



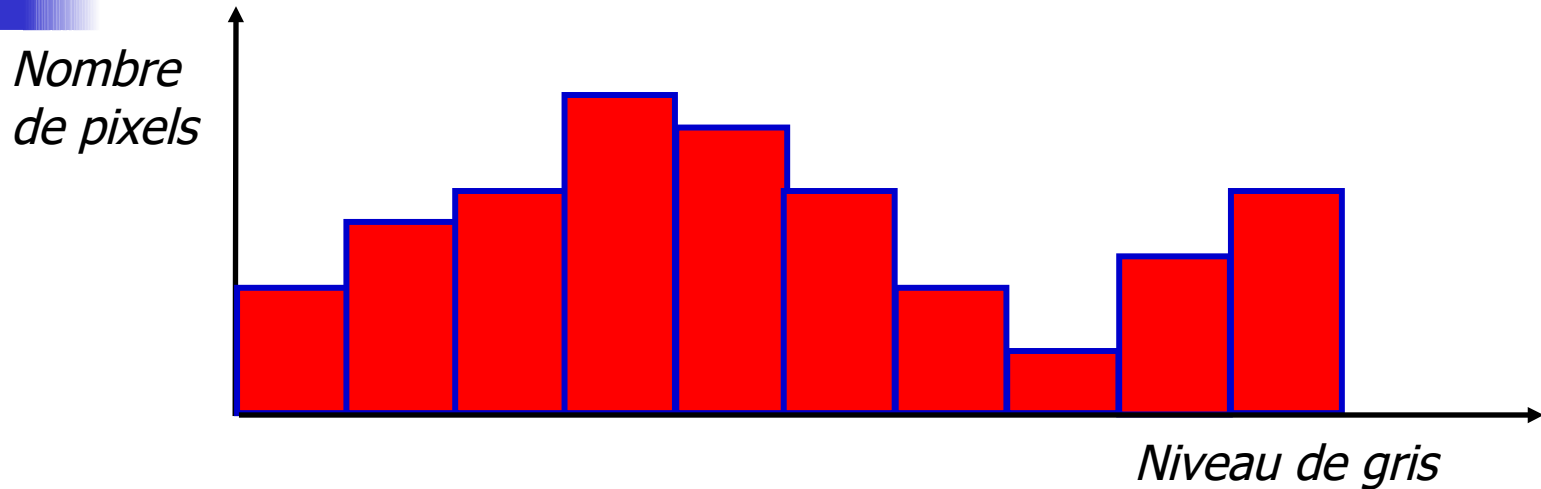
# Traitement d'images

---

## **Traitements de base**

NGUYEN Thi Oanh – IPH  
oanhnt@soict.hust.edu.vn

# Histogramme d'une image



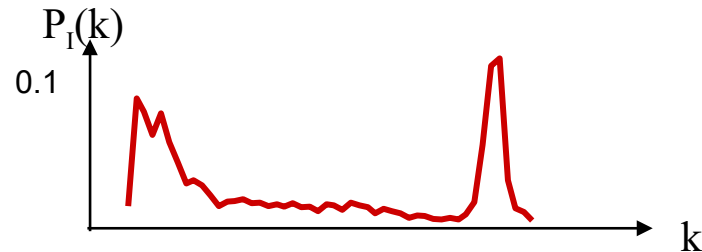
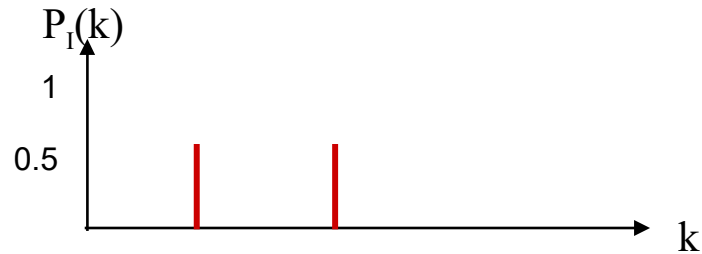
- L'histogramme représente la distribution des niveaux de gris (ou de couleurs) dans une image

**$H(k)$  = nombre de pixels de l'image ayant la valeur  $k$**

- Pour ignorer l'influence de la taille de l'image ==> Normaliser l'histogramme :

