

# Filtrage

## Traitement d'Images

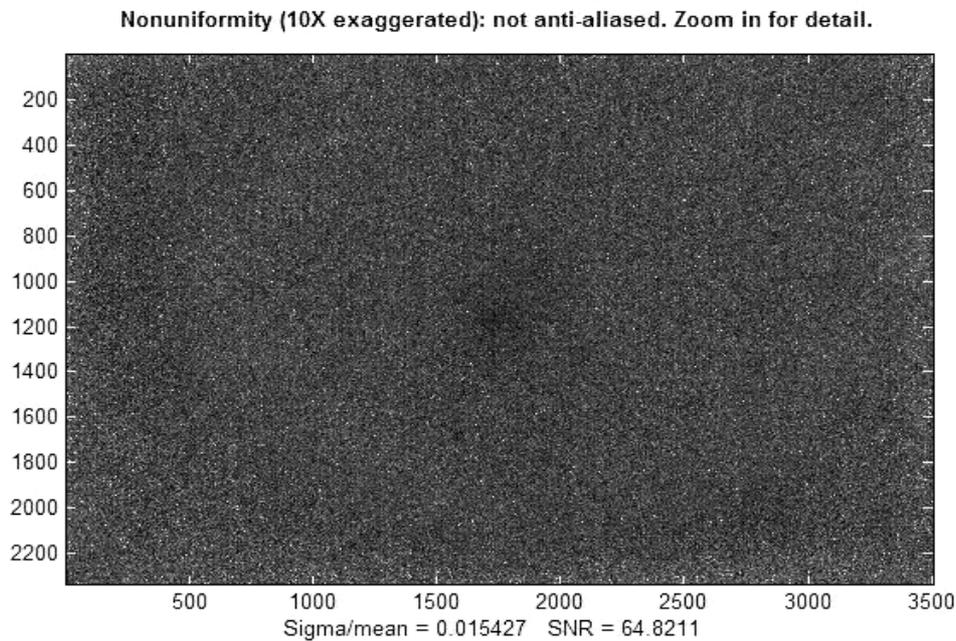
Lionel Lacassagne  
Institut d'Electronique Fondamentale  
lionel.lacassagne@u-psud.fr

Bruit: origine

---

- Origines
  - qualité de l'optique
  - bruit du capteur: bruit électronique liée à l'acquisition et à la conversion
  - bruit dans l'image
    - faible illumination de la scène  
analogie avec photos en extérieur et photo en studio
    - condition météo: transparence de l'air – brume, poussière, etc
    - etc ...
  - etc ...

- Exemple de bruit électronique



3

## Bibliographie

---

- H. Maître, *Le traitement des images*, Hermès éditions.
- J.-P. Cocquerez & S. Philipp, *Analyse d'images : filtrage et segmentation*, Masson éditions.
- S. Bres, J.-M. Jolion & F. Lebourgeois, *Traitement et analyse des images numériques*, Hermès éditions.
- Images
  - copyright (c) Isabelle Bloch, Henri Maitre, Sylvie Mascle Le Hégarat, Wikipedia

4

- Bruit additif
  - valeur ajoutée (ou soustraite) en tout pixel
  - bruit uniforme
  - bruit gaussien
- Bruit impulsionnel
  - un pourcentage de l'image est modifié: remplacement par nouvelle valeur (tirage aléatoire, si valeur < seuil => modification)
  - deux versions
  - bruit poivre et sel: valeur

5

---

### Traitement du Signal et filtrage numérique en 2 slides

---

- Fonction de transfert dans le domaine fréquentiel (transformée en Z):

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^N b_k \cdot z^{-k}}{\sum_{k=0}^M a_k \cdot z^{-k}} \quad H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_N z^{-N}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_M z^{-M}}$$

- FIR = Filtre à Réponse Impulsionnelle Finie (*Finite Impulse Response*)
  - Filtre non récursif: M=0: Y ne dépend que de X
- IIR = Filtre à Réponse Impulsionnelle Infinie (*Infinite Impulse Response*)
  - Filtre récursif: Y dépend de X et de Y(aux instants précédents)

