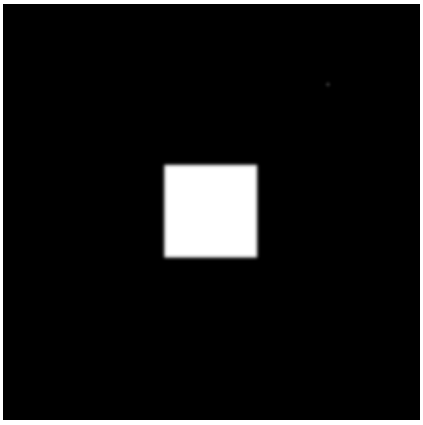

Figure 22: Après le filtre gausienne

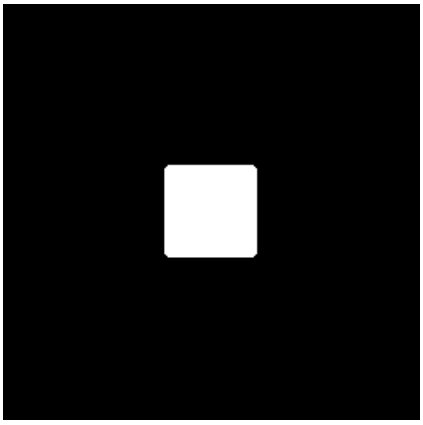

Figure 23: Après le filtre median de type 1

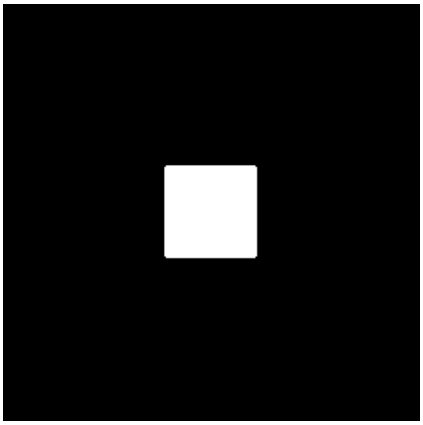

Figure 24: Après le filtre median de type 2

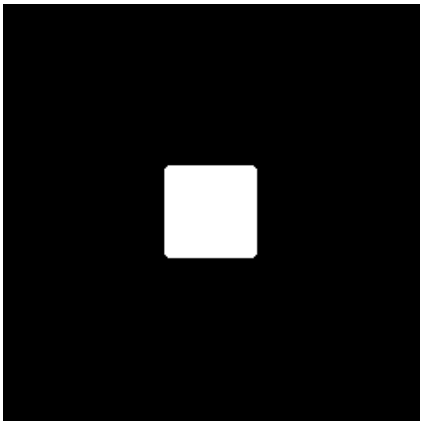

Figure 25: Après le filtre median de type 3

3 Restauration

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

   On utilise pyramide.tif pour tester cette restauration. On d'abord appliquer les filtre linéaires et ensuite on ajout des bruit gaussienne (br = 5). Après, on faire le filtre invers (la figure 27). On trouve que le filtre inverse peut restaurer l'image s'il n'y en a pas de bruit, comme la Figure 26, mais si l'on ajout un peu de bruit, cette méthode n'est plus valable, nous ne pouvons pas reconstruire la photo par cette méthode car le bruit est beaucoup plus grand qu'avant.

Figure 26: Après le filtre inverse de gaussien (sans bruit)

Figure 27: Après le filtre inverse de gaussien (avec bruit)

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

   On trouve qu'il y a un point de pixels très lumineux dans le coin supérieur droit de l'original de carre_flou.tif. Après filtrage, il reste des pixels légèrement plus lumineux au même endroit. Comme le filtre linéaire est effectué par convolution, le point lumineux devrait apparaître comme étant exactement le noyau de convolution utilisé après le filtrage dans la figure 28.

