

TP2

Télécom Paris – IMA201

Cristian Alejandro Chávez Becerra

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 ?



Image originale

Image tourné 45

La différence est que la méthode du plus proche voisin ne lisse pas les couleurs, elle prend les valeurs existantes de l'image originale, et non les couleurs les unes des autres, ce qui peut lisser l'image. En revanche, la méthode bilinéaire peut prendre une valeur dans la gamme des valeurs de ses pixels adjacents, ce qui génère un plus grand lissage.

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

Cristian Alejandro Chávez Becerra



Image tourné 8 fois avec ppv

Image tourné 8 fois avec bilinear

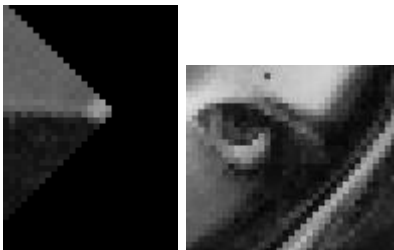
Nous pouvons comprendre que l'image pivotée à l'aide de ppv a une qualité inférieure en raison de la nature même de la méthode, qui lisse son interpolation, mais en prenant les mêmes valeurs de l'image originale, elle reproduit les mêmes valeurs de certains de ses pixels voisins, avec davantage de blocs de mêmes couleurs, ce qui explique qu'elle semble plus pixellisée.

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 ?

Cristian Alejandro Chávez Becerra



Rotation 45 + Zoom 50



ppv

Cristian Alejandro Chávez Becerra



Rotation 45 + Zoom 50



Bilinear

Comme un pixel de la nouvelle image représente une plus grande surface de l'image originale, nous sommes confrontés à un problème de sous-échantillonnage (selon le théorème de Shannon). Il convient donc d'appliquer un filtre passe-bas à l'image avant le dézoomage afin que le pixel de la nouvelle image puisse mieux représenter l'image originale.

3 Filtrage linéaire et m'édian

Dans cette section on se propose de comparer les filtrages linéaire et m'édian. Pour appliquer un filtrage linéaire il faut d'abord générer un noyau. Le filtrage linéaire consiste à convoluer le noyau avec l'image. Vous pouvez utiliser les commandes `get_gau_ker` et `get_cst_ker` pour générer un noyau gaussien ou constant. Pour appliquer ces noyaux il faut utiliser la commande `filtre_linéaire`. La commande `médian_filter` applique un filtre m'édian.

Cristian Alejandro Chávez Becerra

— Expliquer le rapport entre la taille du noyau (size) renvoyé par get_gaussian et le paramètre cette commande.

```
ss=int(max(3,2*np.round(2.5*s)+1))
ms=(ss-1)//2
X=np.arange(-ms,ms+0.99)
y=np.exp(-X**2/2/s**2)
out=y.reshape((ss,1))@y.reshape((1,ss))
out=out/out.sum()
```

nous calculons ss en utilisant s, ce résultat ss sera le n=ss où nxn est égal aux dimensions de notre noyau résultant comme on peut le voir dans la ligne

```
out=y.reshape((ss,1))@y.reshape((1,ss))
```

dans cette ligne, nous effectuons une multiplication de matrices (ou plutôt de vecteurs) de dimension ss qui nous donnera les dimensions de notre noyau résultant



Image original

Image avec kernel gaussien appliqué

— 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).