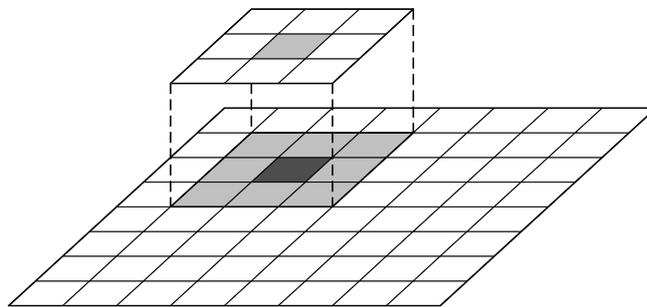
$$y[n] = \sum_{k=0}^N b_k \cdot x[n - k] - \sum_{k=1}^M a_k \cdot y[n - k]$$

- Equation aux différences

$$y[n] = b_0 \cdot x[n] + b_1 \cdot x[n-1] + b_2 \cdot x[n-2] + \dots + b_N \cdot x[n-N] - a_1 \cdot y[n-1] - a_2 \cdot y[n-2] + \dots + a_M \cdot y[n-M]$$

6

- Noyau de convolution & FIR:
  - Convolution 2D en TI: filtrage en 2 dimensions
  - La convolution est définie par ses coefficients qu'on applique point-à-point à l'image
  - D'un point de vue informatique: réduction
  - Autre nom (info) : stencil (à cause de son application et du balayage dans l'image)
- Exemple
  - moyennage 3x3 ! Somme des pixels divisée par 9  
(=multiplication/accumulation point-à-point par 1/9)



7

### Filtrage Linéaire: filtre moyenneur

---

- filtre moyenneur: moyenne des voisins

- filtre 3x3

- filtre 5x5

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

- avantage

- très simple et rapide

- inconvénient

- problème de la division par 9

- peu robuste

$$\frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

- Remarques générales

- filtres de taille impaire pour centrer le résultat

- filtre normalisé: 1 en entrée donne 1 en sortie

8

## Filtrage Linéaire: filtre binomial

---

- Approximation du filtre de Gauss: moyenne pondérée des voisins

- filtre 3x3

$$1/16 \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

- filtre 5x5

- avantage

- rapide

- plus robuste que moyenneur

- inconvénient

- un plus complexe à calculer

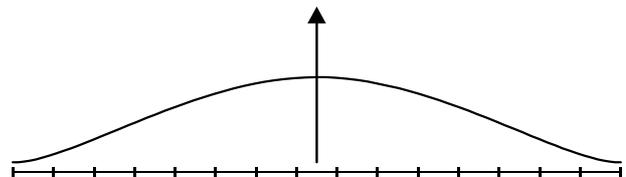
$$1/64 \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

9

## Filtrage Linéaire: filtre gaussien

---

$$G(x, y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\left(\frac{x^2+y^2}{\sigma^2}\right)}$$



- Largeur de la fenêtre

- typiquement  $2\sigma+1$

- avantage

- filtre paramétrable (sigma)

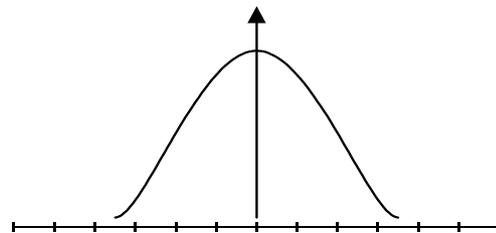
- adapter au problème

- taille de la fenêtre

- valeur de sigma

- Inconvénient

- complexité (calcul flottant et non entier)



10

## Filtrage Linéaire: filtre gaussien

- Construction version #1
  - sigma = 1
  - calcul en flottant
  - normalisation
  - conversion en entier

