Introduction au Traitement d'Images

Séance 4

Isabelle Debled-Rennesson
debled@loria.fr

Contenu

- 6 cours/TD/TP de 2h
- Points abordés :
  - Codage des images
  - Histogrammes
  - Transformations géométriques sur les images
  - **Filtres**
  - Opérateurs morphologiques
  - **Détection de contours**
  - Segmentation

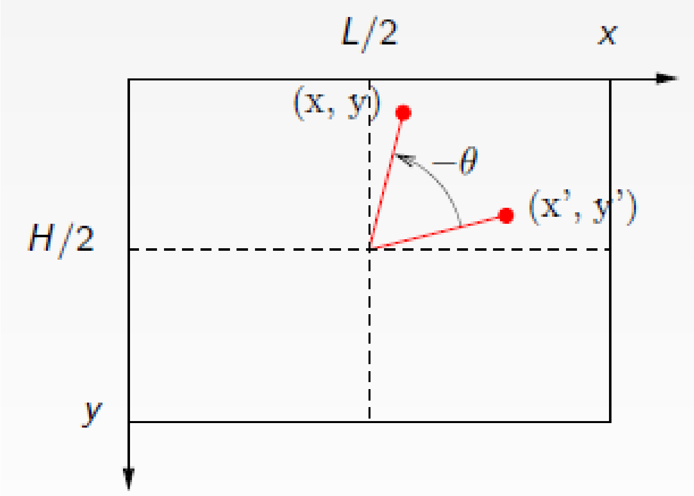
Retour sur les exercices au sujet des transformations géométriques sur les images

Rotation de l'image autour de son centre

- Rotation **horaire** de centre (L/2,H/2)

