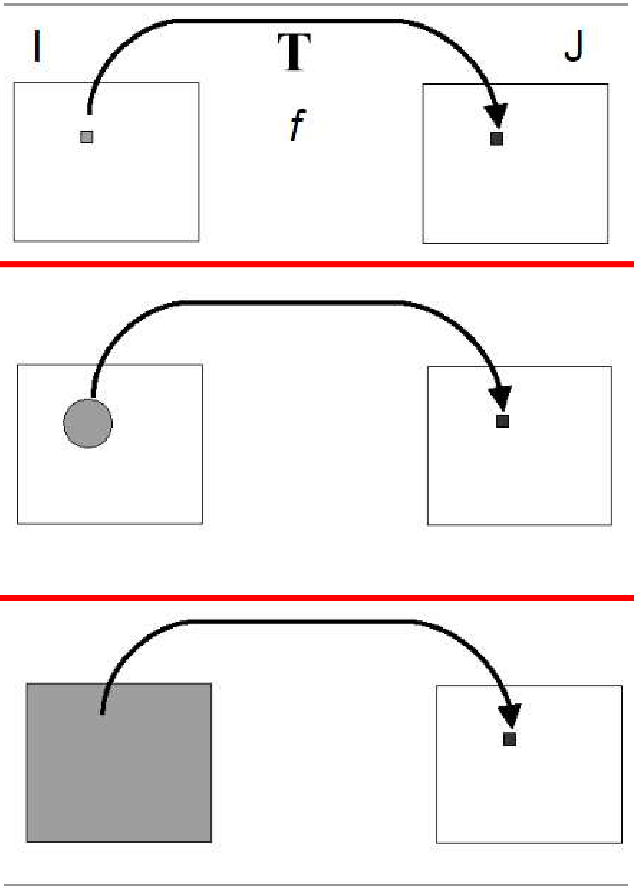
Traitements sur les images

Transformations locales



ponctuelles : J(x0,y0) = *f* [I(x0,y0)]

Opération sur les histogrammes

locales : J(x0,y0) = *f* [I(V)]

V: voisinage de (x0,y0)

Filtres,..

globales : J(x,y) = *f* [I(x,y]

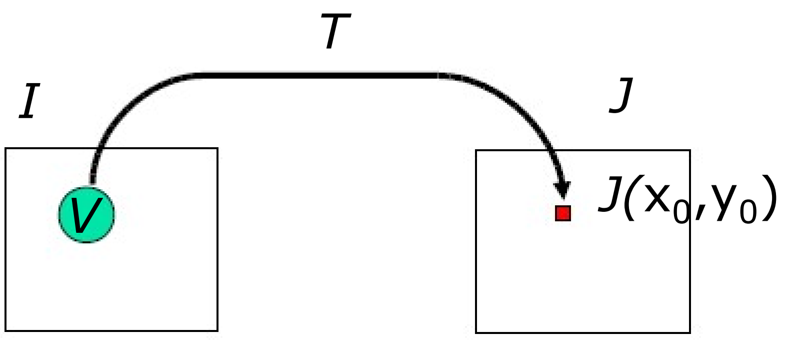
Transformée de fourier,..

Traitements locaux

- Principe
  - Lire la valeur de quelques pixels voisins
  - ⇒ calculer une nouvelle valeur pour un pixel à partir des valeurs du voisinage

$J(x_0,y_0)=T[I(V)]$

Avec V:voisinage de (x_0,y_0)



Traitements locaux

- Idée
  - Utiliser la redondance spatiale pour 2 catégories de problèmes différents
    - Améliorer l'image numérique
      - ⇒ Augmenter la qualité de son rendu visuel
        - Atténuation, suppression de dégradation
    - Simplifier l'image
      - ⇒ Dans le but de faciliter une analyse ultérieure
        - Élimination de structures inutiles du pt de vue de l'analyse visée

Traitements locaux

- Un outil : la convolution (domaine spatial)
  - Filtrage linéaire correspond à une multiplication dans le domaine fréquentiel de la fonction indicatrice du complémentaire du support du signal à éliminer et du signal à retrouver

    Avec f’(x,y) = f(x,y)+ s(x,y)

    f’(x,y) = image observée

    f(x,y): image initiale

    s(x,y): signal à éliminer (en général inconnu)

  - Cette opération peut se réaliser directement dans le domaine spatial par la convolution

Traitements locaux

- Objectifs de la convolution (éliminer le bruit, extraire les contours)

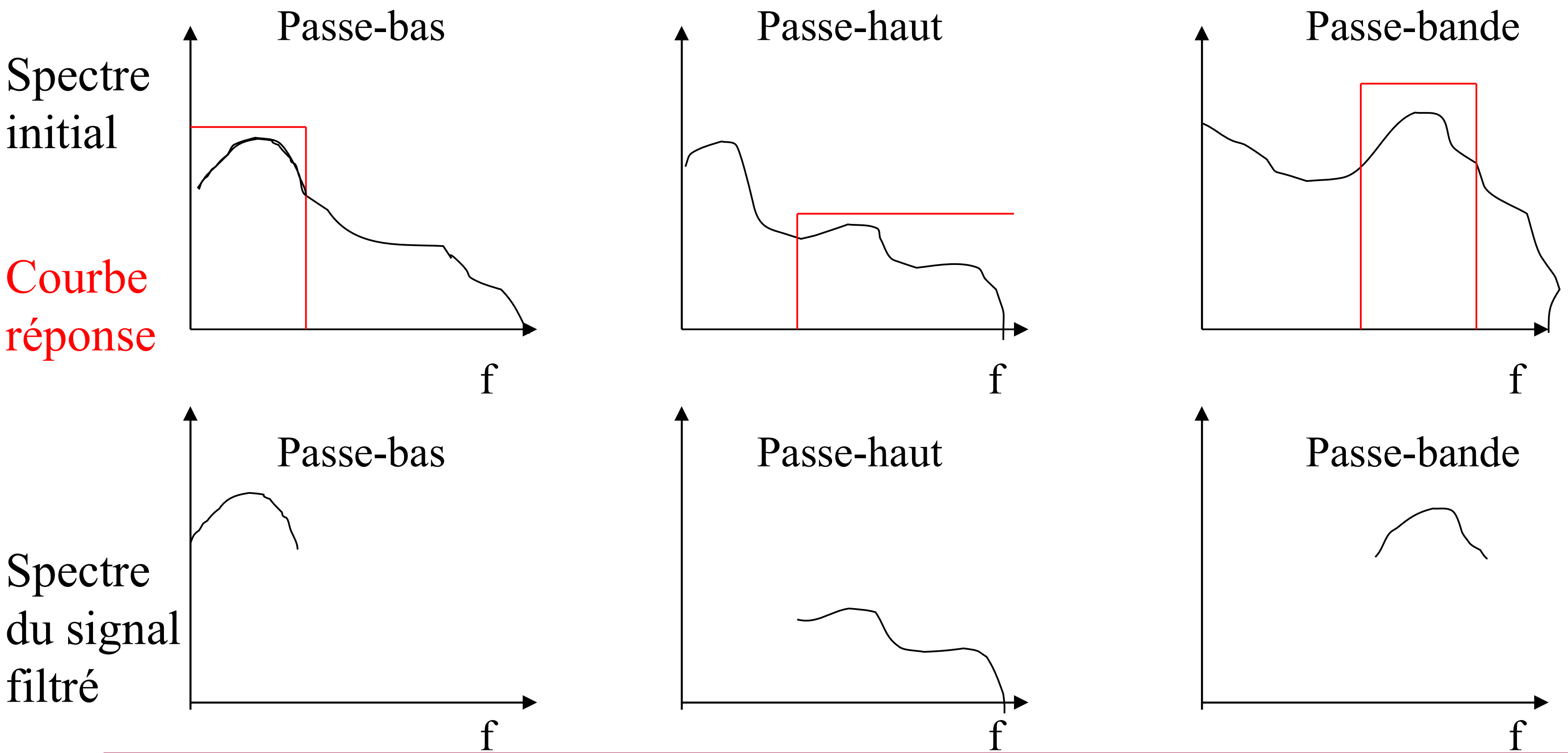
  Illustration de ce qui se passe dans le domaine fréquentiel

  - Supprimer certaines composantes de l'image du domaine des fréquences
    - filtre passe-haut : sélection des hautes fréquences (contours, ... Mais aussi bruit) ;
    - filtre passe-bas : sélection des basses fréquences

      ➔ lissage, défocalisation de l'image.

