

Filtrage d'une image par convolution

1. Introduction

Le filtrage d'une image numérique permet de modifier son spectre spatial. On peut par exemple chercher à atténuer les hautes fréquences pour la rendre moins nette, à réduire le bruit, ou au contraire à accentuer les hautes fréquences pour accentuer la netteté. La dérivation est aussi une opération de filtrage, employée pour la [détection des bords](#).

Le principe du filtrage numérique par convolution pour les signaux unidimensionnels est expliqué dans [Introduction aux filtres numériques](#).

Ce document explique le principe du filtrage des images (signal bidimensionnel) par convolution et montre des exemples de filtres passe-bas utilisés couramment. Le filtrage par convolution s'applique aux filtres linéaires à réponse impulsionnelle finie, dont la réponse impulsionnelle est relativement petite. Il existe d'autres techniques de filtrage, comme le filtrage par transformée de Fourier ou le filtrage récursif.

2. Principe

On considère une image numérique en niveaux de gris, dont la valeur au pixel (i, j) est notée $X_{i,j}$. On adopte la convention qui consiste à placer l'indice de colonne (indice i) en premier, car cet indice correspond à l'axe x sur l'image. La réponse impulsionnelle dans le cas d'un signal numérique à deux dimensions est elle-même un signal à deux dimensions, c'est-à-dire une matrice, dont les coefficients seront notés $h_{m,n}$. Pour simplifier, on se limite au cas des réponses impulsionnelles carrées, avec un nombre de termes sur chaque dimension impair $N = 2P + 1$. Par exemple, pour $P = 1$:

$$H = \begin{pmatrix} h_{00} & h_{10} & h_{20} \\ h_{01} & h_{11} & h_{21} \\ h_{02} & h_{12} & h_{22} \end{pmatrix} \quad (1)$$

Cette matrice est aussi appelée le masque de convolution. L'image filtrée Y s'obtient en effectuant le produit de convolution entre X et H , ce qui donne pour un pixel (i, j) de l'image finale :

$$Y_{i,j} = \begin{array}{lll} h_{00}X_{i-1,j-1} & +h_{10}X_{i,j-1} & +h_{20}X_{i+1,j-1} \\ +h_{01}X_{i-1,j} & +h_{11}X_{i,j} & +h_{21}X_{i+1,j} \\ +h_{02}X_{i-1,j+1} & +h_{12}X_{i,j+1} & +h_{22}X_{i+1,j+1} \end{array} \quad (2)$$

Le pixel $Y_{i,j}$ est donc obtenu en faisant une combinaison linéaire (ou moyenne pondérée) du pixel $X_{i,j}$ de l'image initiale et de ses 8 proches voisins. La relation ci-dessus se généralise aux masques de convolution 5×5 , 7×7 , etc.

On voit que la relation ne peut s'appliquer sur les bords de l'image, plus précisément sur les P rangées horizontales et verticales des bords. Il faut donc prévoir un traitement spécial pour ces rangées. La solution la plus simple, consiste à remplir ces rangées avec un niveau constant (par exemple noir). Une autre solution est de répliquer sur ces rangées les rangées voisines. Dans ces deux cas, l'information sur les bords est perdue. On peut aussi décider de ne pas modifier ces rangées, ce que nous allons faire. En tout cas, on évite de réduire la taille de l'image.

La matrice H est la réponse impulsionnelle du filtre. Pour le voir, considérons une impulsion unité, c'est-à-dire une image constituée d'un seul pixel (i, j) de valeur 1, tous les autres étant nuls. L'image de sortie est alors constituée de la matrice H centrée sur le pixel (i, j) . Elle comporte donc 9 pixels non nuls. Cette image est la réponse impulsionnelle, appelée aussi fonction d'étalement du point.

On considère ici le cas où la réponse impulsionnelle est invariante par translation (les coefficients de la matrice sont constants), mais il est possible d'appliquer un masque de convolution qui dépend de la position du pixel sur l'image, par exemple de sa distance au centre. Cette possibilité de filtrer localement en fonction de la position du point est même la propriété remarquable du filtrage par convolution, qu'il est impossible d'obtenir par un filtrage dans l'espace des fréquences spatiales.