  $x' = L/2 + (x-L/2)\cos(\theta) + (y-H/2)\sin(\theta)$

  $y' = H/2 + (y-H/2)\cos(\theta) - (x-L/2)\sin(\theta)$



- Recherche de l'antécédent

  $x = L/2 + (x' - L/2)\cos(\theta) - (y' - H/2)\sin(\theta)$

  $y = H/2 + (y' - H/2)\cos(\theta) + (x' - L/2)\sin(\theta)$

Rotation de l'image autour de son centre – question 2 TD/TP S3

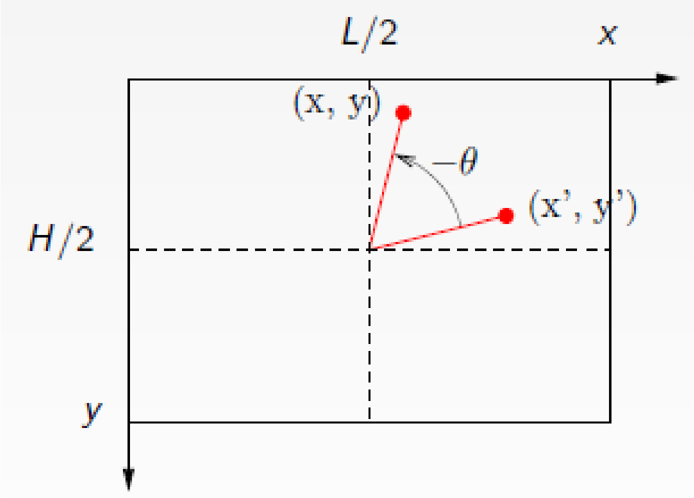
- Rotation de 90 degrés

  θ = 90 degrés

- Recherche de l'antécédent

  $x = L/2 + (x' - L/2)\cos(\theta) - (y' - H/2)\sin(\theta)$

  $y = H/2 + (y' - H/2)\cos(\theta) + (x' - L/2)\sin(\theta)$



$X = -y' + L/2 + H/2$

$Y = x' - L/2 + H/2$

Rotation de l'image autour de son centre – question 2 TD/TP S3

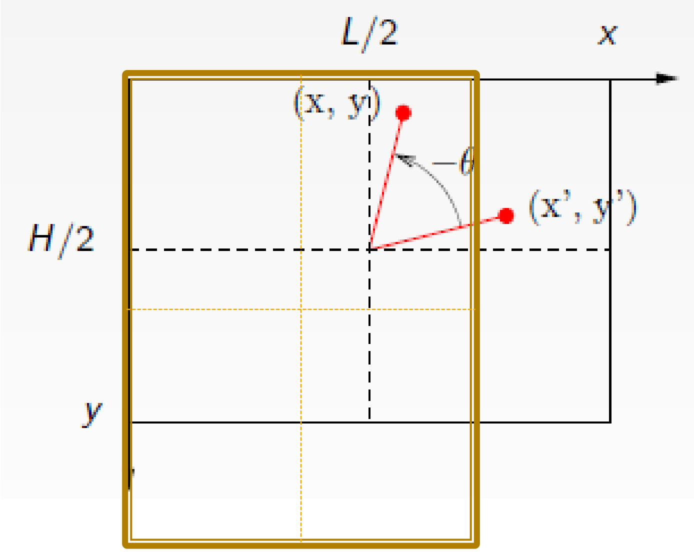
- Rotation de 90 degrés

  θ = 90 degrés

- Recherche de l'antécédent

  $x = L/2 + (x' - L/2)\cos(\theta) - (y' - H/2)\sin(\theta)$

  $y = H/2 + (y' - H/2)\cos(\theta) + (x' - L/2)\sin(\theta)$



$X = -y' + L/2 + H/2 + (L-H)/2$

$Y = x' - L/2 + H/2 + (L-H)/2$

Mais attention décalage de (L-H)/2 des centres des deux images

Rotation de l'image autour de son centre – question 2 TD/TP S3

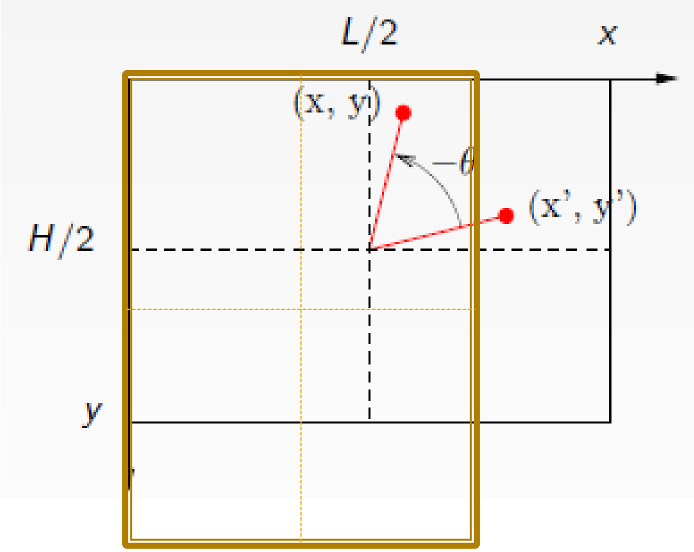
- Rotation de 90 degrés

  θ = 90 degrés

- Recherche de l'antécédent

  $x = L/2 + (x' - L/2)\cos(\theta) - (y' - H/2)\sin(\theta)$

  $y = H/2 + (y' - H/2)\cos(\theta) + (x' - L/2)\sin(\theta)$



$X = -y' + L/2 + H/2 + (L-H)/2$

$Y = x' - L/2 + H/2 + (L-H)/2$

$X = -y' + L$

$Y = x'$

Mais attention décalage de (L-H)/2 des centres des deux images

Changement d'échelle – questions 3 et 4 TD/TP S3

- Changement d'échelle = homothétie de centre l'origine

- Soient Sx et Sy les facteurs d'échelle suivant chaque axe (agrandissement ou réduction)

  x' = Sx *x   **x = 1/Sx *x'**

  y' = Sy *y   **y = 1/Sy *y'**

- Algorithme

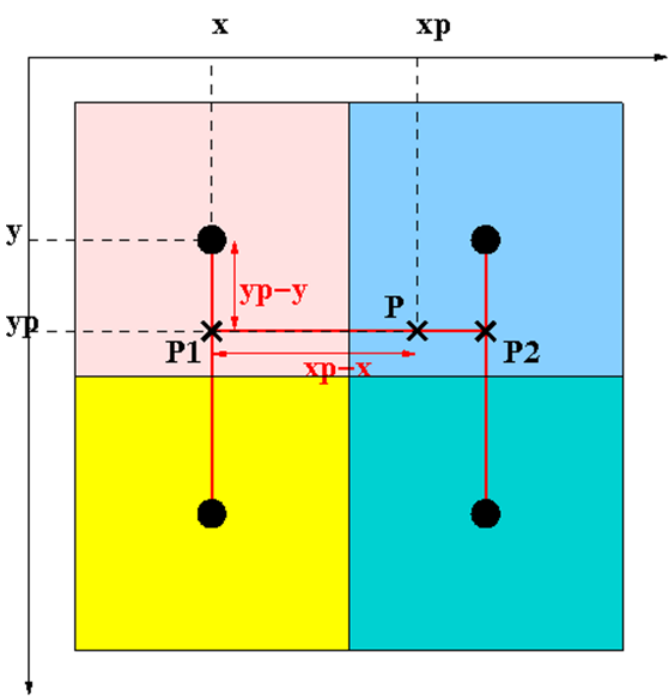
```
W' = Sx*W
H' = Sy*H
Créer l'image résultat R de taill
for(y=0; y<H'; y++)
   for(x=0; x<W'; x++)
          R(x, y) = I(x/Sx, y/Sy)
```

Problème : **`x/Sx et y/Sy` `ne sont pas forcément` `des entiers`**

Attention l'approximation peut engendrer un entier aux limites du tableau → à tester

Question 4 S3 -Interpolation bilinéaire

- Permet d'atténuer l'aliasing
- L'**interpolation bilinéaire** = méthode d'interpolation pour les fonctions de 2 variables sur une grille régulière.
  - permet de calculer la valeur d'une fonction en un point quelconque, à partir de ses 2 plus proches voisins dans chaque direction.



Autre méthode pour calculer I_P : succession de deux interpolations linéaires en considérant les points P1 et P2

$$I_{P1} = I(x, y) + (y_P - y)(\, I(x\,, y + 1) - I(x\,, y)\,)$$
$$I_{P2} = I(x + 1\,, y) + (y_P - y)(\, I(x + 1\,, y + 1) - I(\, x + 1\,, y)\,)$$

$$I_P = I_{P1} + (x_P - x)(\, I_{P2} - I_{P1})$$

Et si les 4 pixels voisins n'existent pas ?

Filtrage des images

Sources : cours d'Anne Vialard, Benoit Naegel, Bertrand Kerautret, Stéphane Paris

Filtrage – Filtres linéaires

- La classe des **filtres linéaires** est utilisée couramment en traitement du signal
- Un opérateur f de traitement d'images est dit linéaire si:
  - **$f(I+J)=f(I)+f(J)$**
  - Autrement dit, filtrer la somme arithmétique de deux images revient au même que de filtrer les deux images séparément, puis effectuer la somme arithmétique des résultats

Filtrage – Filtres linéaires - Convolution

- Fondement du filtrage linéaire

- Soit deux images I et M; la convolution d'une image I par l'image M de dimension (2p+1)x(2p+1) est définie par:

$$I*M(x,y)=\sum_{k=0->2p}\sum_{t=0->2p}I(x+k-p,y+t-p).M(k,t)$$

- L'image M est appelée "**noyau de convolution**" ou "**masque de convolution**"