2E-05	2E-04	0.001	0.002	0.001	2E-04	2E-05
2E-04	0.003	0.013	0.022	0.013	0.003	2E-04
0.001	0.013	0.059	0.097	0.059	0.013	0.001
0.002	0.022	0.097	0.159	0.097	0.022	0.002
0.001	0.013	0.059	0.097	0.059	0.013	0.001
2E-04	0.003	0.013	0.022	0.013	0.003	2E-04
2E-05	2E-04	0.001	0.002	0.001	2E-04	2E-05
1	12.18	54.6	90.02	54.6	12.18	1
12.18	148.4	665.1	1097	665.1	148.4	12.18
54.6	665.1	2981	4915	2981	665.1	54.6
90.02	1097	4915	8103	4915	1097	90.02
54.6	665.1	2981	4915	2981	665.1	54.6
12.18	148.4	665.1	1097	665.1	148.4	12.18
1	12.18	54.6	90.02	54.6	12.18	1
1	12	54	90	54	12	1
12	148	665	1096	665	148	12
54	665	2980	4914	2980	665	54
90	1096	4914	8103	4914	1096	90
54	665	2980	4914	2980	665	54
12	148	665	1096	665	148	12
1	12	54	90	54	12	1

0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003
1	4.482	7.389	4.482	1
4.482	20.09	33.12	20.09	4.482
7.389	33.12	54.6	33.12	7.389
4.482	20.09	33.12	20.09	4.482
1	4.482	7.389	4.482	1
1	4	7	4	1
4	20	33	20	4
7	33	54	33	7
4	20	33	20	4
1	4	7	4	1

## Filtrage Linéaire: filtre gaussien

- Construction version #1
  - sigma = 2
  - calcul en flottant
  - normalisation
  - conversion en entier

0.004	0.008	0.011	0.013	0.011	0.008	0.004
0.008	0.015	0.021	0.024	0.021	0.015	0.008
0.011	0.021	0.031	0.035	0.031	0.021	0.011
0.013	0.024	0.035	0.04	0.035	0.024	0.013
0.011	0.021	0.031	0.035	0.031	0.021	0.011
0.008	0.015	0.021	0.024	0.021	0.015	0.008
0.004	0.008	0.011	0.013	0.011	0.008	0.004
1	1.868	2.718	3.08	2.718	1.868	1
1.868	3.49	5.078	5.755	5.078	3.49	1.868
2.718	5.078	7.389	8.373	7.389	5.078	2.718
3.08	5.755	8.373	9.488	8.373	5.755	3.08
2.718	5.078	7.389	8.373	7.389	5.078	2.718
1.868	3.49	5.078	5.755	5.078	3.49	1.868
1	1.868	2.718	3.08	2.718	1.868	1
1	1	2	3	2	1	1
1	3	5	5	5	3	1
2	5	7	8	7	5	2
3	5	8	9	8	5	3
2	5	7	8	7	5	2
1	3	5	5	5	3	1
1	1	2	3	2	1	1

0.015	0.021	0.024	0.021	0.015
0.021	0.031	0.035	0.031	0.021
0.024	0.035	0.04	0.035	0.024
0.021	0.031	0.035	0.031	0.021
0.015	0.021	0.024	0.021	0.015
1	1.455	1.649	1.455	1
1.455	2.117	2.399	2.117	1.455
1.649	2.399	2.718	2.399	1.649
1.455	2.117	2.399	2.117	1.455
1	1.455	1.649	1.455	1
1	1	1	1	1
1	2	2	2	1
1	2	2	2	1
1	2	2	2	1
1	1	1	1	1

## Filtrage Linéaire: filtre gaussien

- Construction version #2
  - sigma = 1
  - quantification en virgule fixe (Q8)
  - conversion en entier

2E-05	2E-04	0.001	0.002	0.001	2E-04	2E-05
2E-04	0.003	0.013	0.022	0.013	0.003	2E-04
0.001	0.013	0.059	0.097	0.059	0.013	0.001
0.002	0.022	0.097	0.159	0.097	0.022	0.002
0.001	0.013	0.059	0.097	0.059	0.013	0.001
2E-04	0.003	0.013	0.022	0.013	0.003	2E-04
2E-05	2E-04	0.001	0.002	0.001	2E-04	2E-05
0.005	0.061	0.275	0.453	0.275	0.061	0.005
0.061	0.746	3.344	5.514	3.344	0.746	0.061
0.275	3.344	14.99	24.71	14.99	3.344	0.275
0.453	5.514	24.71	40.74	24.71	5.514	0.453
0.275	3.344	14.99	24.71	14.99	3.344	0.275
0.061	0.746	3.344	5.514	3.344	0.746	0.061
0.005	0.061	0.275	0.453	0.275	0.061	0.005
0	0	0	0	0	0	0
0	0	3	5	3	0	0
0	3	14	24	14	3	0
0	5	24	40	24	5	0
0	3	14	24	14	3	0
0	0	3	5	3	0	0
0	0	0	0	0	0	0