3. Implémentation

La fonction suivante effectue la convolution. La matrice H doit avoir des dimensions impaires. On peut choisir de filtrer seulement la moitié gauche de l'image.

```
import numpy

def convolution2D(X,H,moitie):
    s = X.shape
    py = (H.shape[0]-1)/2
    px = (H.shape[1]-1)/2
    Y = X.copy()
    if moitie:
        imax = int(s[1]/2)
    else:
        imax = s[1]-px
    for i in range(px,imax):
        for j in range(py,s[0]-py):
            somme = 0.0
            for k in range(-px,px+1):
                for l in range(-py,py+1):
                    somme += X[j+1][i+k]*H[l+py][k+px]
            Y[j][i] = somme
    return Y
```

On peut aussi utiliser la fonction `scipy.signal.convolve2d` qui fera le calcul plus rapidement (car elle est implémentée en C compilé) et possède d'autres options pour les pixels des bords.

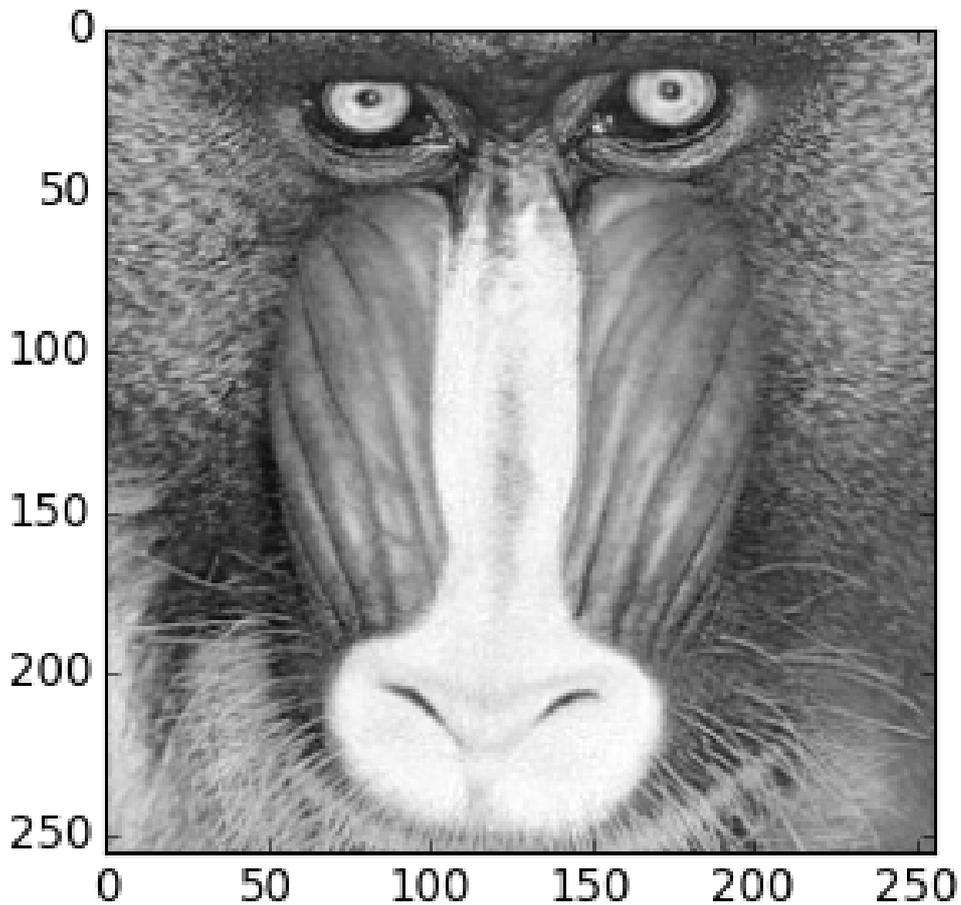
4. Exemples

4.a. Filtre moyenneur

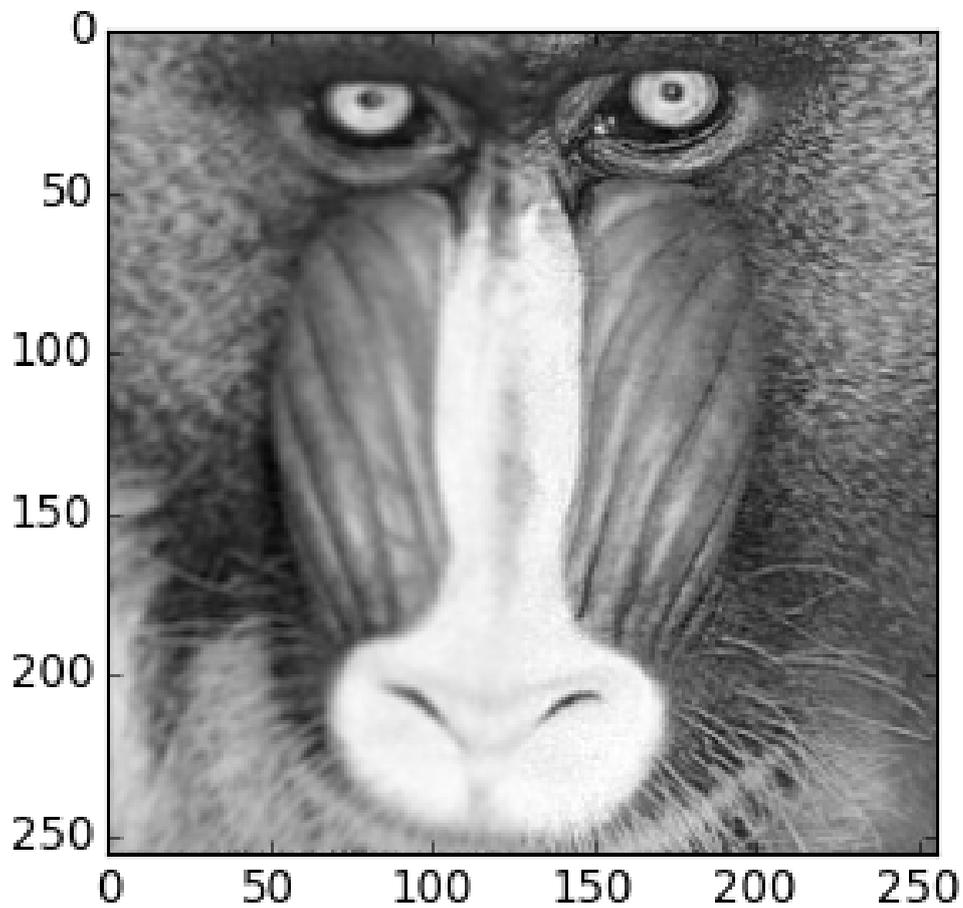
Ce filtre calcule, pour chaque pixel, la moyenne du pixel avec ses 8 proches voisins. Le masque de convolution est donc constitué de 9 coefficients égaux à $1/9$. Pour voir son

effet, on prend la couche rouge d'une image en couleur :

```
import imageio
from matplotlib.pyplot import *
img = imageio.imread(".././.././../simul/image/babouin.png")
red = img[:, :, 0]
green = img[:, :, 1]
blue = img[:, :, 2]
figure(figsize=(4,4))
X1 = red*1.0
imshow(X1, cmap=cm.gray)
```



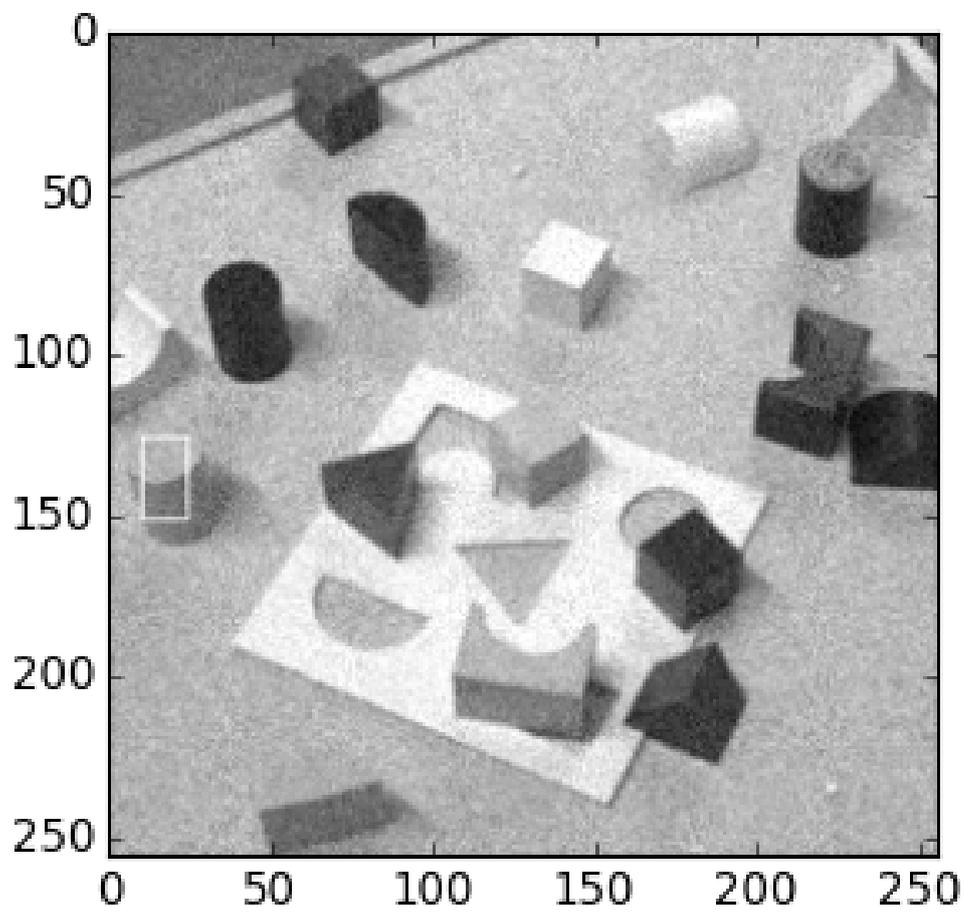
```
h = numpy.ones((3,3))*1.0/9
Y = convolution2D(X1,h,True)
figure(figsize=(4,4))
imshow(Y, cmap=cm.gray)
```