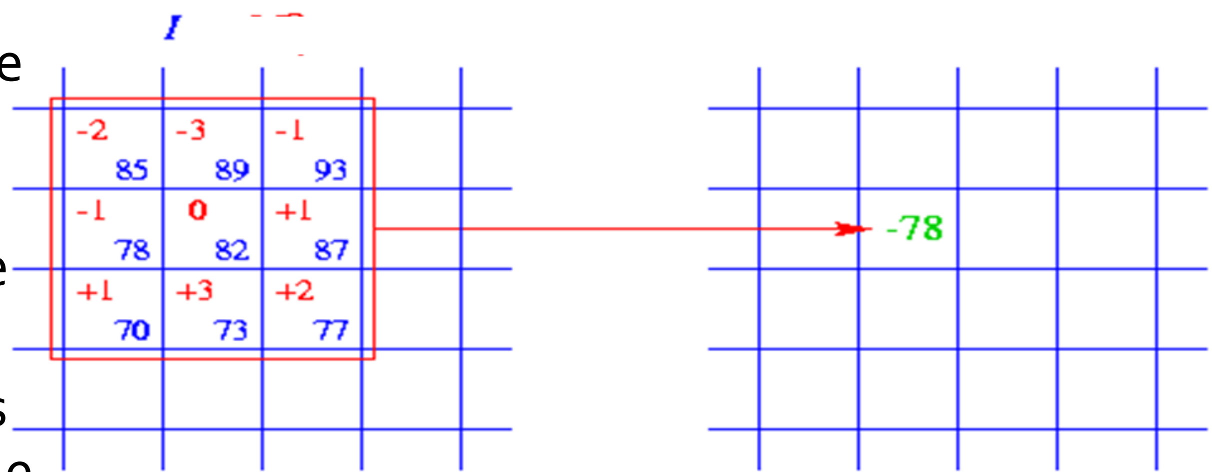
- Théoriquement, l'opération est réversible (opération de ***déconvolution***)

Filtrage – Filtres linéaires - Convolution

- Algorithme pratique:
  - Entrée: Image I
  - Sortie: Image R

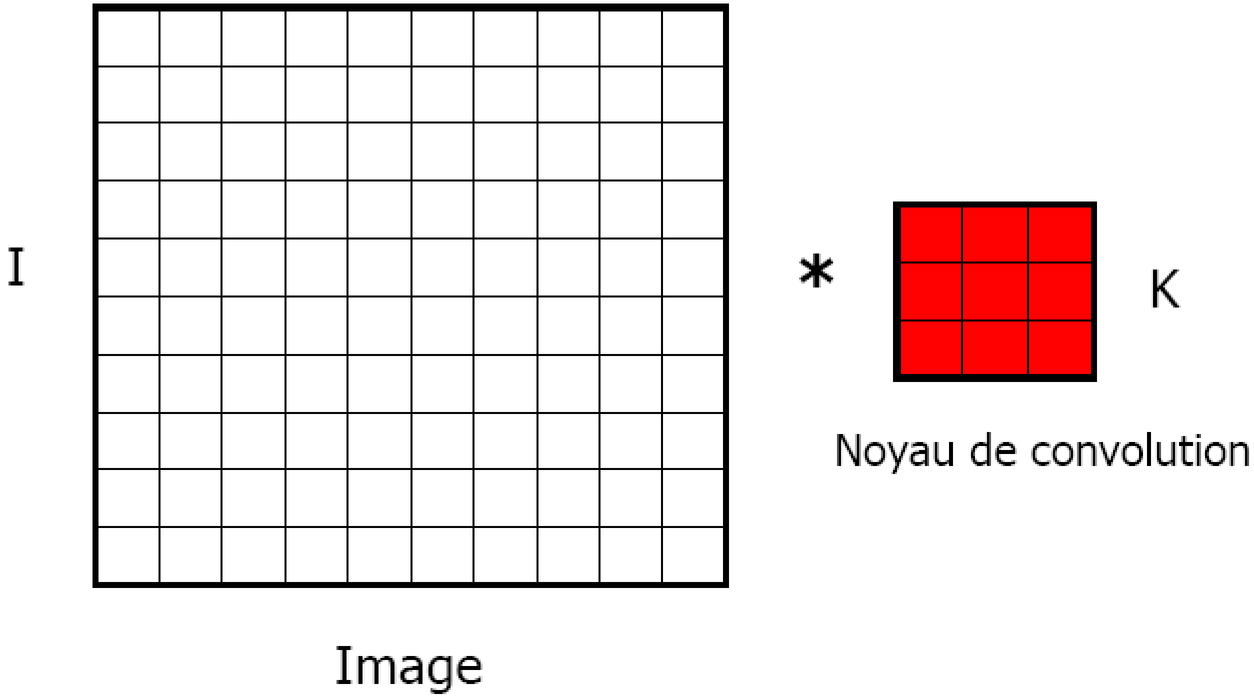
-2	-3	-1
-1	0	+1
+1	+3	+2

  - Pour tous les points p de I:
    - On translate le masque au point p
    - Pour tous les points r du masque , on calcule le produit M(r)*I(r)
    - On additionne tous les résultats et on stocke le résultat dans R(p)