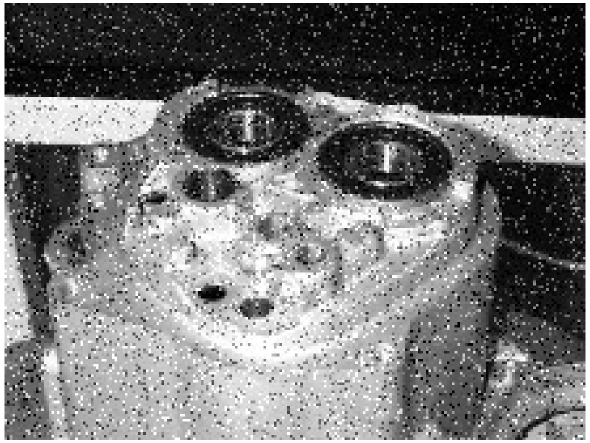
Traitements locaux

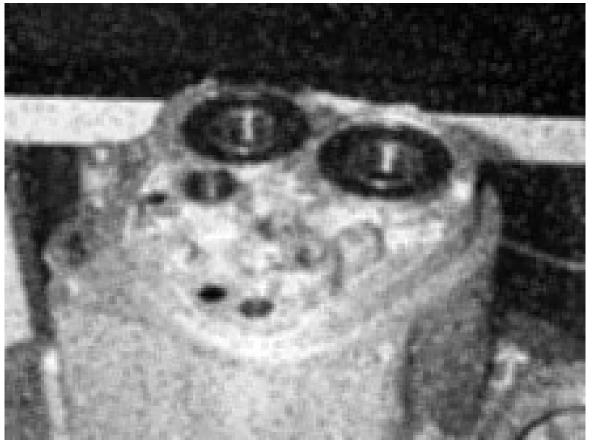
- Illustration des filtres passe-bas et passe-haut



Traitements locaux

- Illustration des filtres passe-bas
  - Bruit = hautes fréquences
    - élimination du bruit
    - Affaiblissement des contours (qui sont aussi des hautes fréquences)





Traitements locaux

- Illustration des filtres passe-haut et passe-bas

Image Originale

Image après filtre passe-haut

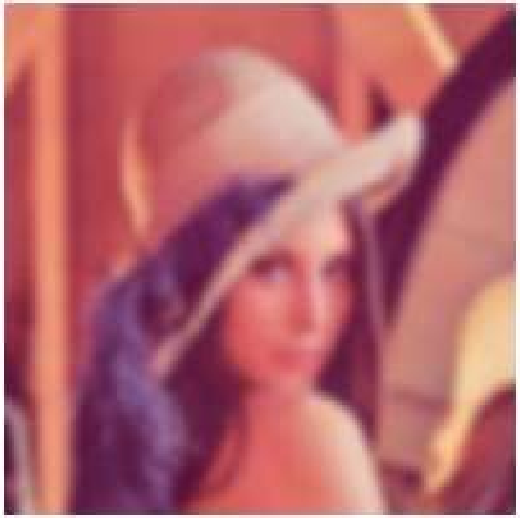


Image après filtre passe-bas

Traitements locaux

- Filtrage dans le domaine spatial
  - Un outil : la convolution
  - Un opérateur produit dans l'espace des fonctions

$$(f,g) \rightarrow h$$

$$(f \otimes g)(x) = \int_{-\infty}^{\infty} f(x-t)g(t)dt$$

- Avec

  f la fonction ou l'image initiale

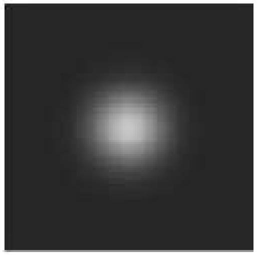
  g un motif de référence

  h l'image transformée
- Propriété
  - La convolution est commutative

Traitements locaux

- Un outil : la convolution

*



=

f: Image

g: motif de référence (masque)

h: Image convoluée (résultat)

Traitements locaux

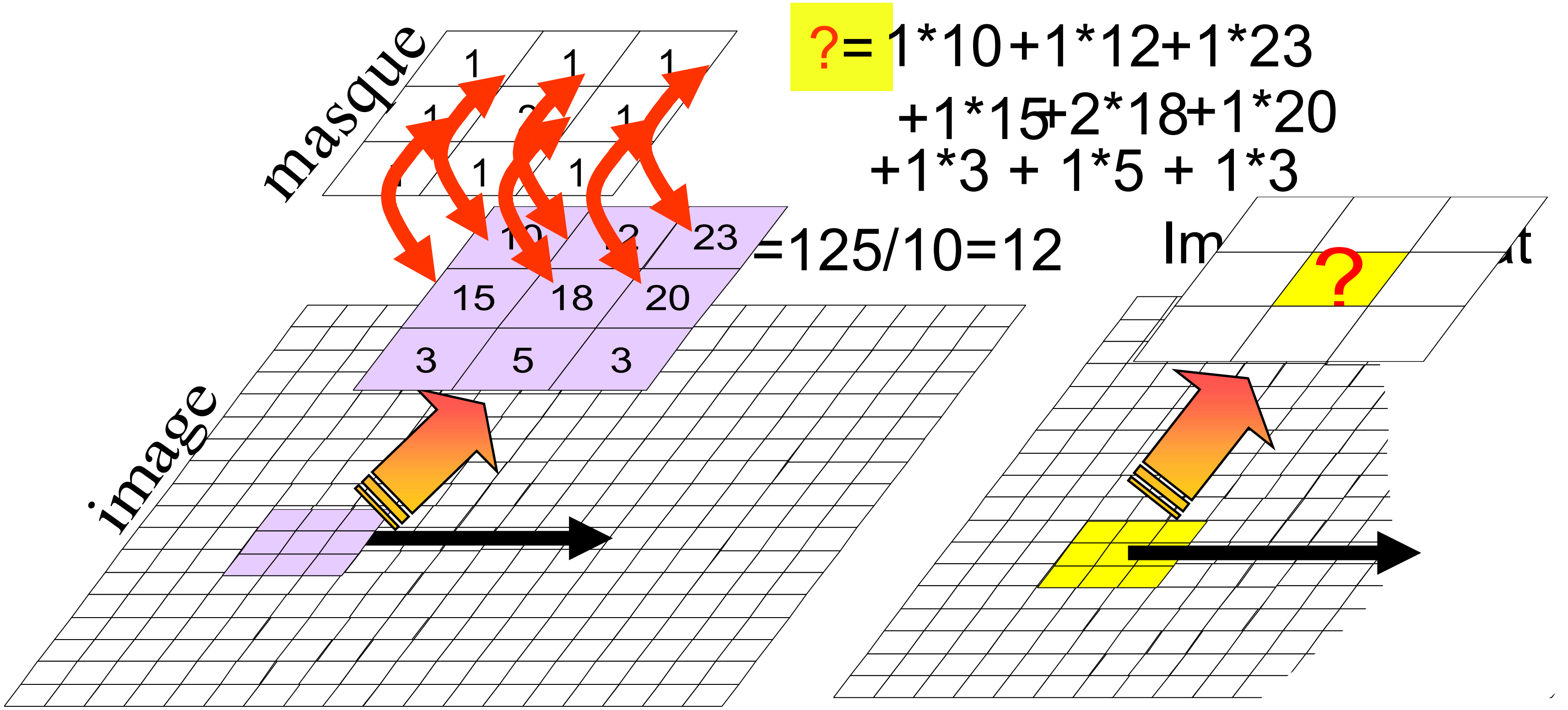
- La convolution discrète
  - Une image
    - a un support borné
    - est définie par une matrice $(f_{ij})_{ij}$
      - Avec i: indice de ligne,j, indice de colonne
  - Si le support de la fonction de référence g est un carré de côté 2p+1 centré à l'origine on a

$$f \otimes g(i,j) = \sum_{\alpha=-p}^{+p} \sum_{\beta=-p}^{+p} f_{i-\alpha,j-\beta} \cdot g(\alpha,\beta) = \sum_{\alpha=-p}^{+p} \sum_{\beta=-p}^{+p} f_{i-\alpha,j-\beta} \cdot a_{\alpha,\beta}$$