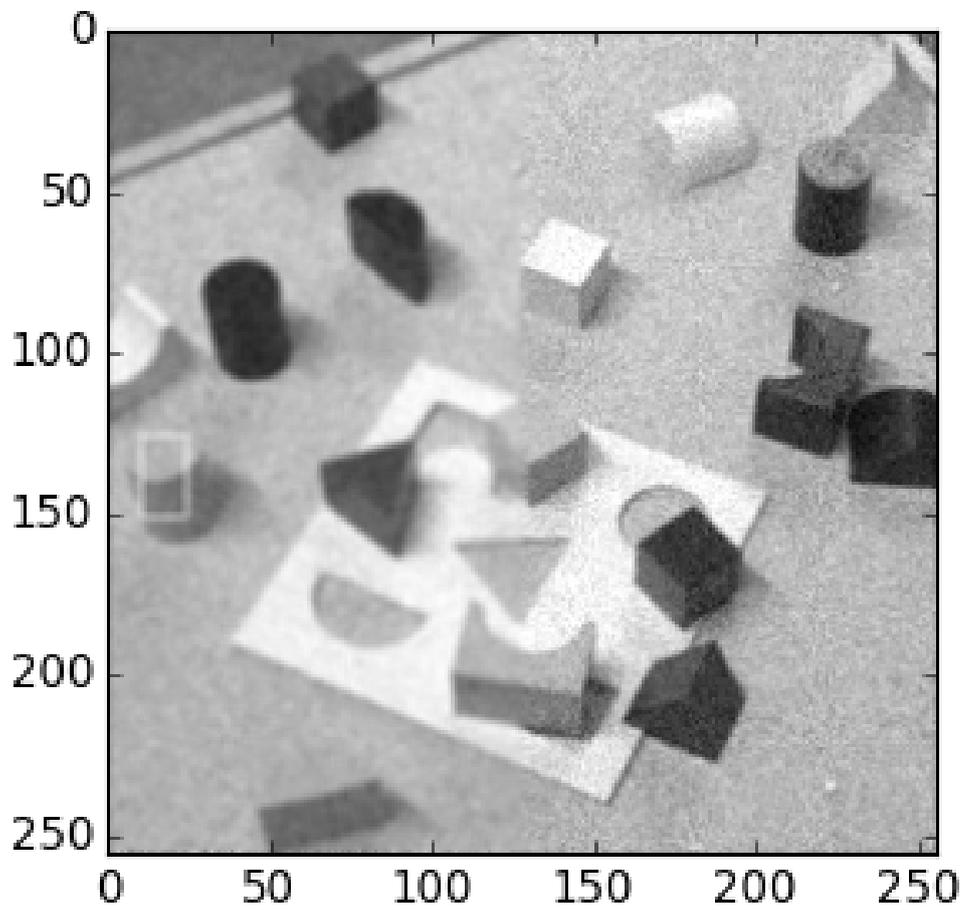
Il s'agit d'un filtre passe-bas, qui réduit la netteté de l'image. Cela peut être vu comme une détérioration de l'image, mais ce type de filtrage est parfois nécessaire avant d'appliquer d'autres traitements.