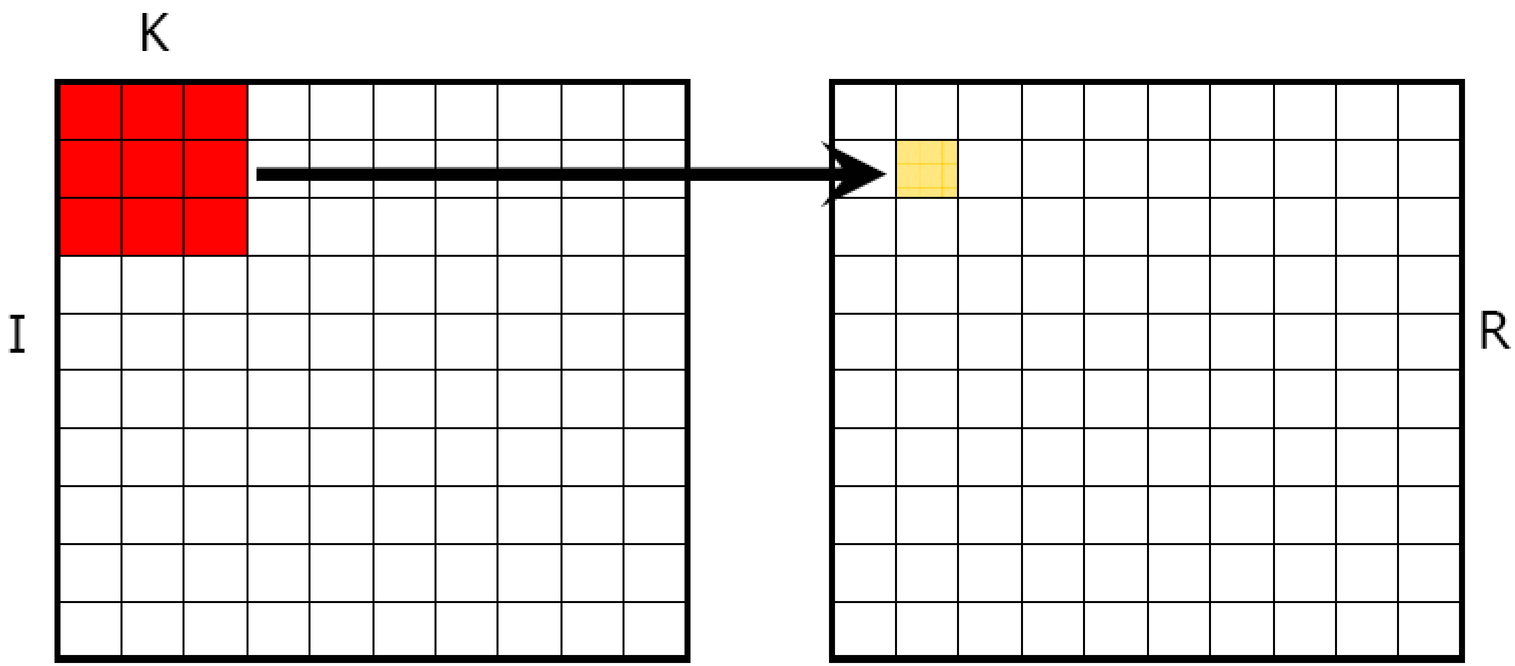
$$-2x85 -3x89 -1x93 -1x78 +0x82 +1x87 +1x70 + 3x73 +2x77 = -78$$

Convolution – détails 1/5



Sources : Anne Vialard

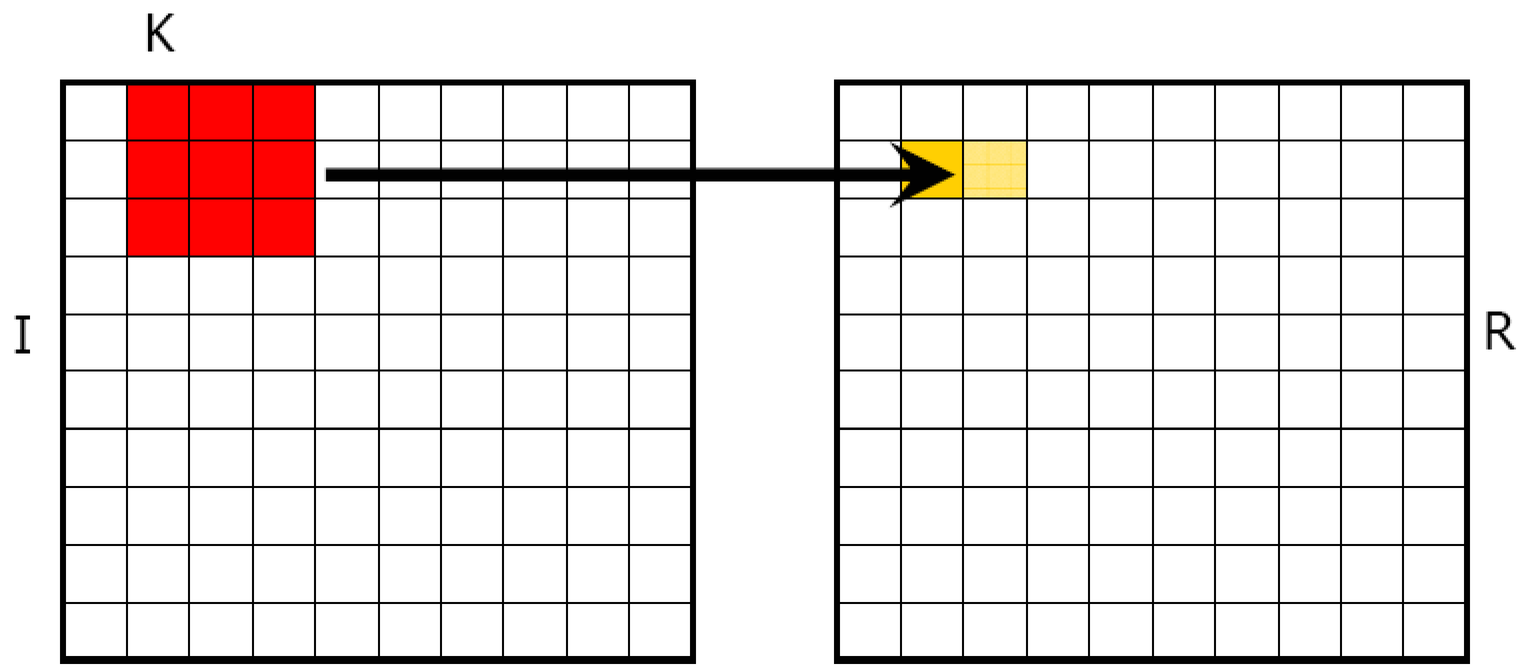
Convolution – détails 2/5



$$
\begin{aligned}
R(1,1) = {} & I(0,0)\,K(0,0) + I(1,0)\,K(1,0) + I(2,0)\,K(2,0) \\
& + I(0,1)\,K(0,1) + I(1,1)\,K(1,1) + I(2,1)\,K(2,1) \\
& + I(0,2)\,K(0,2) + I(1,2)\,K(1,2) + I(2,2)\,K(2,2)
\end{aligned}
$$