Avec $a_{\alpha,\beta}$ coefficient du masque de convolution

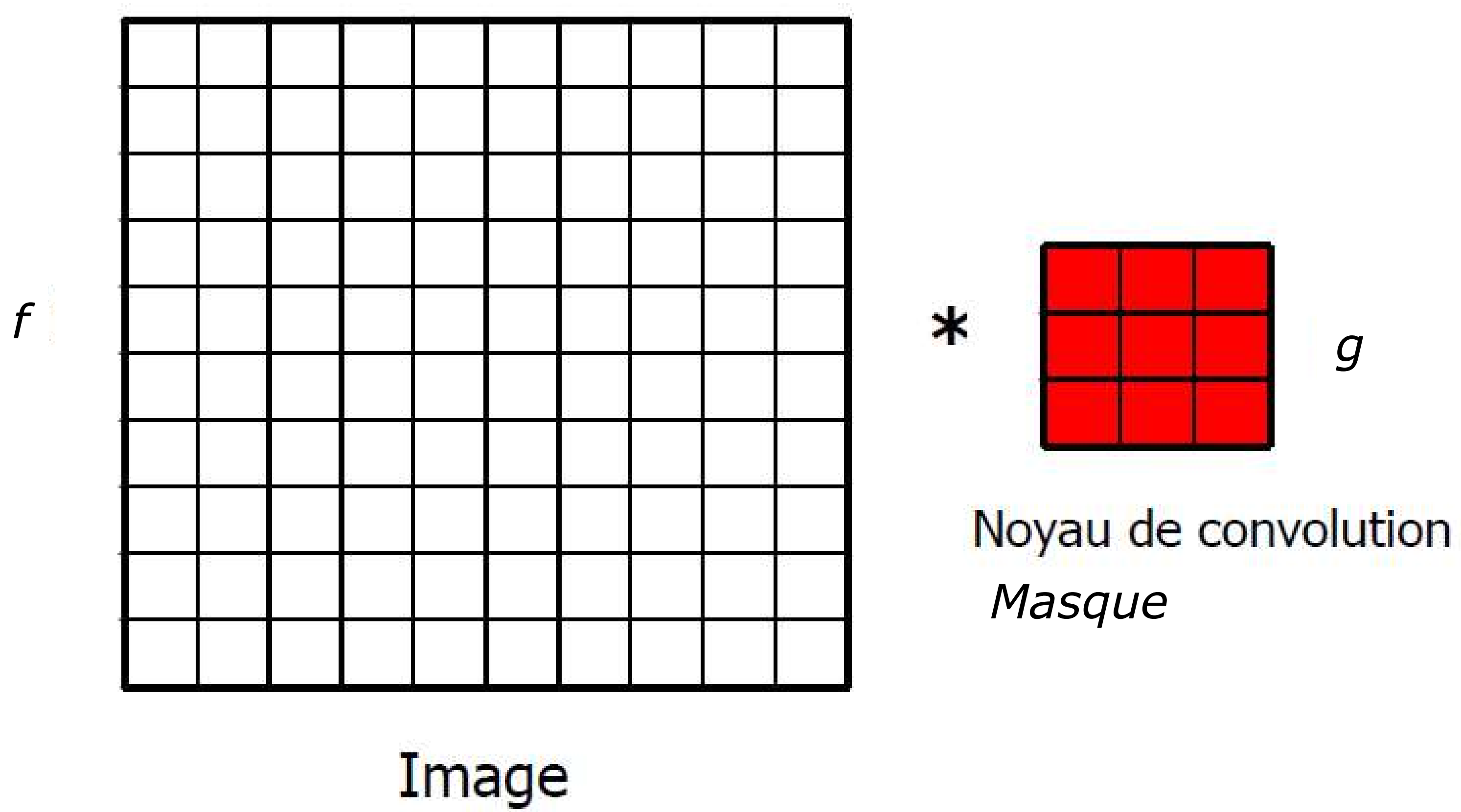
Traitements locaux

- Convolution numérique



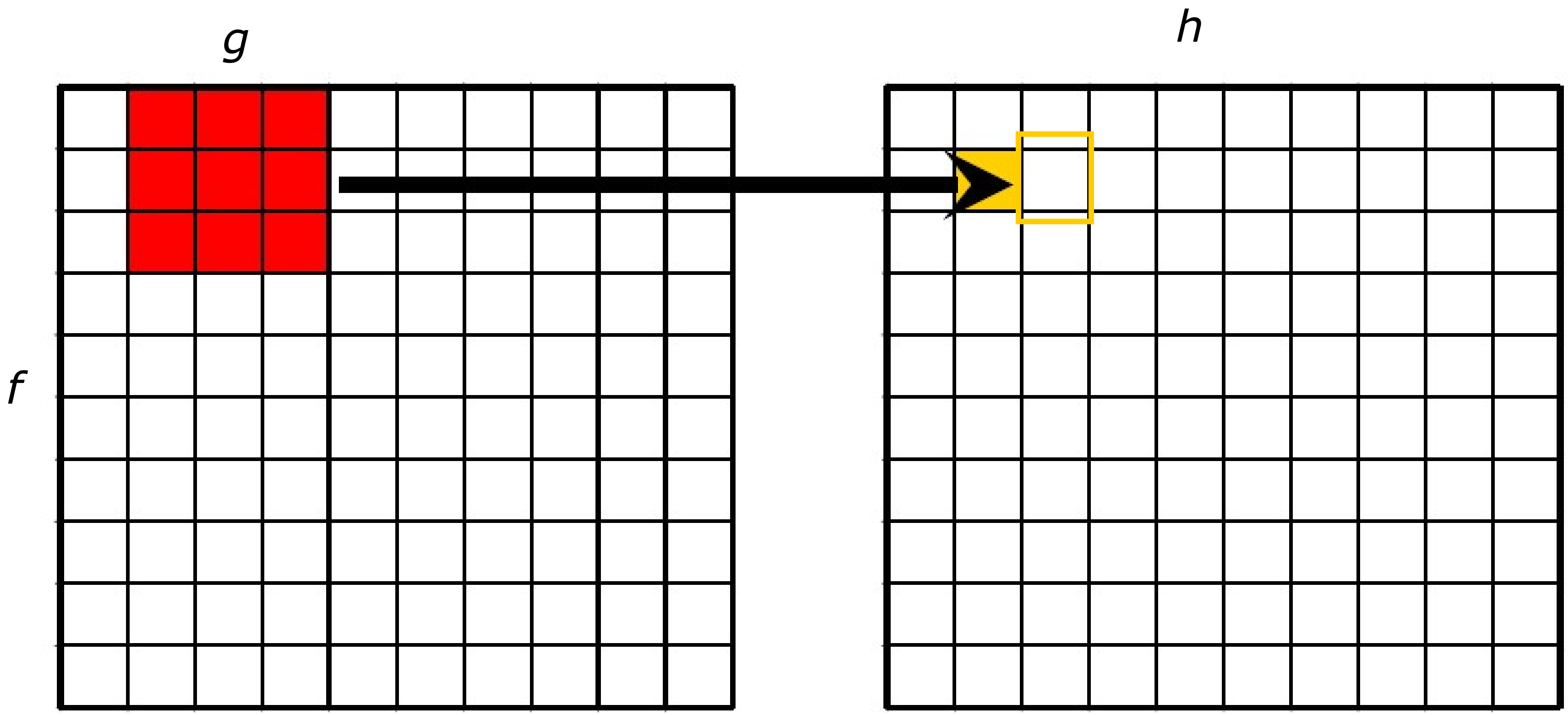
Traitements locaux

- Convolution numérique



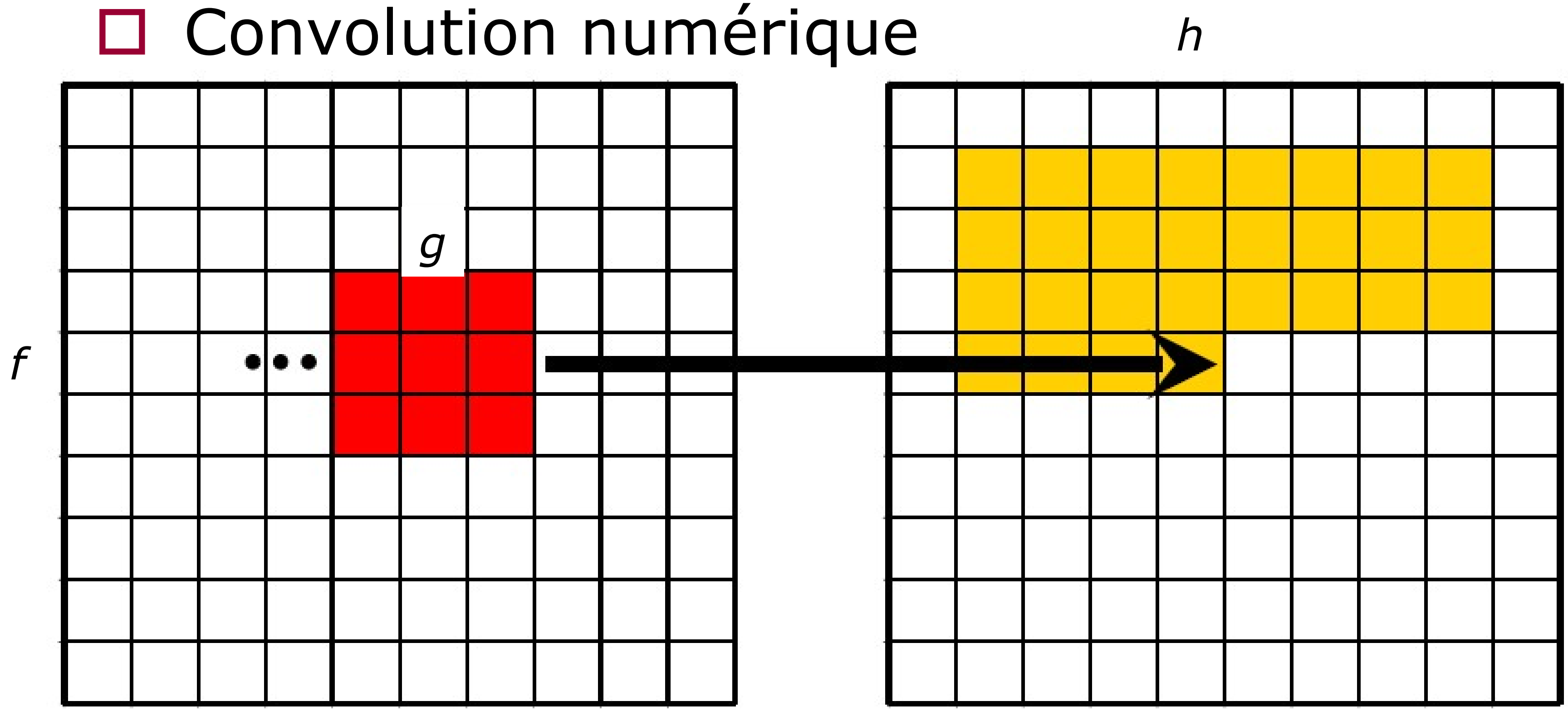
Traitements locaux

- Convolution numérique



Traitements locaux

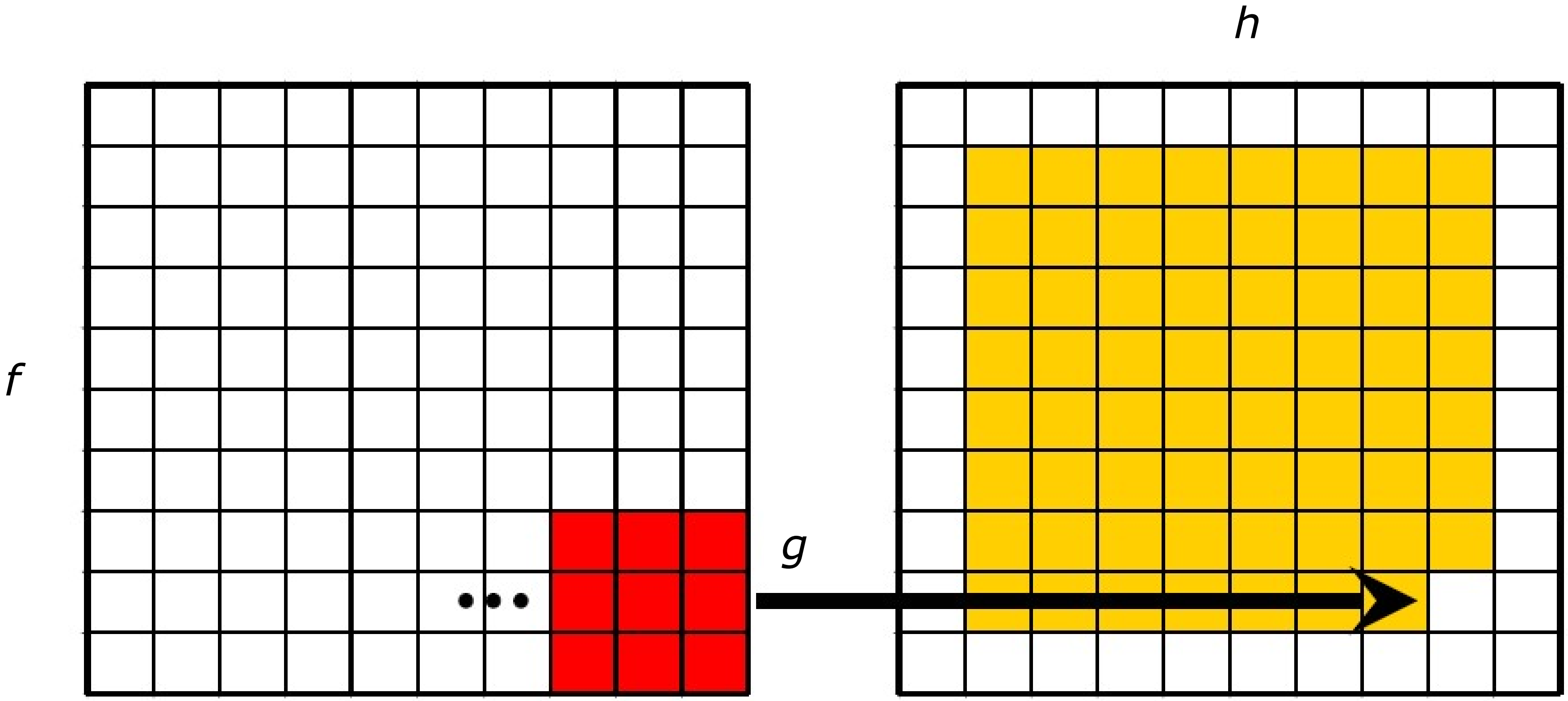
- Convolution numérique



H(x,y) =f(x-1,y-1)*g(0,0) + f(x,y-1)*g(1,0)+f(x+1,y-1)*g(2,0)
+ f(x-1,y)*g(0,1) + f(x,y)*g(1,1)+f(x+1,y)*g(2,1)
+ f(x-1,y+1)*g(0,2) + f(x,y+1)*g(1,2)+f(x+1,y+1)*g(2,2)