0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003
0.746	3.344	5.514	3.344	0.746
3.344	14.99	24.71	14.99	3.344
5.514	24.71	40.74	24.71	5.514
3.344	14.99	24.71	14.99	3.344
0.746	3.344	5.514	3.344	0.746
0	0	0	0	0
0	3	5	3	0
3	14	24	14	3
5	24	40	24	5
3	14	24	14	3
0	3	5	3	0

13

## Filtrage Linéaire: filtre gaussien

- Construction version #2
  - sigma = 2
  - quantification en virgule fixe (Q8)
  - conversion en entier

0.004	0.008	0.011	0.013	0.011	0.008	0.004
0.008	0.015	0.021	0.024	0.021	0.015	0.008
0.011	0.021	0.031	0.035	0.031	0.021	0.011
0.013	0.024	0.035	0.04	0.035	0.024	0.013
0.011	0.021	0.031	0.035	0.031	0.021	0.011
0.008	0.015	0.021	0.024	0.021	0.015	0.008
0.004	0.008	0.011	0.013	0.011	0.008	0.004
1.074	2.006	2.918	3.307	2.918	2.006	1.074
2.006	3.747	5.452	6.178	5.452	3.747	2.006
2.918	5.452	7.933	8.989	7.933	5.452	2.918
3.307	6.178	8.989	10.19	8.989	6.178	3.307
2.918	5.452	7.933	8.989	7.933	5.452	2.918
2.006	3.747	5.452	6.178	5.452	3.747	2.006
1.074	2.006	2.918	3.307	2.918	2.006	1.074
1	2	2	3	2	2	1
2	3	5	6	5	3	2
2	5	7	8	7	5	2
3	6	8	10	8	6	3
2	5	7	8	7	5	2
2	3	5	6	5	3	2
1	2	2	3	2	2	1

0.015	0.021	0.024	0.021	0.015
0.021	0.031	0.035	0.031	0.021
0.024	0.035	0.04	0.035	0.024
0.021	0.031	0.035	0.031	0.021
0.015	0.021	0.024	0.021	0.015
3.747	5.452	6.178	5.452	3.747
5.452	7.933	8.989	7.933	5.452
6.178	8.989	10.19	8.989	6.178
5.452	7.933	8.989	7.933	5.452
3.747	5.452	6.178	5.452	3.747
3	5	6	5	3
5	7	8	7	5
6	8	10	8	6
5	7	8	7	5
3	5	6	5	3

14

## Filtrage Linéaire: filtre gaussien

---

- Comparaison  $\sigma = 1$ 
  - par normalisation: somme très grande (problème d'overflow)
  - par quantification: somme trop petite (due à  $\sigma = 1$ )

1	12	54	90	54	12	1
12	148	665	1096	665	148	12
54	665	2980	4914	2980	665	54
90	1096	4914	8103	4914	1096	90
54	665	2980	4914	2980	665	54
12	148	665	1096	665	148	12
1	12	54	90	54	12	1

0	0	0	0	0	0	0
0	0	3	5	3	0	0
0	3	14	24	14	3	0
0	5	24	40	24	5	0
0	3	14	24	14	3	0
0	0	3	5	3	0	0
0	0	0	0	0	0	0

1	4	7	4	1
4	20	33	20	4
7	33	54	33	7
4	20	33	20	4
1	4	7	4	1

0	3	5	3	0
3	14	24	14	3
5	24	40	24	5
3	14	24	14	3
0	3	5	3	0

15

## Filtrage Linéaire: filtre gaussien

---

- Comparaison  $\sigma = 2$ 
  - par normalisation: 7x7 OK, 5x5 trop plat (dû à  $\sigma = 2$ )
  - par quantification: OK