Sources : Anne Vialard

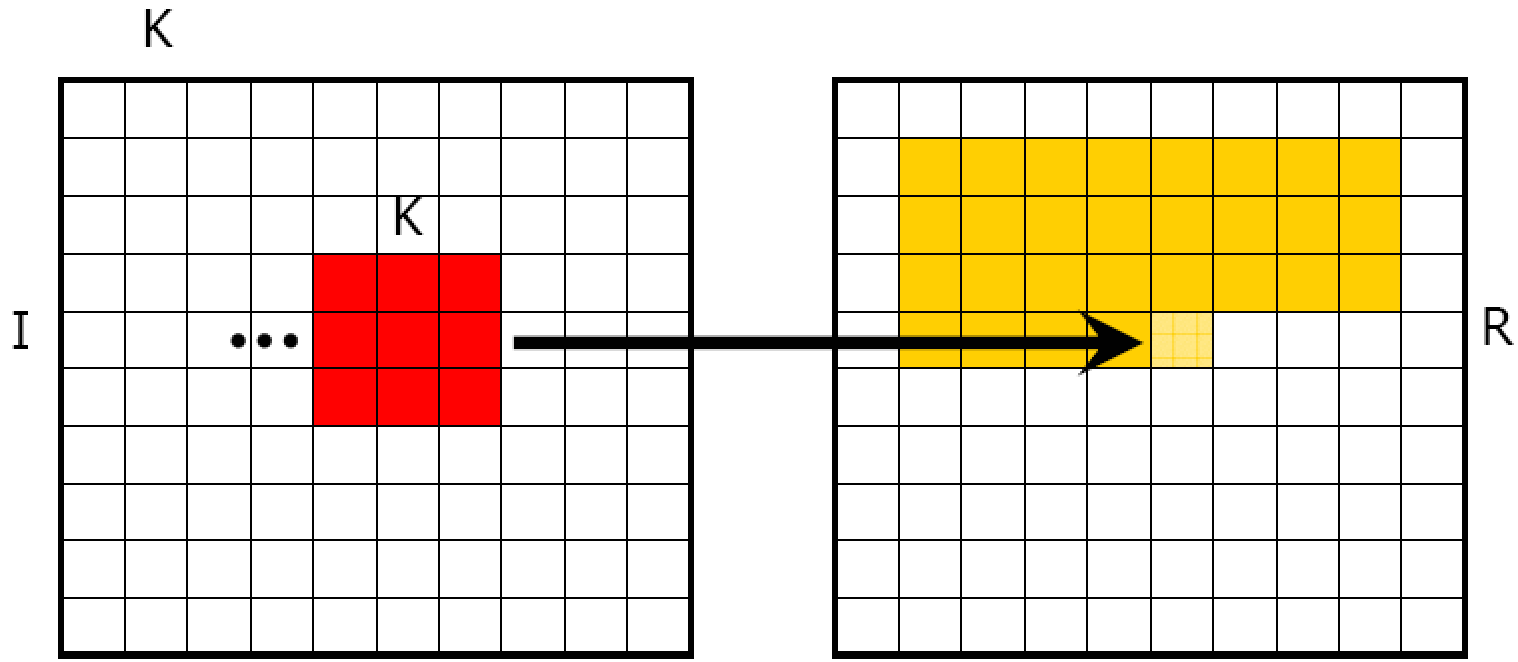
Convolution – détails 3/5



$$
\begin{aligned}
R(2,1) = {} & I(1,0)\,K(0,0) + I(2,0)\,K(1,0) + I(3,0)\,K(2,0) \\
& + I(1,1)\,K(0,1) + I(2,1)\,K(1,1) + I(3,1)\,K(2,1) \\
& + I(1,2)\,K(0,2) + I(2,2)\,K(1,2) + I(3,2)\,K(2,2)
\end{aligned}
$$