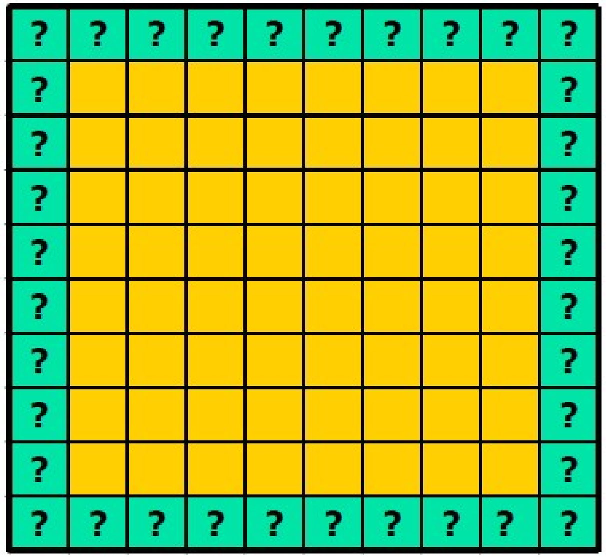
Traitements locaux

- Convolution numérique



Traitements locaux

- Convolution numérique
  - Problème : Que faire avec les bords de l'image ?
    - Mettre à zéro (0)
    - Convolution partielle sur une portion du noyau
    - … (pas de solution miracle)



Traitements locaux

- Convolution numérique
  - Traitement appliqué dépend du choix
    - du voisinage
      - Forme du masque ou noyau du filtre
      - Taille p du masque ou noyau du filtre
    - Des coefficients a_{ij} qui définissent le masque