1	1	2	3	2	1	1
1	3	5	5	5	3	1
2	5	7	8	7	5	2
3	5	8	9	8	5	3
2	5	7	8	7	5	2
1	3	5	5	5	3	1
1	1	2	3	2	1	1

1	2	2	3	2	2	1
2	3	5	6	5	3	2
2	5	7	8	7	5	2
3	6	8	10	8	6	3
2	5	7	8	7	5	2
2	3	5	6	5	3	2
1	2	2	3	2	2	1

1	1	1	1	1
1	2	2	2	1
1	2	2	2	1
1	2	2	2	1
1	1	1	1	1

3	5	6	5	3
5	7	8	7	5
6	8	10	8	6
5	7	8	7	5
3	5	6	5	3

16

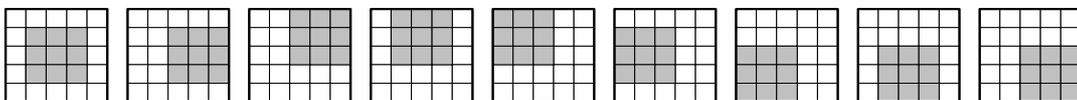
- Algorithme naïf
  - copie des pixels dans un tableau T ( $n=k*k$  pixel)
  - tri du tableau (tri par insertion, tri par sélection, tri à bulles)
  - le médian est au milieu, en  $T[n/2]$
- Remarque: tri rapide inefficace car pas assez de pixels
  - $O(n \cdot \log n)$ : constante cachée
- Avantage
  - résistance accru au bruit
- Algorithme rapide:
  - voir en TD

17

## filtrage non linéaire: filtrage de Nagao

---

- Algorithme
  - chaque noyau 3x3 renvoie deux valeurs:
    - valeur (pondérée) moyenne, médian
    - critère de sélection: mesure d'homogénéité (étendue, écart type, ...)
  - choix du noyau 3x3 maximisant/minimisant le critère
- Avantage
  - qualité des transitions
  - grande résistance au bruit
- Inconvénient
  - complexité de calcul



18

- Exemple

- Nota

- ici variance et étendue conduisent au même résultat

	10	22	3	4	10
	17	5	10	9	9
	11	10	5	2	12
	12	26	6	21	9
	3	10	17	4	10
moyenne	10.3	7.8	7.1		
	11.3	10.4	9.2		
	11.1	11.2	9.6		
variance	6.0	6.1	3.6		
	6.7	8.0	5.3		
	7.0	8.3	6.3		
étendue	19	20	10		
	21	24	19		
	23	24	19		



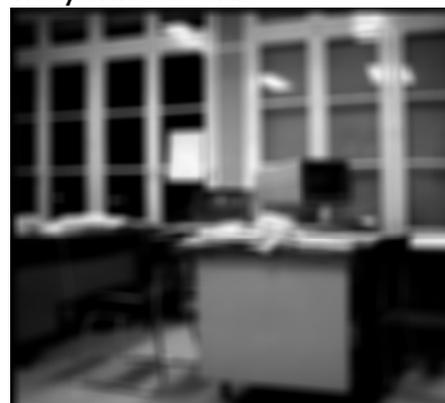
bureau



moyenneur 3x3



moyenneur 5x5



moyenneur 7x7



bureau



gaussien 0.75



gaussien 2.08



gaussien 4.08

21



bureau



médian 3x3



médian 5x5



nagao

22



bruit gaussien



moyenneur 3x3



moyenneur 7x7



gauss 0.75



gauss 4.08



bruit gaussien



médian 3x3



médian 7x7



nagao

25

---

filtre: bruitage gaussien (variance 120)