Sources : Anne Vialard

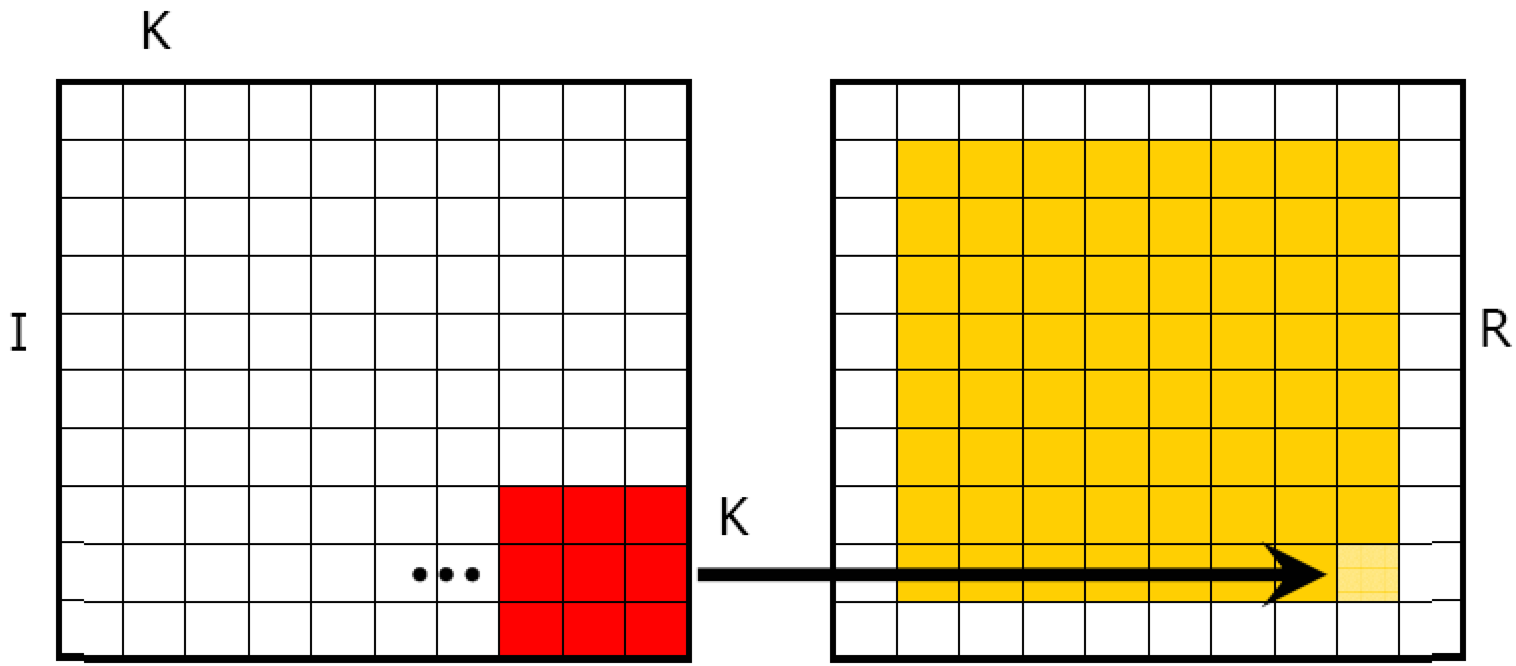
Convolution – détails 4/5



$$
\begin{aligned}
R(x,y) = {} & I(x-1,y-1)\ K(0,0) + I(x,\ y-1)\ K(1,0) + I(x+1,\ y-1)\ K(2,0) \\
& + I(x-1,y)\ K(0,1) + I(x,y)\ K(1,1) + I(x+1,y)\ K(2,1) \\
& + I(x-1,y+1)\ K(0,2) + I(x,y+1)\ K(1,2) + I(x+1,y+1)\ K(2,2)
\end{aligned}
$$

Sources : Anne Vialard

Convolution – détails 5/5

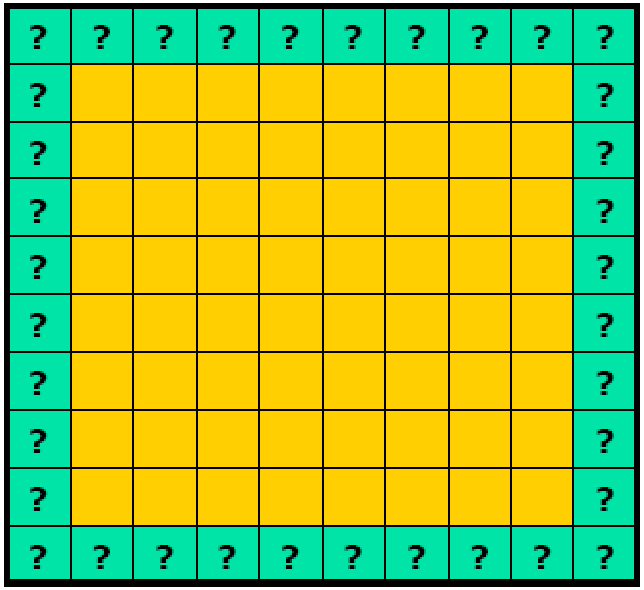


$$
\begin{aligned}
R(N-2,M-2) = {} & I(N-3,M-3)\,K(0,0) + I(N-2,M-3)\,K(0,1) + I(N-1,M-3)\,K(0,3) \\
& + I(N-3,M-2)\,K(1,0) + I(N-2,M-2)\,K(1,1) + I(N-1,M-2)\,K(1,2) \\
& + I(N-3,M-1)\,K(2,0) + I(N-2,M-1\,)K(2,1) + I(N-1,M-1)\,K(2,2)
\end{aligned}
$$

Sources : Anne Vialard

Filtrage – Filtres linéaires - Convolution

- Problème : Que faire avec les bords de l'image ?
  - Mettre à zéro
  - Convolution partielle
    - Sur une portion du noyau
  - Compléter les valeurs manquantes en construisant le miroir de l'image
  - …



Familles de filtres

- Filtre passe-bas
  - atténue le bruit et les détails
- Filtre passe-haut
  - accentue les détails et les contours

Sources : Anne Vialard

Filtrage – Filtres linéaires – Convolution - masque moyenneur

- Exemple de masque de convolution: **le masque moyenneur**

1	1	1
1	1	1
1	1	1

R(i,j)=1/9(I(i-1,j-1)+I(i,j-1)+I(i+1,j-1)
+I(i-1,j)+I(i,j)+I(i+1,j)
+I(i-1,j+1)+I(i,j+1)+I(i+1,j+1))

- Pour ne pas introduire de biais dans l'image, on normalise en divisant par la somme des poids (ici 9)

Filtrage – Filtres linéaires - Convolution - masque moyenneur

- Exemple de masque de convolution: **le masque moyenneur**
  - Le filtre moyenneur est un filtre *passe-bas car il conserve les basses fréquences, mais supprime les hautes fréquences*

Masque moyenneur - exemple

- Plus le filtre grossit, plus le lissage devient important

Original

Moyenne 5x5

Moyenne 11x11

Sources : Anne Vialard

Filtrage – Filtres linéaires – Convolution – filtre passe-haut

- Exemple de filtre passe-haut

-1	-1	-1
-1	16	-1
-1	-1	-1

$$R(i,j)=1/24(-I(i-1,j-1)-I(i,j-1)-I(i+1,j-1)$$
$$-I(i-1,j)+16*I(i,j)-I(i+1,j)$$
$$-I(i-1,j+1)-I(i,j+1)-I(i+1,j+1))$$

- Pour ne pas introduire de biais dans l'image, on normalise en divisant par la somme des poids (ici 24)

Filtrage – Filtres linéaires – Convolution – filtre passe-haut

- Exemple de filtre passe-haut
  - Améliore la sensation de netteté
  - Mais renforce le bruit

Filtrage – Filtres linéaires – Convolution – filtre passe-haut

- Exemple de filtre passe-haut
  - Améliore la sensation de netteté
  - Mais renforce le bruit

Après quelques itérations ...