Traitements locaux

- Convolution numérique
  - Traitement appliqué dépend du choix
    - du voisinage
      - Forme du masque ou noyau du filtre
      - Taille p du masque ou noyau du filtre
    - Des coefficients a_{ij} qui définissent le masque

Traitements locaux

- Famille de filtres
  - Filtres de lissage
    - *éliminent* des éléments *perturbateurs / non significatifs* dans les images numériques
    - Filtres linéaires
      - Filtres moyenneurs
      - Filtres gaussiens
    - Filtres non linéaires
      - Filtre médian
      - Rank value Filters (généralisation du filtre médian)
  - Filtres dérivateurs
    - Opérateurs qui mettent en évidence certaines variations spatiales dans les images

Filtres de Lissage

- Filtres linéaires moyenneurs
  - Remplacer le niveau de gris d'un pixel par la moyenne des niveaux des pixels voisins

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

  - La somme des coefficients est égale à 1 pour conserver la dynamique de l'image

    $\Rightarrow$ On divise le résultat de la convolution par

    La somme des coefficients du masque

Filtres de Lissage

- Filtre linéaire moyenneur
  - Permet de lisser l'image (smoothing)
    - Remplace chaque pixel par la valeur moyenne de ses voisins
    - Réduit le bruit et les détails non-importants
    - Brouille ou rend floue l'image (blur edges)
  - Filtre dont tous les coefficients sont égaux

- Exemple de filtres moyenneurs

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

ou 1/9

1	1	1
1	1	1
1	1	1

3x3

1/25

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

5x5

Filtres de Lissage

- Filtre linéaire moyenneur

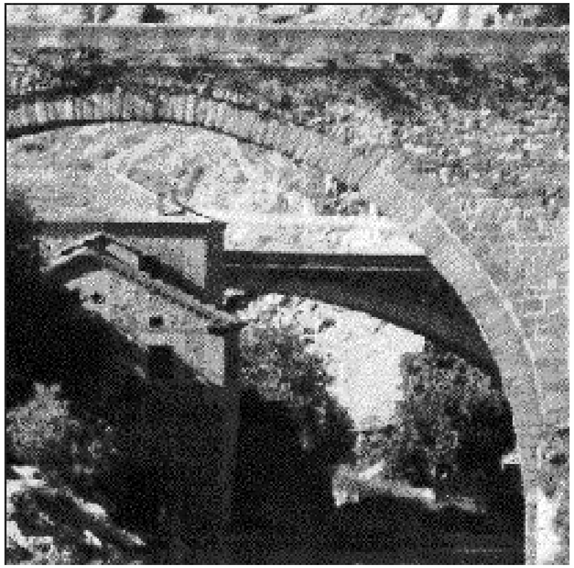


image originale

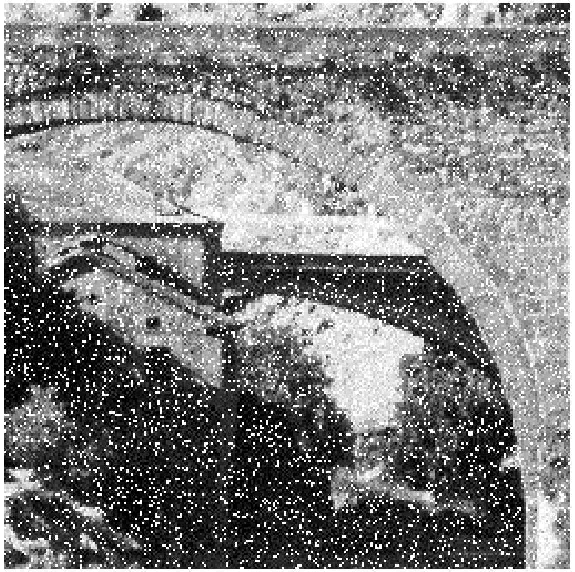


image bruitée

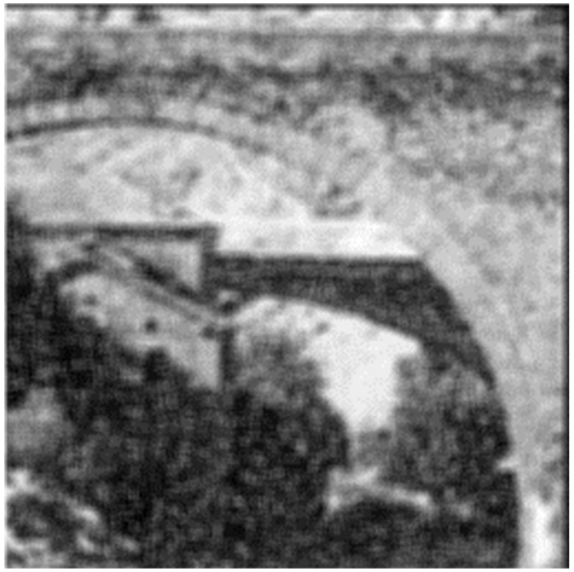
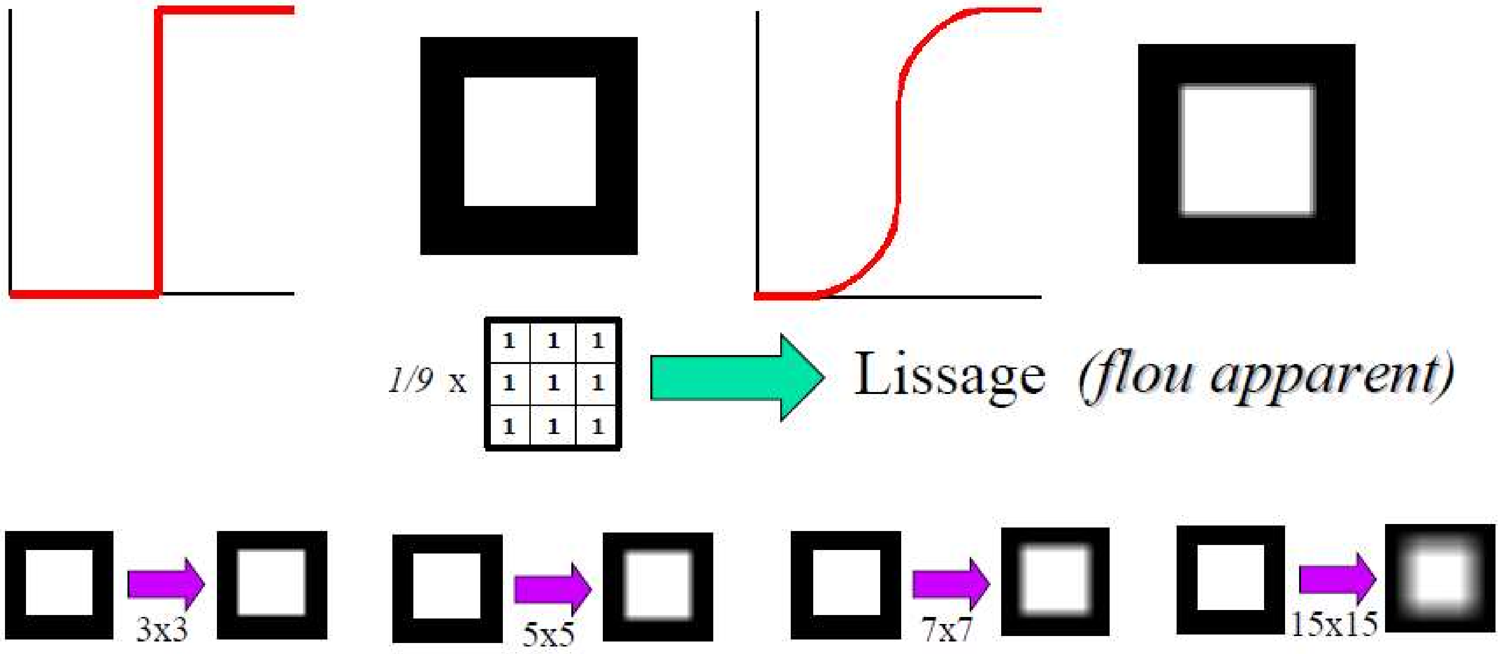


image bruitée filtrée
(taille 5x5)

Filtres de Lissage

- Filtre linéaire moyenneur



Plus le filtre grossit , plus le lissage devient important et plus le flou s’accentue !