Cristian Alejandro Chávez Becerra

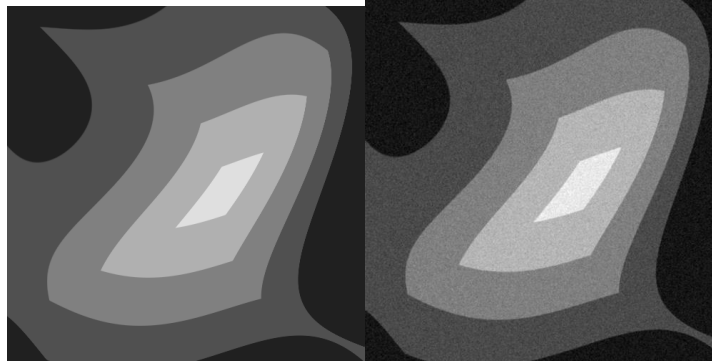


Imagen original

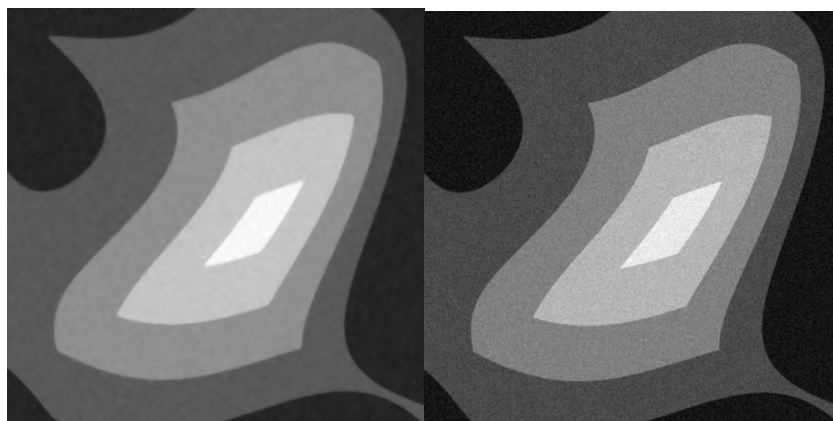
Imagen filtrada después de aplicar ruido

Nous pouvons évaluer la quantité de bruit résiduel en calculant la différence entre la variance d'une région dans l'image originale et la variance de la même région dans l'image résultante après application du bruit et passage à travers un filtre linéaire.

```
var_image(im,0,0,10,10)
✓ 0.0s
0.0
```

```
[59] var_image(filtre_lineaire(noise(im,10),get_cst_ker(2)),0,0,10,10)
✓ 0.1s
... 27.306956096340556
```

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

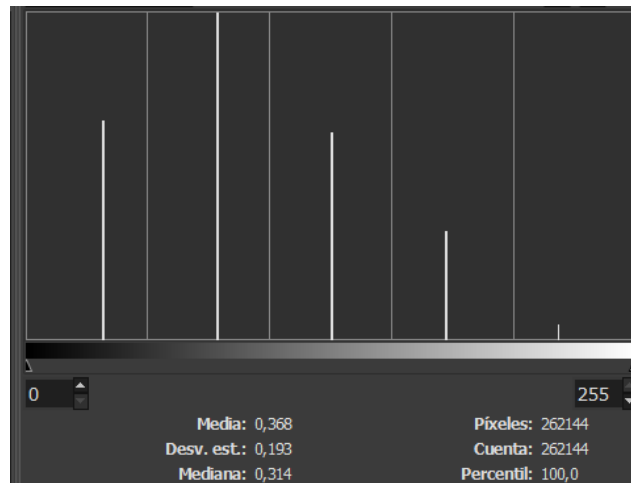


Cristian Alejandro Chávez Becerra

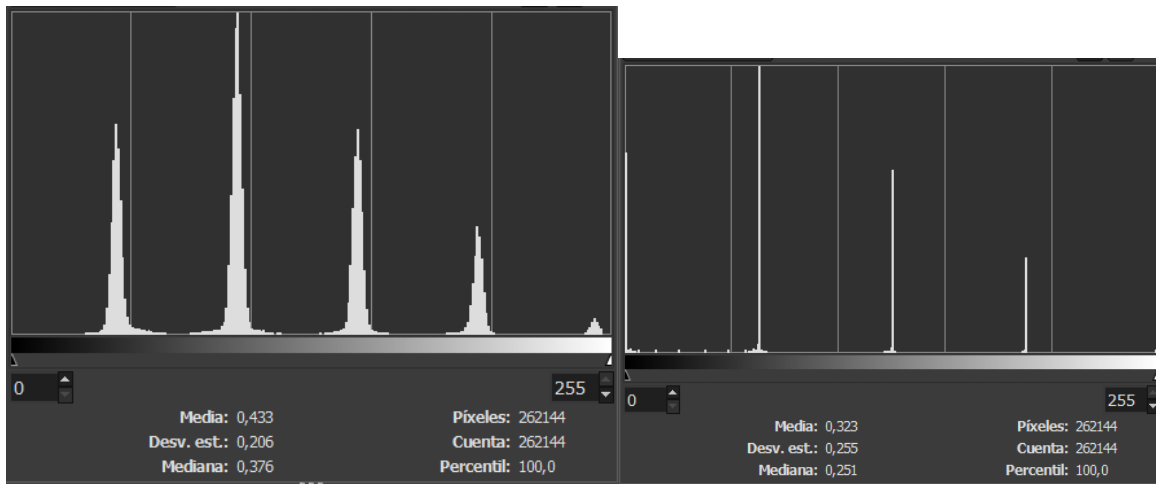
Image avec filtre médian

Image avec filtre linéaire

On constate que le filtre médian favorise davantage les tons clairs pris par le bruit gaussien ajouté qui éclaircit, en général, un peu plus l'image, et donc les médianes sont également distribuées un peu plus vers les tons clairs.