---



bruit gaussien

26



moyenieur 3x3



moyenieur 7x7



gauss 0.75



gauss 4.08

27



bruit gaussien



médian 3x3



médian 7x7



nagao

28



bruit impulsionnel 2%



moyenneur 3x3



moyenneur 7x7



gauss 0.75



gauss 4.08



bruit gaussien



médian 3x3



médian 7x7



nagao

31

---

filtrage: bruit impulsionnel (10%)

---



bruit impulsionnel 10%

32



moyenieur 3x3



moyenieur 7x7



gauss 0.75



gauss 4.08

33



bruit gaussien



médian 3x3

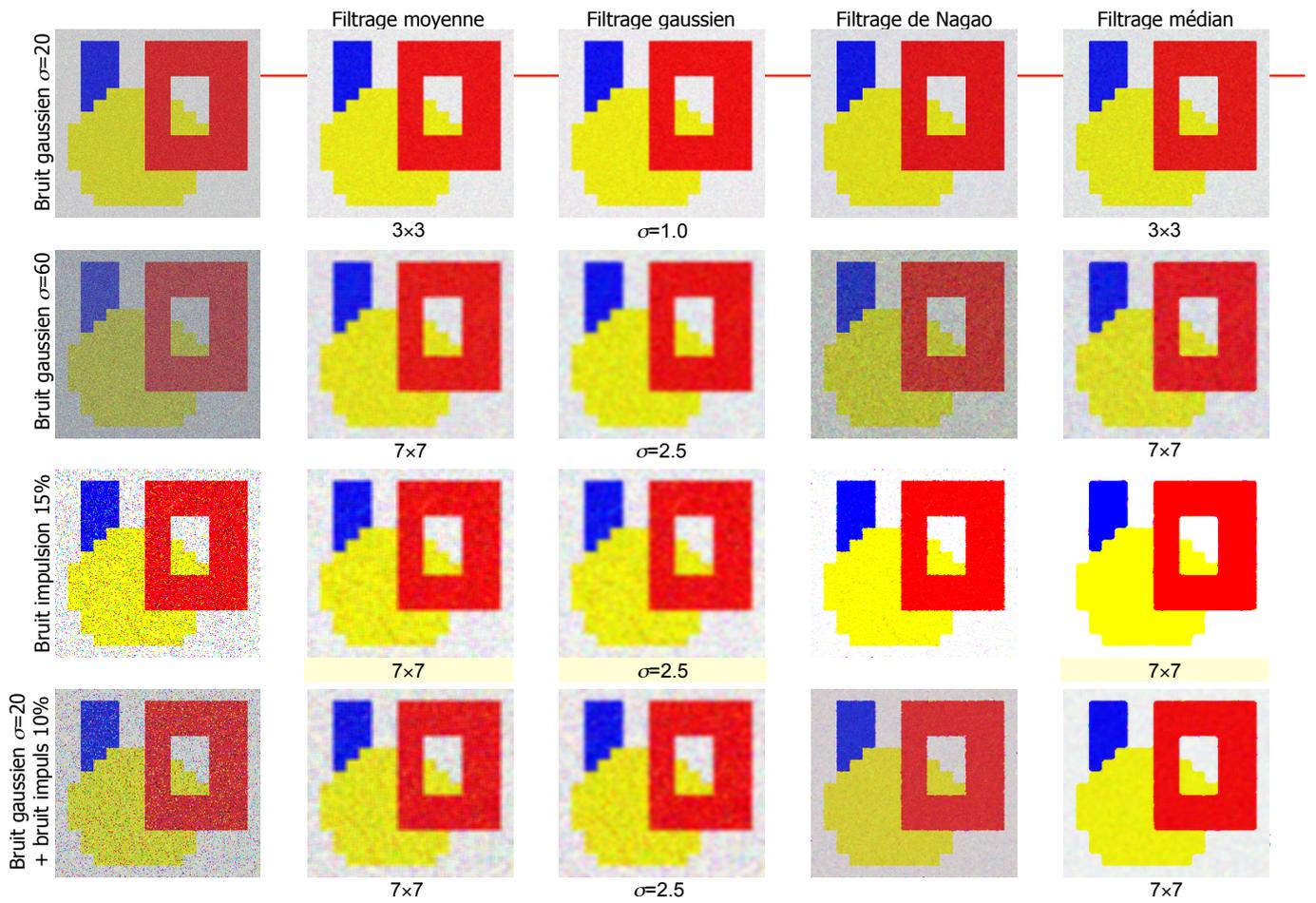


médian 7x7



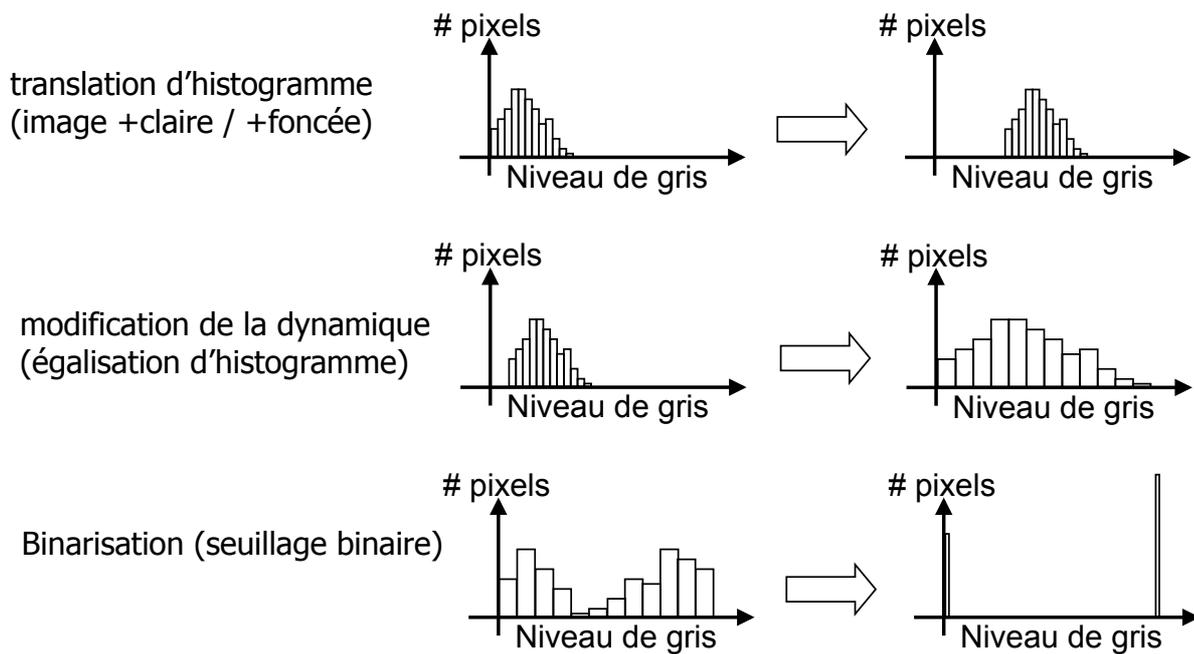
nagao

34



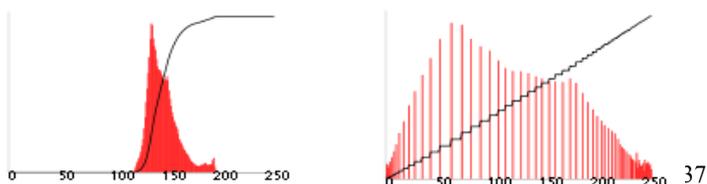
## Amélioration d'images

- Exemples de méthodes fondées sur des modifications de l'histogramme de l'image :