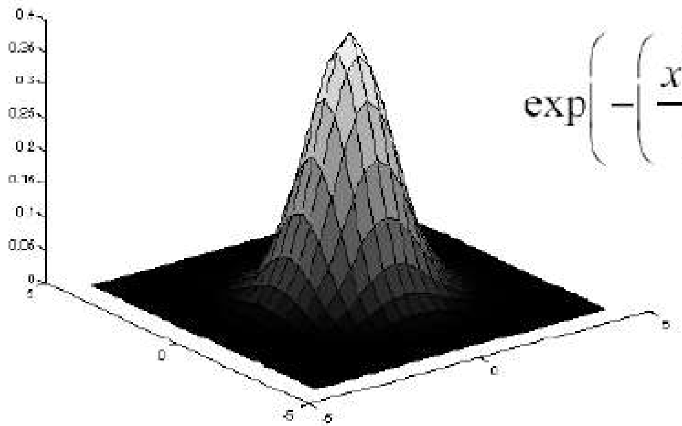
Filtres de Lissage

- Filtre linéaire moyenneur

Original | Moyenne 5x5 | Moyenne 11x11

Filtres de Lissage

- Filtre gaussien



Fonction gaussienne en 3D

$$\exp\left(-\left(\frac{x^2+y^2}{2\sigma^2}\right)\right)$$

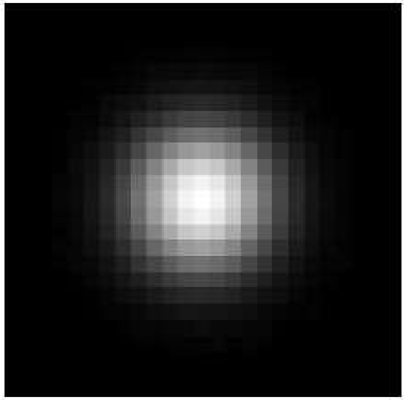
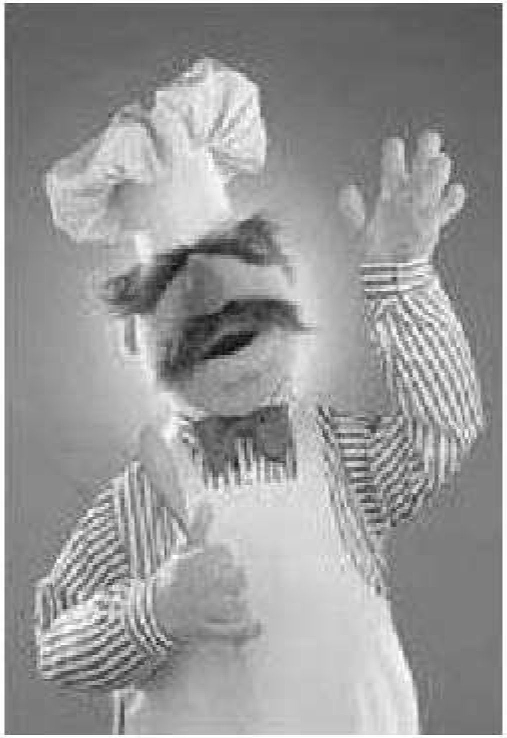


Image d'une gaussienne

$$\frac{1}{98}\times\begin{bmatrix}1 & 2 & 3 & 2 & 1\\ 2 & 6 & 8 & 6 & 2\\ 3 & 8 & 10 & 8 & 3\\ 2 & 6 & 8 & 6 & 2\\ 1 & 2 & 3 & 2 & 1\end{bmatrix}$$

Filtres de Lissage

- Filtre gaussien



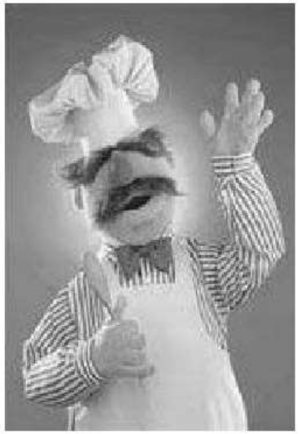
Original

Gauss 5x5

Gauss 11x11

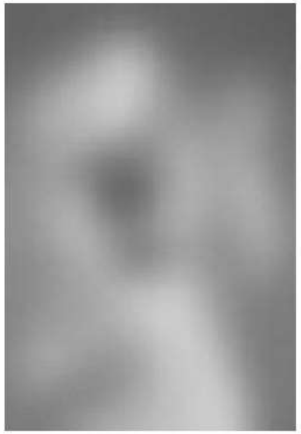
Filtres de Lissage

☐ Filtre moyenneur / Filtre gaussien

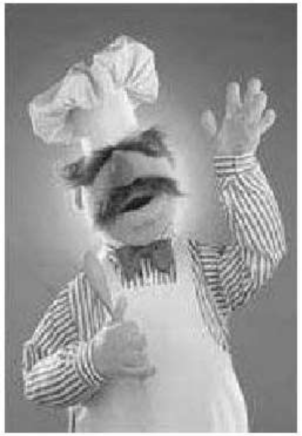


Original

Moyenne 5x5



Moyenne 11x11



Original

Gauss 5x5

Gauss 11x11

Le filtre gaussien donne un meilleur lissage et une meilleure réduction du bruit que le filtre moyenneur

Filtres de Lissage

- Filtres non linéaires
  - Filtre médian
    - Non linéaire donc ne peut pas s'implémenter comme un produit de convolution
    - ⇒ On remplace la valeur d'un pixel par la valeur médiane dans son voisinage NxN

30	10	20
10	250	20
25	10	30

Moyenne = 45

bruit

10, 10, 10, 20, **20**, 25, 30, 30, 250

médiane

Filtres de Lissage

- Filtres non linéaires
  - Filtre médian

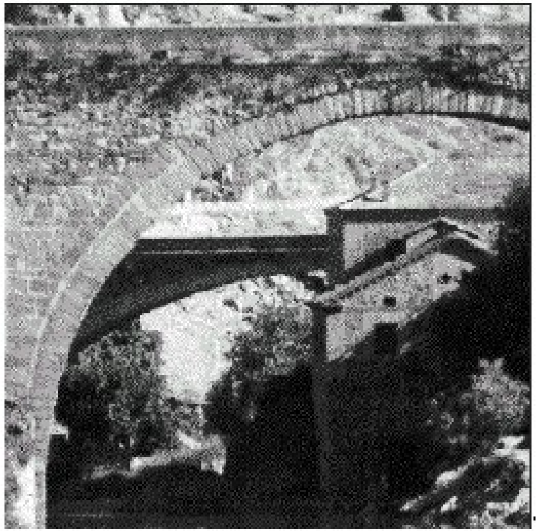


Image originale

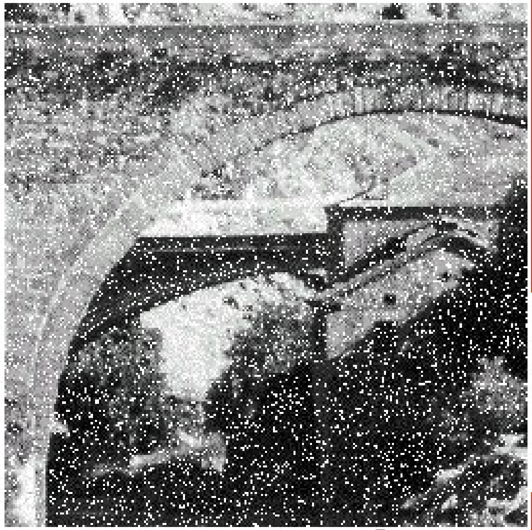
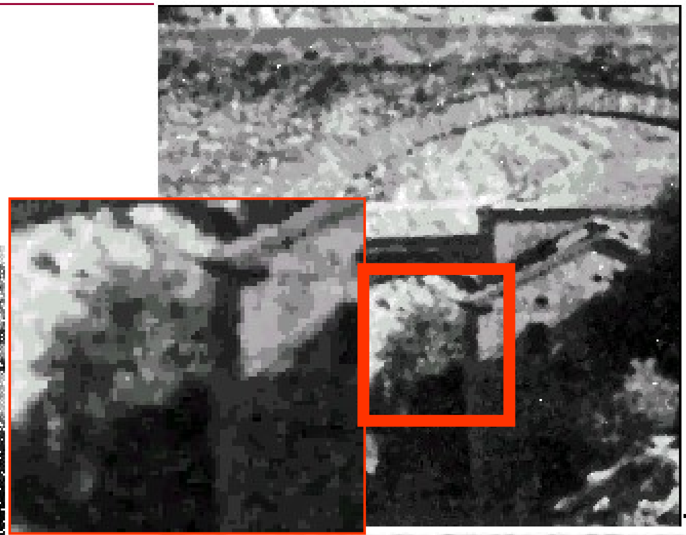