L'image suivante comporte un bruit important, qui a été ajouté artificiellement. Les photographies peuvent être très bruitées lorsqu'elles ont été prises dans des conditions de lumière faible.

```
img = imageio.imread("../..../simul/image/objets-bruit.png")
X2 = numpy.array(img)
figure(figsize=(4,4))
imshow(X2,cmap=cm.gray)
```

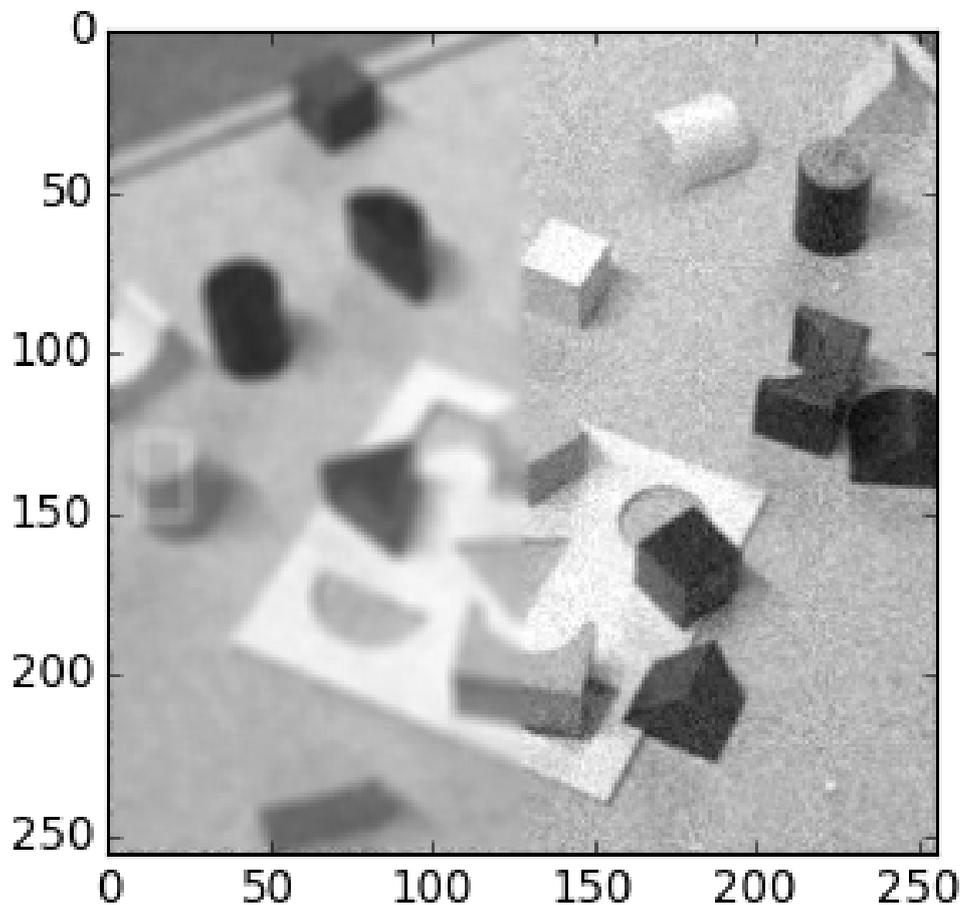


```
h = numpy.ones((3,3))*1.0/9
Y = convolution2D(X2,h,True)
figure(figsize=(4,4))
imshow(Y,cmap=cm.gray)
```



Le bruit est bien réduit (sur la moitié gauche). Pour un effet plus important, il faut augmenter la taille du masque :

```
h = numpy.ones((5,5))*1.0/25
Y = convolution2D(X2,h,True)
figure(figsize=(4,4))
imshow(Y,cmap=cm.gray)
```



La réduction du bruit s'accompagne d'une réduction de la netteté.