## Egalisation d'histogramme

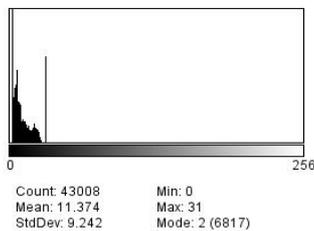
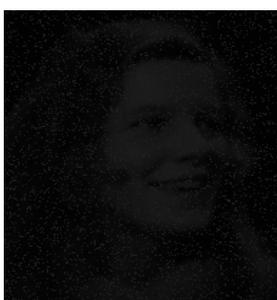
- Soient:
  - un histogramme sur L niveaux de gris
  - $n_k$ , le nombre de points ayant le niveau k = Le nombre d'occurrence du niveau k
  - n, le nombre total de points
  - La probabilité d'occurrence du niveau k est alors:  $p_x(k) = p(x=k) = \frac{n_k}{n}, 0 \leq k \leq L$
- Alors
  - L'histogramme  $p(x)$  peut être vu comme la densité de probabilité de x
  - L'histogramme cumulé  $H(x)$  peut être vu comme la probabilité de x 
$$H(x) = \sum_{k=0}^{k=x} p_x(k)$$
- On veut
  - H uniforme (impossible en discret)
  - Le nouvel histogramme  $h(x)$  est: 
$$h(x) = \text{round} \left( \frac{H(x) - H(0)}{n - H(0)} \times (L - 1) \right)$$
- Remarque
  - L'histogramme est sur  $[0..L-1]$
- Intérêt
  - calcul très simple par LUT  
Look Up Table



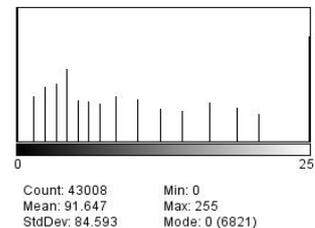
## Egalisation d'histogramme: des cas "d'école"

Après égalisation, l'histogramme ne contient que quelques raies

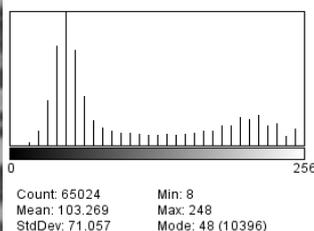
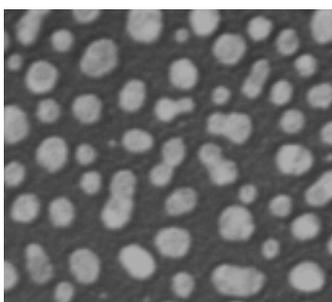
Il manque des niveaux de gris (importance de la quantification, ici 8 bits)



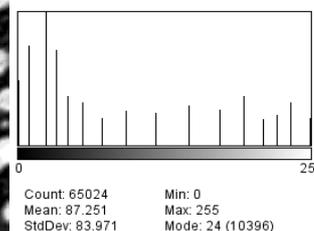
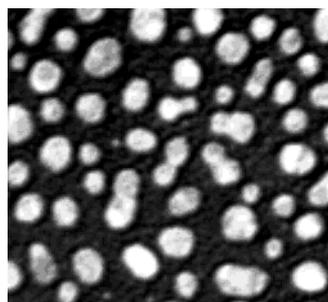
Avant égalisation



Après égalisation



Avant égalisation

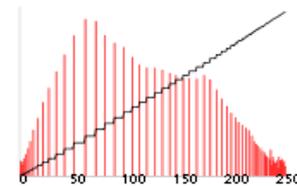
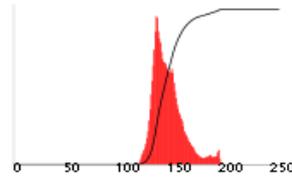


Après égalisation

## Egalisation d'histogramme

---

- Photo : Hawkes Bay (NZ), Phillip Capper

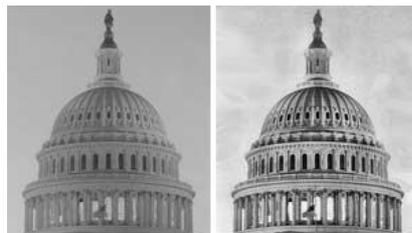


39

## Egalisation d'histogramme: CLAHE

---

- Algorithme moderne: CLAHE
  - Contrast Limited Adaptive Histogram Equalization
- Initialement: amélioration de contraste pour images médicales (scan, irm)
- Image: copyright © Vision=Systems



40

## Egalisation d'histogramme : RETINEX (Rétine+Cortex)

---

- Algorithme neuro-mimétique
- <http://dragon.larc.nasa.gov/retinex/pao/news/>