Histogramme de l'image originale

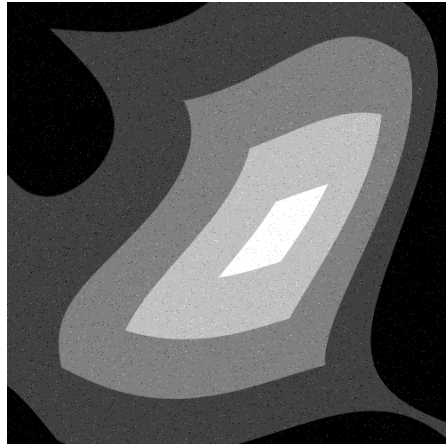


Filtre médian à histogramme

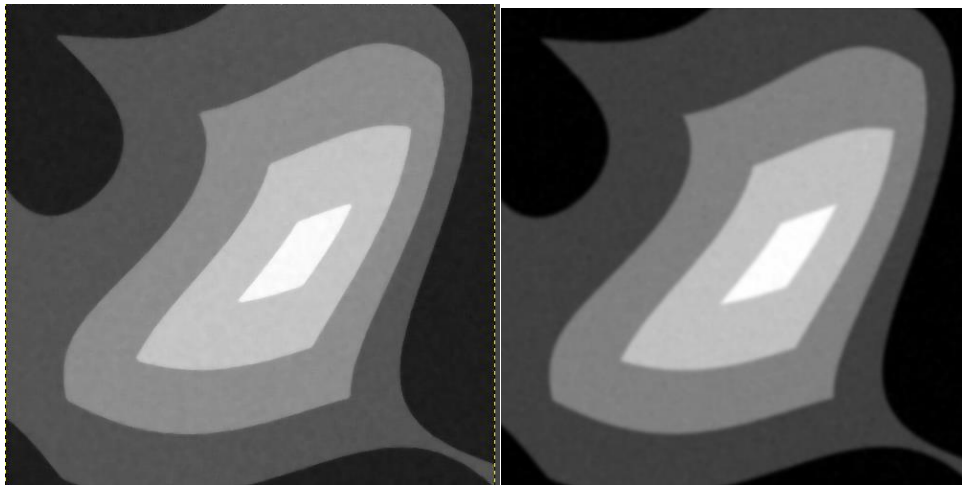
Histogrammee filtre lineare

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

Cristian Alejandro Chávez Becerra



Pyra-impulse.tif



Filtre median

Filtre lineaire

On constate que les filtres ont permis d'améliorer le bruit (poivre et sel ou impulsif) de l'image originale et que, comme dans la question précédente, la médiane a donné un ton plus clair à l'image.

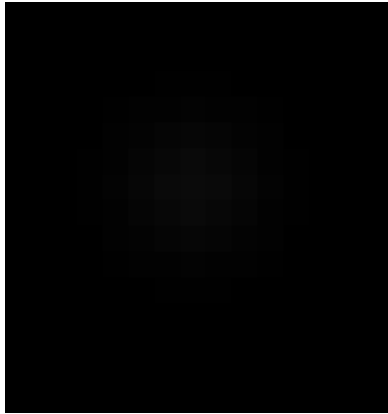
Expliquer la différence de comportement entre filtrage linéaire et médian sur le point lumineux situé en haut à droite de l'image carre orig.tif.

Le filtre médian, en prenant la médiane des valeurs, a estompé le point blanc, car la médiane de 0, 1, 1, 1, 1, 1, 1, 1, 1, 1 est $(1+1)/2 = 1$.

1 1 1		1 1 1
1 0 1	->	1 1 1
1 1 1		1 1 1

Cristian Alejandro Chávez Becerra

En revanche, le filtre linéaire permet d'apprécier, bien que de manière un peu floue, le point blanc, car en raison de la nature de ce type de filtre, la tonalité du pixel blanc a contribué au résultat final.