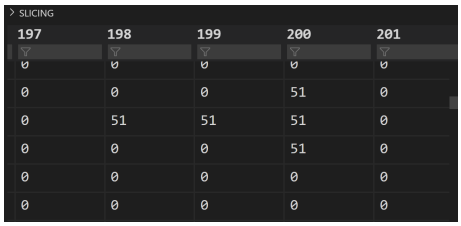

Figure 28: Noyau de convolution

3. 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.

   On utilise la fonction wiener pour restaurer cette image. On varie le paramètre λ et les résultats sont de Figure 29 à 34. On constate que le filtrage du bruit s'améliore quand le lambda augmente, mais si la valeur lambda est trop grande, l'effet sur le bruit n'est plus perceptible et les bords de l'image va devenir de plus en plus flous.

4 Applications

4.1 Comparaison filtrage linéaire et médi

1. Pour une image simple telle que carre 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)

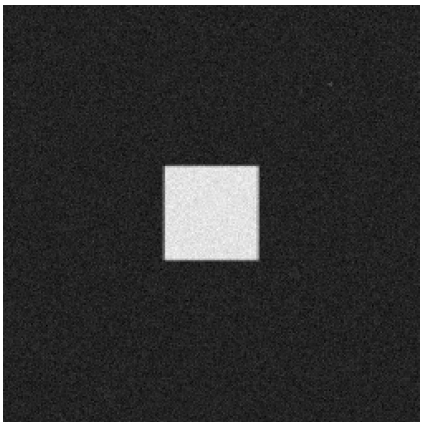

Figure 29: Carre_flou.tif avec bruit(br = 10)

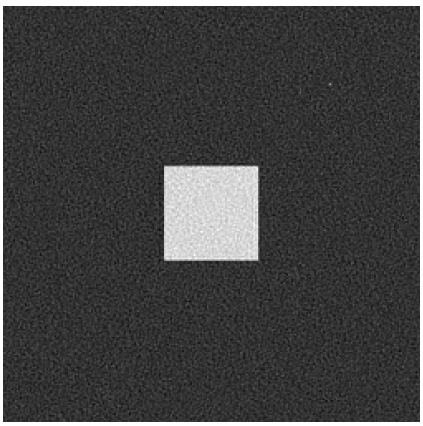

Figure 30: Après la restauration de wiener ($\lambda = 0.5$)

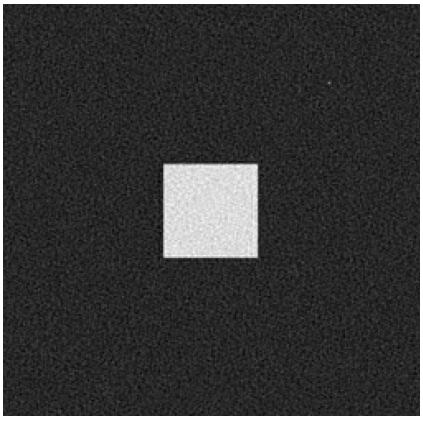

Figure 31: Après la restauration de wiener ($\lambda = 1.0$)

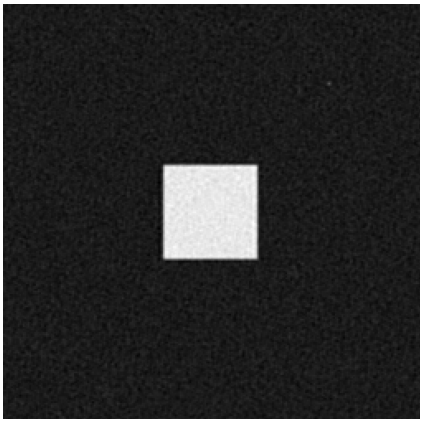

Figure 32: Après la restauration de wiener ($\lambda = 5.0$)