Filtrage – Filtres linéaires – Convolution – filtre passe-haut

- Exemple de filtre passe-haut
  - Améliore la sensation de netteté
  - Mais renforce le bruit

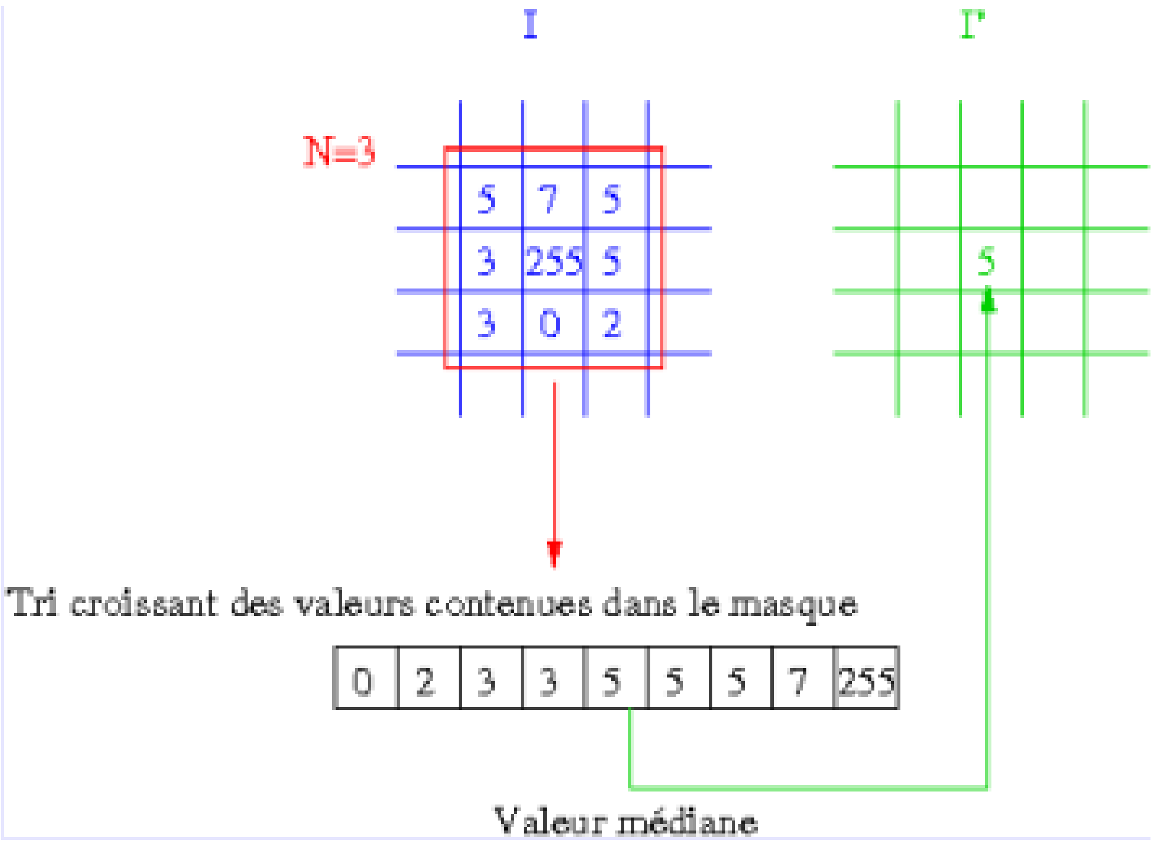
Quelques autres itérations …

Filtrage – Filtres non linéaires

- Résultat d'un filtre non linéaire irréversible
- Avantage : permet de s'adapter à la nature du pixel considéré (pixel de bruit ou pixel utile)
- Suppression totale du bruit en théorie

Filtrage – Filtres non linéaires
Filtre médian

- Masque de taille nxn avec n impair
- Algorithme
  - Entrée : I
  - Sortie : R
  - Pour tout pixel p de I,
    - Translater le masque en p
    - Trier les pixels voisins selon l'ordre croissant de leurs niveaux de gris
    - Affecter la valeur **médiane** dans R(p)

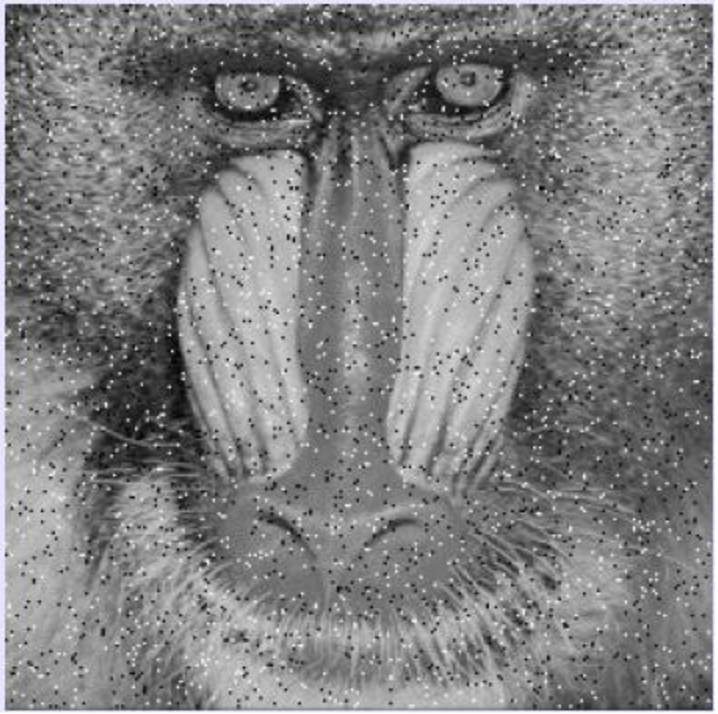


[Quelques notions sur le bruit - 1

- Bruit
  - Phénomène parasitaire généralement aléatoire provenant de phénomènes divers : transmission, éclairage, …
  - Bruit additif : impulsionnel et gaussien
  - Bruit multiplicatif

Quelques notions sur le bruit - 2

- Bruit poivre et sel
  - Ne prend que 2 valeurs : 0 et M (niveau de gris maximum représentable de l'image)
  - Présence ou non de bruit en un point aléatoire
    - Pour une image de dimension wxh en **poivre et sel de p%,** il suffit de colorier whp/2 pixels en noir et whp/2 pixels en blanc