4.b. Filtre gaussien

Le filtre gaussien [1]) est un filtre passe-bas dont la réponse impulsionnelle continue est définie par la fonction suivante :

$$h(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (3)$$

Il faut l'échantillonner afin d'obtenir la réponse impulsionnelle discrète (la matrice H). Pour cela, on fixe la valeur de P puis on détermine σ afin que la valeur sur le bord de la matrice soit égale à une petite fraction ϵ de la valeur au centre. On obtient :

$$\sigma = \frac{P}{\sqrt{-2 \ln(\epsilon)}} \quad (4)$$

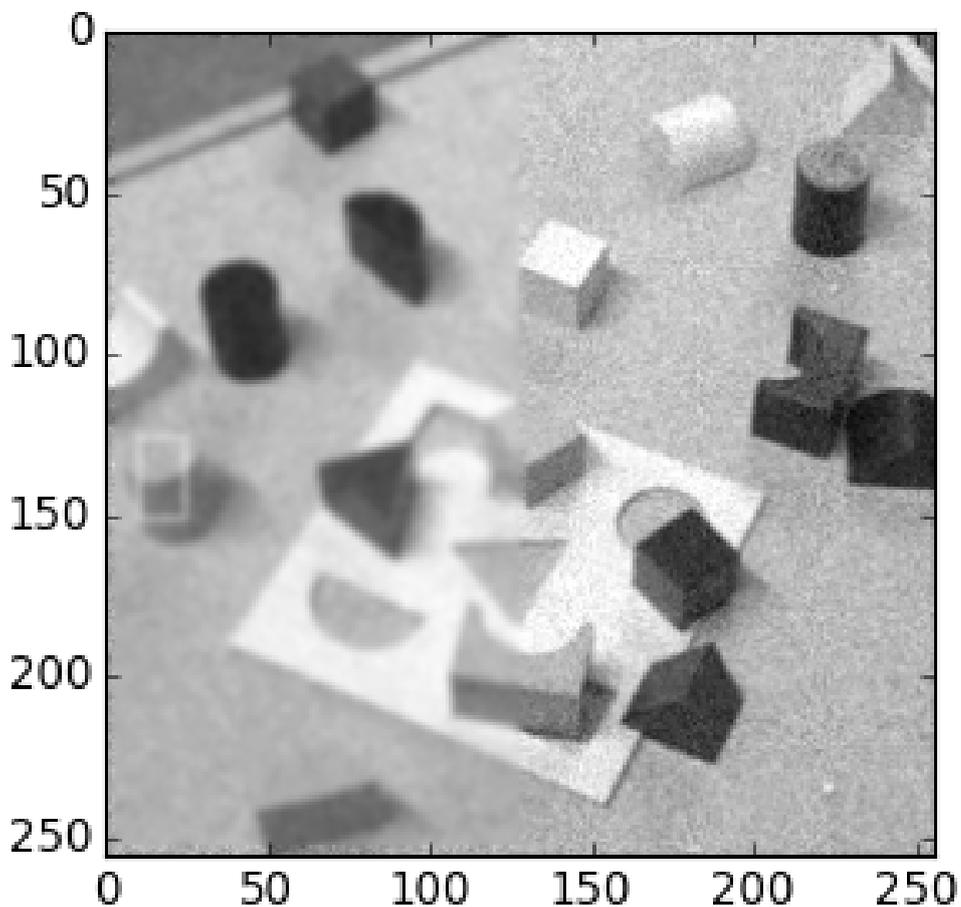
La fonction suivante permet de créer un filtre gaussien. La matrice est normalisée pour que la somme de ses coefficients soit égale à 1.

```
import math
def filtreGaussien(P):
```

```
epsilon = 0.05
sigma = P*1.0/math.sqrt(-2*math.log(epsilon))
h = numpy.zeros((2*P+1,2*P+1))
som = 0
for m in range(-P,P+1):
    for n in range(-P,P+1):
        h[m+P][n+P] = math.exp(-(n*n+m*m)/(2*sigma*sigma))
        som += h[m+P][n+P]
h = h/som
return h
```

Voici par exemple l'effet d'un filtre gaussien 7×7 sur l'image bruitée :

```
h = filtreGaussien(3)
Y = convolution2D(X2,h,True)
figure(figsize=(4,4))
imshow(Y,cmap=cm.gray)
```



Comparé au filtre moyennneur de taille 5×5 , le filtre gaussien 7×7 réduit autant le bruit sans altérer autant la netteté. De manière générale, le filtre gaussien est le meilleur filtre de lissage car sa réponse fréquentielle est aussi une gaussienne. Voir à ce sujet [Introduction aux filtres numériques](#). La réponse fréquentielle du filtre moyennneur est un sinus cardinal, ce qui peut conduire à l'apparition d'artefacts sur certaines images.

Références

- [1] M. Nixon, A. Aguado, *Feature extraction and image processing*, (Academic Press, 2008)