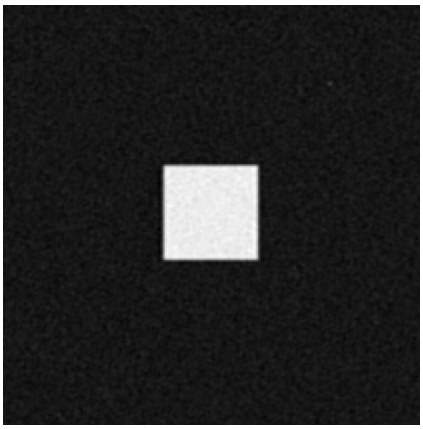

Figure 33: Après la restauration de wiener ($\lambda = 10.0$)

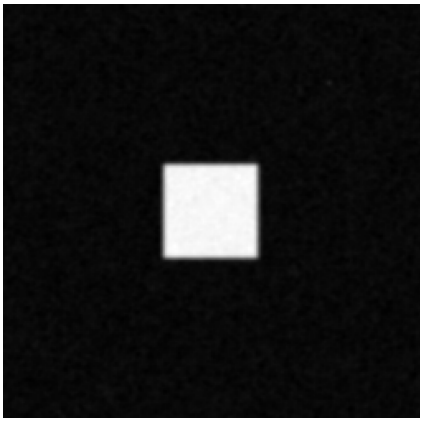

Figure 34: Après la restauration de wiener ($\lambda = 50.0$)

On d'abord créer un filtre médian circulaire de rayon 4 comme la référence. Ensuite, on veux utiliser le for loop de N iterations pour trouver la taille du noyau constant. Donc, on veux trouver

$$\text{argmin}_{\lambda \leq N} \|\text{Var\_image(median filtre)} - \text{Var\_image(filtre de noyau cst de taille } \lambda \times \lambda)\|$$

On calcule la variance de image sur la partie des pixels $[0, 10][0, 10]$. Et on pose $N = 100$. En faisant cet algorithme, on trouve que $\lambda = 2$ est le meilleur noyau constant du filtre linéaire.

4.2 Calcul théorique du paramètre de restauration

1. Modifiez la fonction wiener afin qu'elle utilise le spectre de l'image dégradée à place de $\lambda\omega^2$

   On a ajouté des bruit (br = 10) sur l'image carre_flou.tif modifié la fonction de wiener et en utilisant la nouvelle filtre, on a obtenu les images dans la Figure 35 et Figure 36. On trouve que l'effet de restauration de wiener en utilisant le spectre est meilleur que le wiener avant, si on connait le bruit ajouté.

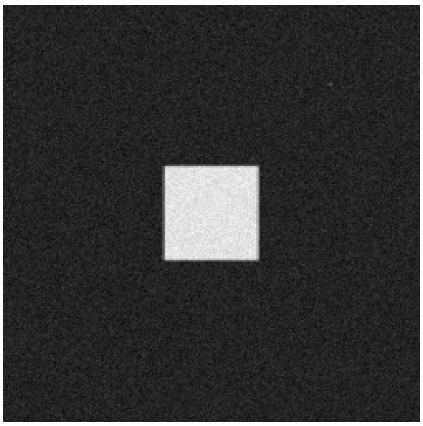

Figure 35: carre_flou.tif

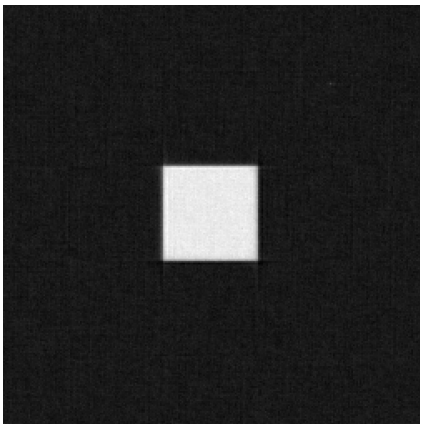

Figure 36: Après la restauration de wiener ($\sigma_b = 10$)