Punto blanco en el filtro linear

4 Restauration

— Appliquer un filtre linéaire à une image puis utilisez la fonction filtre inverse 2 . Que constatezvous ?

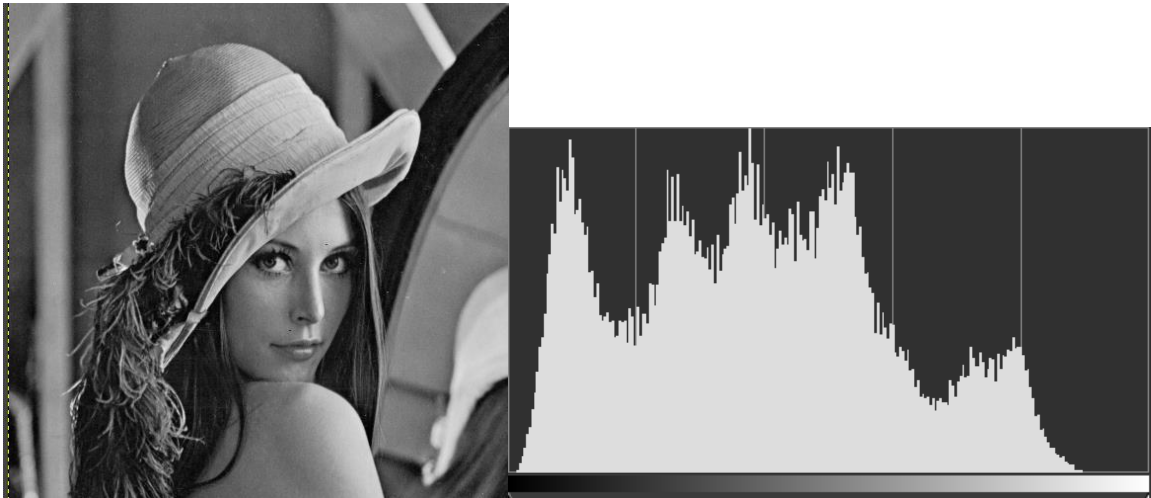


Image originale et histogramme

Cristian Alejandro Chávez Becerra

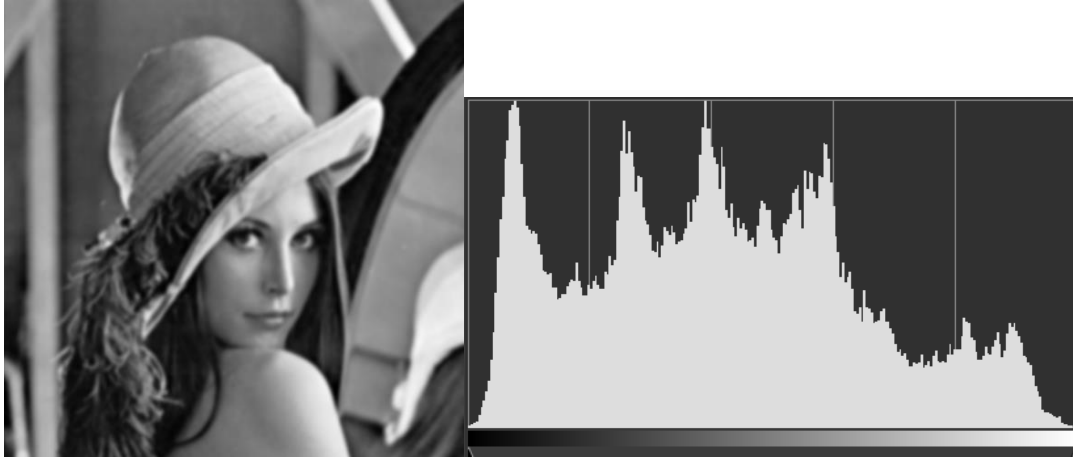


Image avec filtre linéaire et noyau gaussien

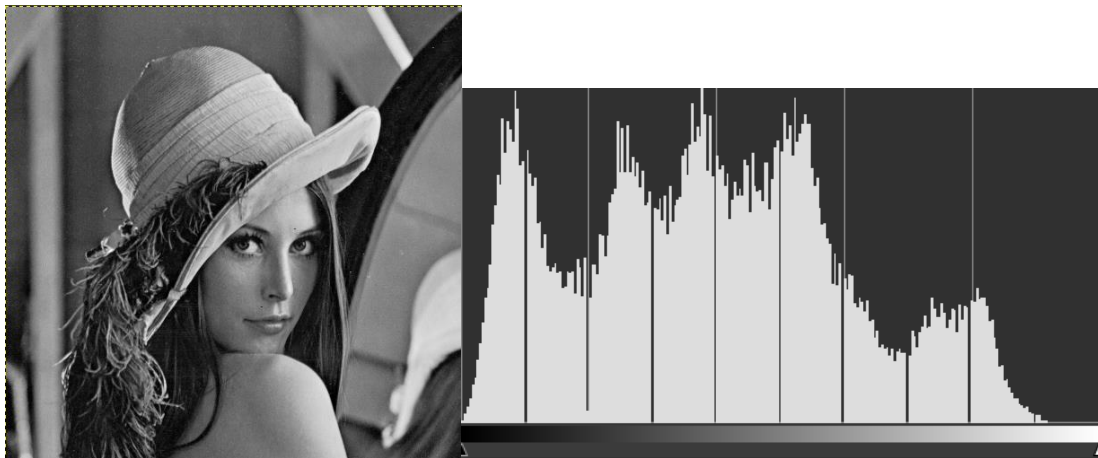


Image passée par la fonction de filtre inverse

Bien que l'histogramme montre que l'image passée par le filtre_inverse n'est pas exactement la même dans ses tonalités, on peut apprécier esthétiquement qu'une grande partie de l'image originale a été récupérée, car si nous considérons le filtre linéaire comme un flou, le flou fonctionne comme un opérateur passe-bas et les images contiennent naturellement beaucoup de basses fréquences, de