Image bruitée

Image bruitée + filtre médian



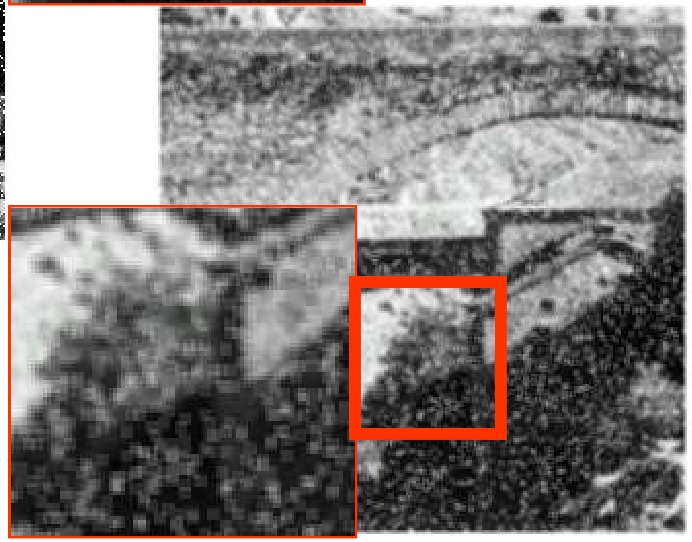
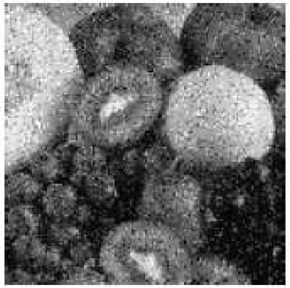


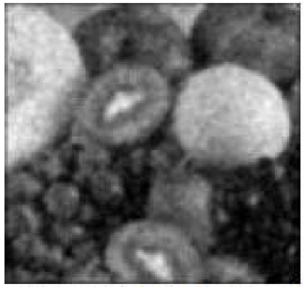
Image bruitée + filtre moyenneur

Filtres de Lissage

- Filtres non linéaires
  - Filtre médian



Bruit "poivre et sel"



3 X 3 Moyenne

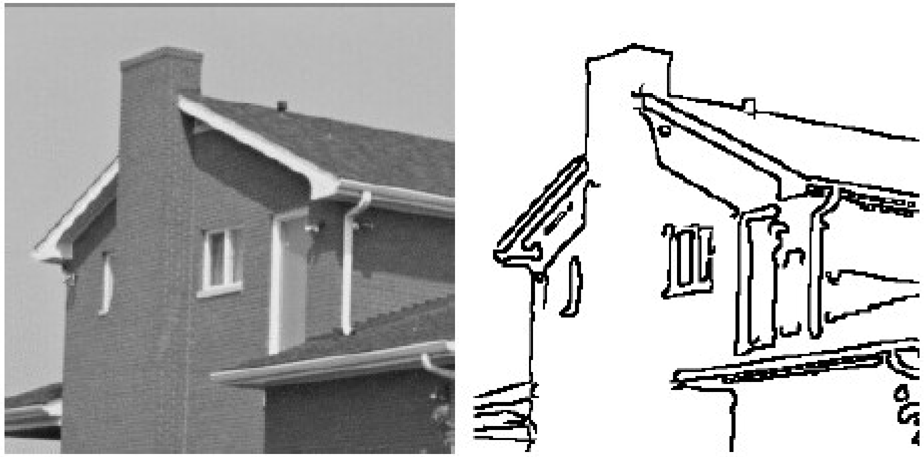
5 X 5 Moyenne

7 X 7 Moyenne

Filtre médian

Traitements Locaux

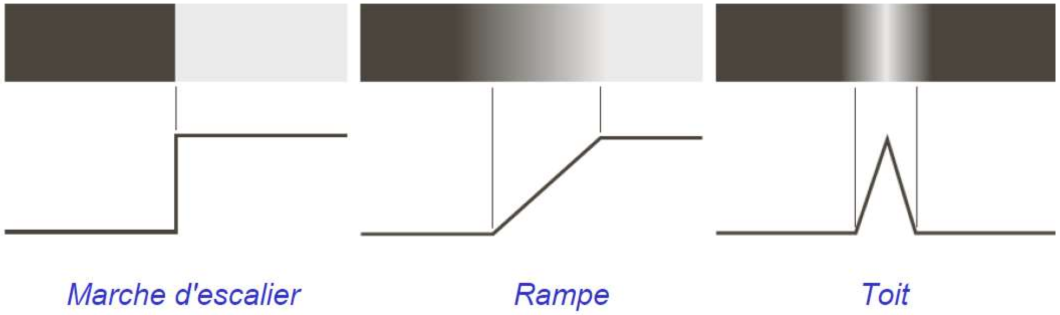
- Filtres dérivateurs
  - Opérateurs qui mettent en évidence certaines variations spatiales dans les images (changements d'intensité)



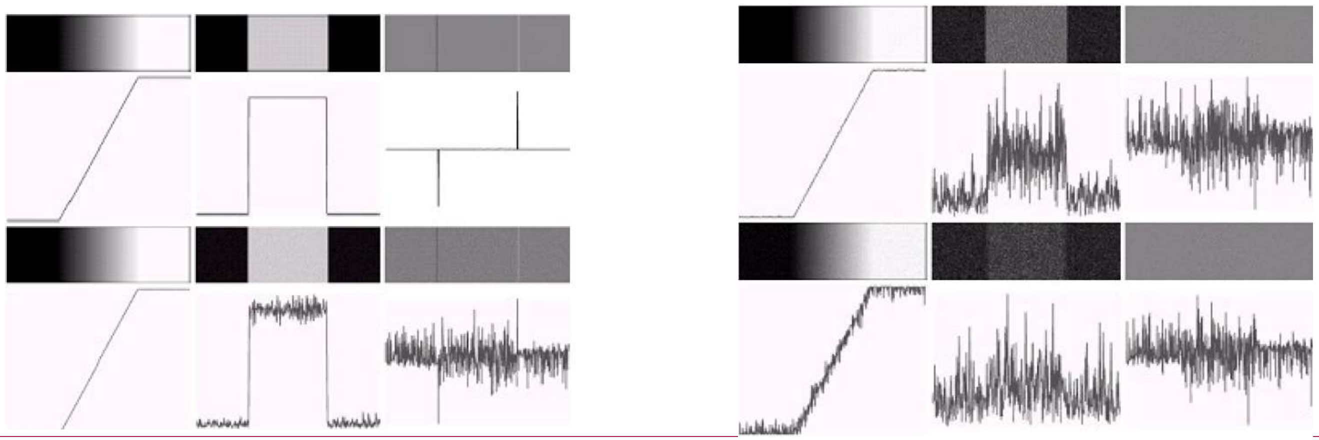
Contours = simplification de l'image utile

Filtres Dérivateurs

- Contours
  - Discontinuités locales de niveaux de gris

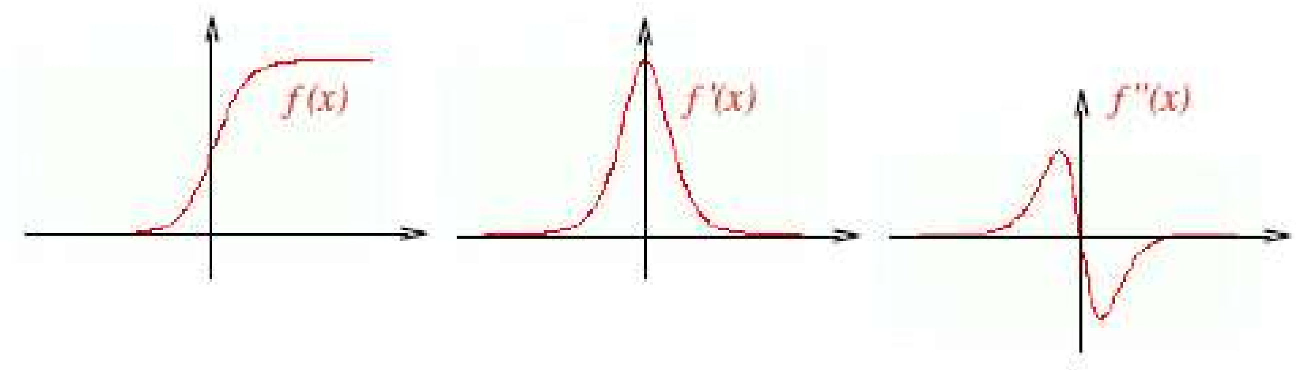


Contour + bruit



Filtres Dérivateurs

- Contours
  - Maxima locaux de la dérivée première
  - Passage par zéro de la dérivée seconde