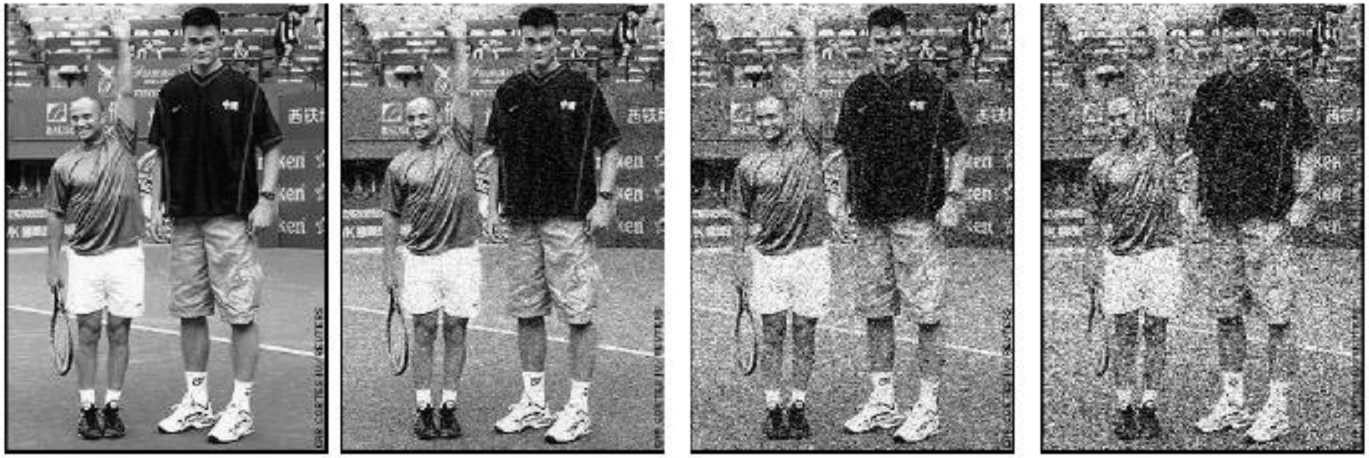
Quelques notions sur le bruit - 3

Sources : http://www.trop.uha.fr/master/downloads/7sourcesdedegradations.pdf

- Bruit gaussien
  - Définition. Loi de distribution Gaussienne de variance σ et moyenne μ :

    Le bruit Gaussien est obtenu en ajoutant à chaque pixel une valeur aléatoire suivant une loi de probabilité Gaussienne.

$$G_{\sigma,\mu}(s) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(s-\mu)^2}{2\sigma^2}}$$



Originale | σ = 20 | σ = 40 | σ = 60

Cf. http://fr.wikipedia.org/wiki/Bruit_blanc pour la simulation

Quelques notions sur le bruit – 4]

- Suppression du bruit
  - Le bruit de type gaussien est plus difficile à supprimer
  - Le bruit généré par les appareils d'acquisition se rapproche généralement du bruit gaussien
  - Nécessité de concevoir des algorithmes de suppression de bruit efficaces, de très nombreuses techniques existent ….

Suppression du bruit - exemples


Bruit poivre et sel


Filtre médian N=3

Suppression du bruit - exemples

Bruit poivre et sel

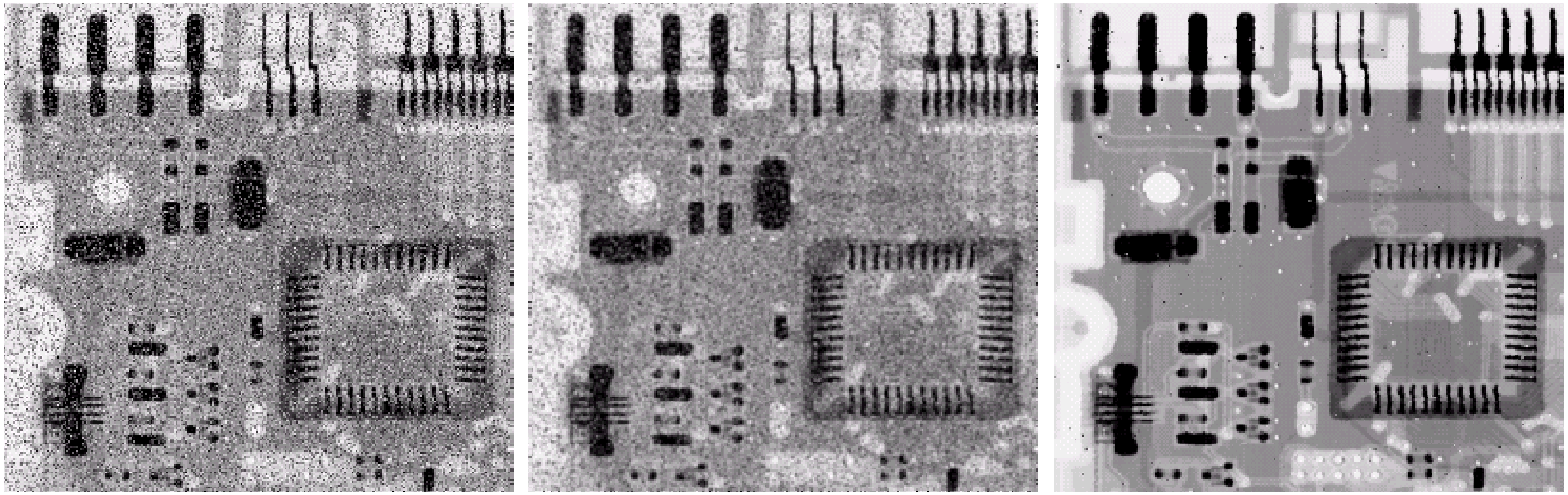
Filtre moyenneur N=5

Suppression du bruit - exemples

Bruit gaussien

Filtre moyenneur N=5

Suppression du bruit - exemples



a b c

FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Filtre non linéaire

- Filtre ***max***

  $R(x,y) = \max\{ I(x+i,y+j) \mid i=-1,0,1,\ j=-1,0,1\}$

  - *Sélectionne les points clairs*
    - Utile pour éliminer le "poivre"

Filtre non linéaire

- Filtre *max*

c. Image with pepper noise; probability of pepper = .04.

d. Result of maximum filtering image (c); mask size = 3 × 3.

Filtre non linéaire

- Filtre ***min***

  $R(x,y) = \min\{\, I(x+i, y+j) \mid i=-1,0,1,\ j=-1,0,1 \}$

  - *Sélectionne les points sombres*
    - Utile pour éliminer le "sel"

Filtre non linéaire

- Filtre *min*

a. Image with salt noise; probability of salt = .04.

b. Result of minimum filtering image (a); mask size = 3 × 3.