sorte que peu d'informations sont perdues, de sorte qu'en multipliant par l'inverse, nous pouvons récupérer beaucoup de caractéristiques de l'image originale.

Que se passe-t-il si vous ajoutez très peu de bruit à l'image floutée avant de la restaurer par la commande précédente ?

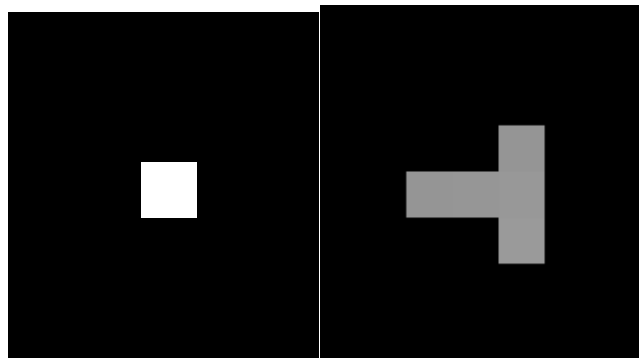
Cristian Alejandro Chávez Becerra



Image avec peu de bruit Image avec peu de bruit après application du filtre_inverse

Comme le bruit a suffisamment d'énergie dans toutes les fréquences, hautes et basses, en multipliant l'inverse de la matrice, on multiplie beaucoup les hautes fréquences de l'image dans lesquelles le bruit prédomine.

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

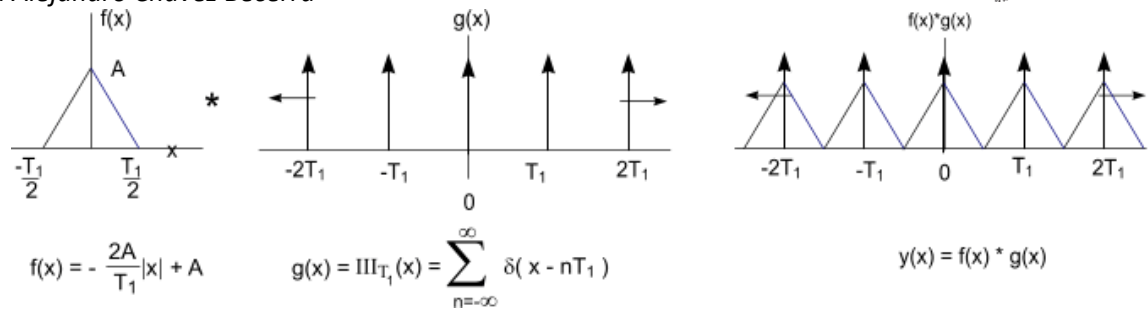


Carre_orig Carre_flou

Il y a deux façons de sonder le noyau de convolution :

