

TP filtrage-restauration

2 Transformation géométrique

La méthode à plus proche voisin préserve mieux les détails que la méthode bilinéaire, mais ce n'est pas flagrant après une seule transformation. Cependant, après 8 rotations, outre la différence de format selon le paramètre `clip`, l'image obtenue par rotation à plus proche voisin a des contours bien plus marqués que l'image obtenue par la méthode bilinéaire. Celle-ci ressemble bien plus à l'image initiale, avec seulement un léger flou généralisé.



image obtenue après 8 rotations méthode proche voisin
bilinéaire



image obtenue après 8 rotations méthode

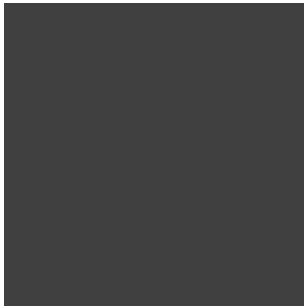
Si on applique la rotation avec un facteur de zoom inférieur à 1, on sous-échantillonne l'image, et le résultat est pixellisé. Pour atténuer l'effet constaté, on peut appliquer un filtre passe-bas.



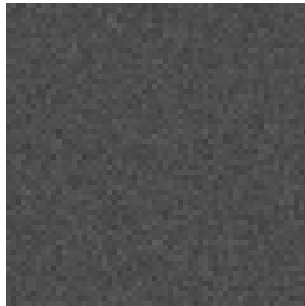
3 Filtrage linéaire et médian

La taille du noyau renvoyé par `get_gau_ker` varie en $25s^2$. Cela vient du fait que le côté du noyau est en $2*2,5*s$, par la ligne `ss=int(max(3,2*np.round(2.5*s)+1))`.

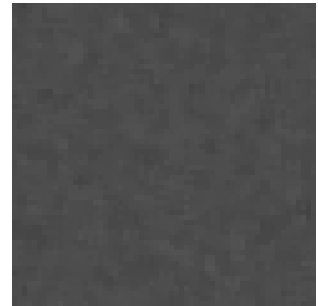
En ajoutant du bruit, puis en mesurant la variance locale sur un carré où l'image initiale semble constante, on voit que le filtre diminue grandement cette valeur (par ex, on a une variance de 25 sur l'image bruitée, contre une variance de 4 sur l'image bruitée puis filtrée, pour la portion suivante)



portion initiale
médian



portion bruitée (bruit gaussien, paramètre 5)



portion filtrée filtre

Avec `median filter`, les contours sont plus préservés que avec le filtre linéaire, qui présente un flou important :



image filtrée par filtre median

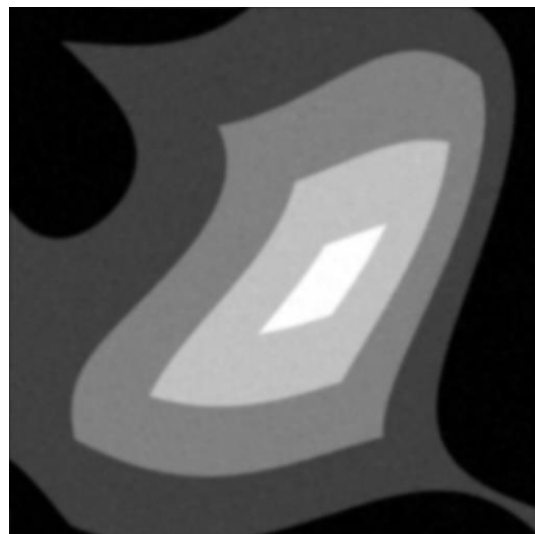


image filtrée par filtre linéaire

Pour l'image **carre_orig.tif**, le filtrage linéaire floute les contours, et le point lumineux situé en haut à droite est estompé en une tache de très faible luminosité, due à la convolution. Avec le filtrage median, le point blanc est tout simplement effacé, car la médiane de ses voisins est un pixel noir.

4. Restauration

Lorsque l'on utilise la fonction `filtre_inverse` après filtrage linéaire (avec le même noyau), l'image est parfaitement restaurée. Cependant, si on ajoute un léger bruit entre le filtrage et le filtrage inverse, l'image est complètement perdue :

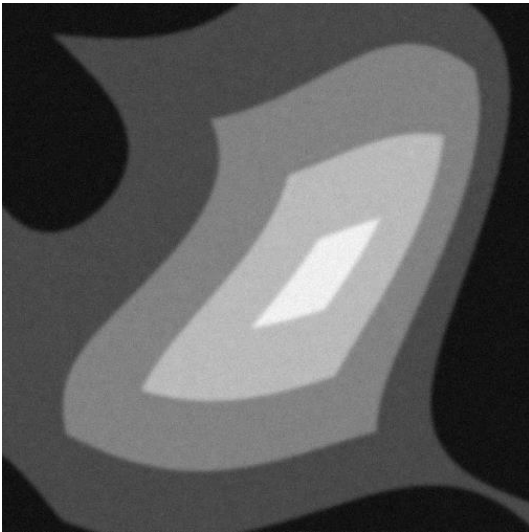


image filtrée bruitée

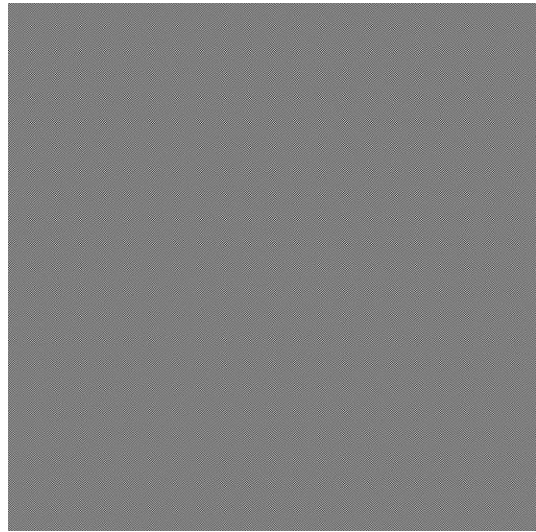


image restaurée par filtre inverse

Pour déterminer le noyau de convolution subi par **carre_flou.tif**, on peut regarder la forme qu'a le point en haut à droite de l'image. Celui-ci a été transformé en une nouvelle forme, qui correspond au noyau par lequel l'image a été convoluée :