Filtres Dérivateurs

- Image = signal bidimensionnel
  - Cas continu
    - dérivée 1ère est représentée par un vecteur gradient qui donne en chaque pixel
      - l 'amplitude du contour (norme du gradient)
      - l 'orientation θ du contour (perpendiculaire à l 'orientation du gradient)

$$I(x,y) = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right)$$

avec

$$\left|\vec{\nabla}\right| = \sqrt{\left(\frac{\partial \boldsymbol{I}}{\partial \boldsymbol{x}}\right)^2 + \left(\frac{\partial \boldsymbol{I}}{\partial \boldsymbol{y}}\right)^2}$$

$$\theta = \frac{\pi}{\mathbf{2}} + \mathbf{arctan}\left(\frac{\partial \boldsymbol{I}}{\partial \boldsymbol{x}}, \frac{\partial \boldsymbol{I}}{\partial \boldsymbol{y}}\right)$$

Filtres Dérivateurs

- Filtrage par convolution
  - Approximation simple des dérivées directionnelles par différences finies calculées par convolution avec des noyaux simples

    Ex:

$\begin{bmatrix} -1 & 1 \end{bmatrix}$ pour l'approximation de $\dfrac{\partial f}{\partial x}$

$\begin{bmatrix} -1 \\ 1 \end{bmatrix}$ $\dfrac{\partial f}{\partial y}$

On utilise plus souvent $\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$ et $\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$

Filtres Dérivateurs

- Filtrage par convolution
  - Ces approximations sont sensibles au bruit
  - $\Rightarrow$ on les combine en général avec un filtre lisseur dans la direction orthogonale à celle de dérivation

$[1\,2\,1]$ ***Combiné avec*** $\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$ ***donne*** $h_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$

$\begin{bmatrix} \mathbf{1} \\ \mathbf{2} \\ \mathbf{1} \end{bmatrix}$ ***Combiné avec*** $[-1\,0\,1]$ ***donne*** $h_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$

$$f_x[i,j] = (f * h_x)[i,j]$$
$$f_y[i,j] = (f * h_y)[i,j]$$

Filtres de Sobel

Filtres Dérivateurs

- Filtrage par convolution
  - Quelques filtres pour approximer le gradient
    - Opérateur de Prewitt

$$h_x = \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix} \quad et \quad h_y = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$$

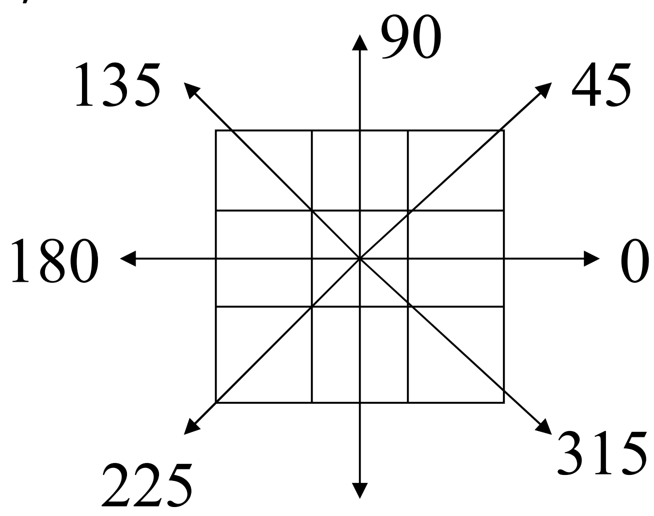
    - Opérateur de Sobel

$$h_x = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \quad et \quad h_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

Filtres Dérivateurs

- Opérateur de Kirsh
  - associe un masque à chaque direction détectable dans le voisinage du pixel considéré
    - voisinage 3x3

      => 8 directions possibles: 0,45, 90, 135, 180, 225, 270, 315



Filtres Dérivateurs

- Opérateur de Kirsh

$$H_0 = \begin{pmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix} \xRightarrow{45°} H_1 = \begin{pmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix} \xRightarrow{90°} H_2 = \begin{pmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{pmatrix}$$

$$\xRightarrow{135°} H_3 = \begin{pmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{pmatrix} \xRightarrow{180°} H_4 = \begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{pmatrix} \xRightarrow{225°} H_5 = \begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{pmatrix}$$

$$\xRightarrow{270°} H_6 = \begin{pmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{pmatrix} \xRightarrow{315°} H_7 = \begin{pmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{pmatrix}$$

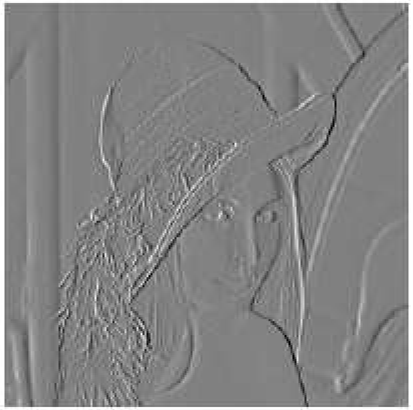
Filtres Dérivateurs

- Opérateur de Kirsh
  - Gradient retenu
    - celui dont l 'amplitude est maximale
    - Orientation retenue = celle correspondant au masque ayant permis d 'obtenir le gradient d 'amplitude maximale
  - Opérateur intéressant
    - prend en compte toutes les directions à partir du point central (pas seulement H et V)
  - Traitement beaucoup plus lent

Filtres Dérivateurs

□ Illustration

Original



Noyau [-1 1]

Noyau [-1 0 1]

Gradient horizontal (Sobel)

Gradient vertical (Sobel)

Module du gradient de Sobel

Filtres Dérivateurs

- Image = signal bidimensionnel
  - Cas continu
    - Dérivée seconde calculée dans la direction du gradient
    - ou approximée par le Laplacien de la fonction intensité

$$\nabla^2 I(x,y) = \left(\frac{\partial \mathrm{I}}{\partial x}\right)^2 + \left(\frac{\partial \mathrm{I}}{\partial y}\right)^2$$

Filtres Dérivateurs

- Approximations discrètes du Laplacien
  - Convolution avec masque L_4 ou L_8

$$L_4 = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} \qquad L_8 = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

$\Rightarrow$ Inconvénient majeur = grande sensibilité au bruit

$\Rightarrow$ utilisation d 'un filtre passe-bas avant

Filtres Dérivateurs

- Laplacien
  - Illustration

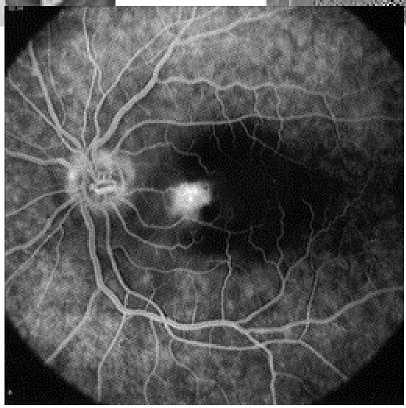


image originale

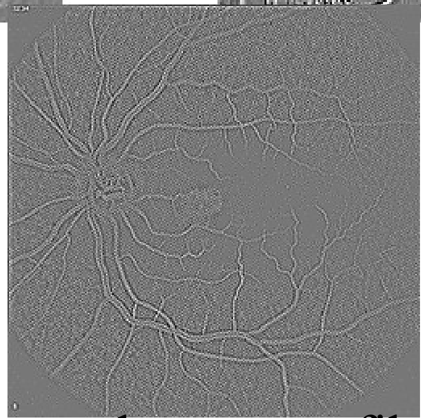


image obtenue par filtrage avec L4