1. puisqu'il y a un pixel blanc, nous pouvons considérer cette région comme une fonction delta de Dirac $\delta(x)$ qui agit en convolution avec le noyau, ce qui nous permet de voir le noyau dans la région resultante

Cristian Alejandro Chávez Becerra



2. Le noyau peut également être étudié en pensant à la convolution de deux matrices

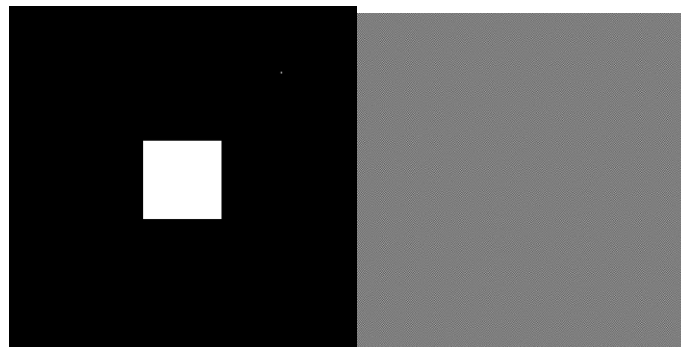
$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} * \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} = (i \cdot 1) + (h \cdot 2) + (g \cdot 3) + (f \cdot 4) + (e \cdot 5) + (d \cdot 6) + (c \cdot 7) + (b \cdot 8) + (a \cdot 9).$$

Ce qui nous permet de dire que le noyau était

$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

— Après avoir ajouté du bruit à cette image utilisez la fonction wiener pour restaurer cette image.

λ (qui est spécifié par l'utilisateur) est proportionnel à la variance du bruit (σ^2) et inversement proportionnel à la dépendance du spectre de l'image en fonction de ω^2 . Faites varier le paramètre λ et commentez les résultats.



Lambda = 1 Lambda=8

Ce que l'on observe est un compromis, car lorsque l'on augmente la valeur de λ , la variance du bruit est plus élevée, de sorte que le bruit est réparti entre les différents tons, mais comme il dépend moins du spectre, il dépend moins de l'image originale, de sorte qu'il prend moins d'informations de l'image originale.

5 Applications

5.1 Comparaison filtrage linéaire et médian

Cristian Alejandro Chávez Becerra

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

Ce que nous faisons, c'est évaluer la valeur qui a la variance la plus similaire.

```
im = skio.imread('images/carre_orig.tif')
imn = noise(im,5)
imm = median_filter(imn, typ=2, r=4)
imv = var_image(imm,0,0,imm.shape[0],imm.shape[1])
dif = np.inf
size = 0
for x in range(100):
    linear_var = var_image(filtre_lineaire(imn, get_cst_ker(i)),
0,0,imm.shape[0],imm.shape[1])
    if (abs(linear_var-imv)) < dif:
        dif = abs(linear_var-imv)
        size = i
```

5.2 Calcul théorique du paramètre de restauration

```
def wiener_spectre(im,K):
    fft2=np.fft.fft2
    ifft2=np.fft.ifft2
    (ty,tx)=im.shape
    (yK,xK)=K.shape
    KK=np.zeros((ty,tx))
    KK[:yK,:xK]=K
    x2=tx/2
    y2=ty/2
    fX=np.concatenate((np.arange(0,x2+0.99),np.arange(-x2+1,-0.1)))
    fY=np.concatenate((np.arange(0,y2+0.99),np.arange(-y2+1,-0.1)))
    fX=np.ones((ty,1))@fX.reshape((1,-1))
    fY=fY.reshape((-1,1))@np.ones((1,tx))
    fX=fX/tx
    fY=fY/ty
    w2=fX**2+fY**2
    w=w2**0.5
    g=fft2(im)
```

Cristian Alejandro Chávez Becerra

```
k=fft2(KK)
ss = (g)**2
sb = im.var()*len(im)/ss
mul=np.conj(k)/(abs(k)**2+(ss/sb)*w2)
fout=g*mul
mm=np.zeros((ty,tx))
y2=int(np.round(yK/2-0.5))
x2=int(np.round(xK/2-0.5))
mm[y2,x2]=1
out=np.real(iff2(fout*(fft2(mm))))
return out
```