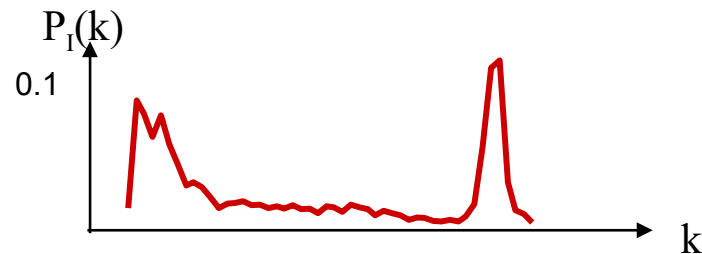
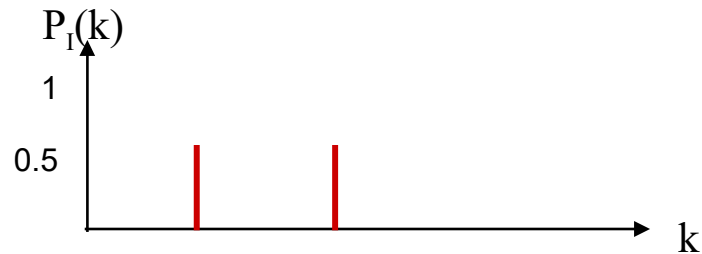
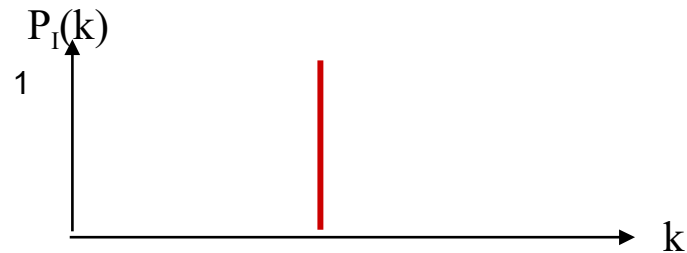
**$h(k) = H(K) / \text{nombre de pixels de l'image}$**

# Histogramme d'une image



*Dynamique d'une image* = [valeur\_min, valeur\_max]

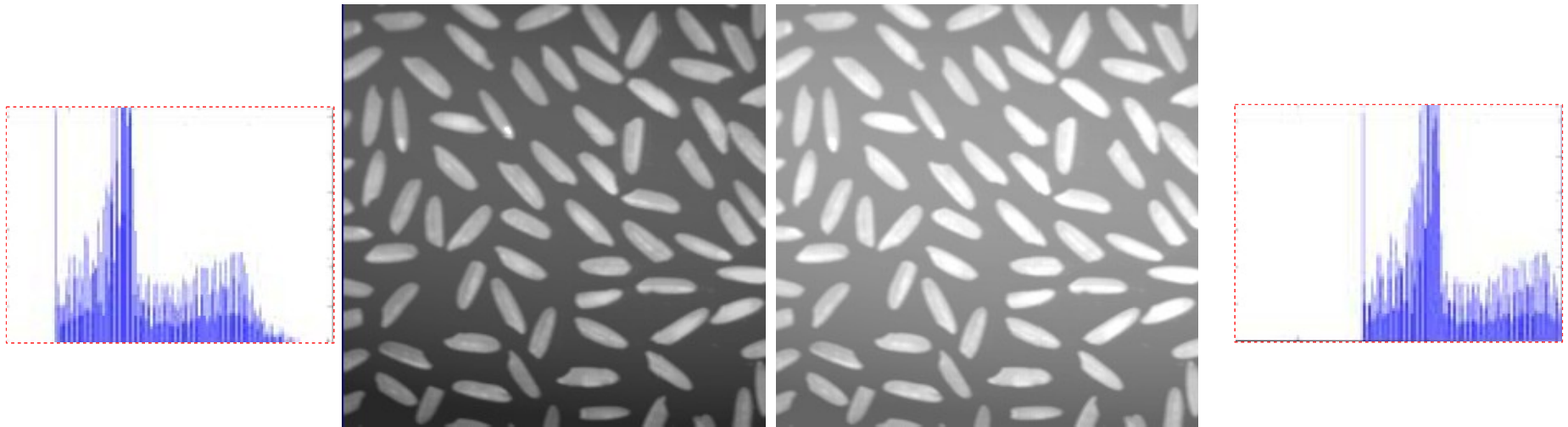
# Histogramme d'une image



*Dynamique d'une image* = [valeur\_min, valeur\_max]

# Luminance ou brillance d'une image

- La luminance (ou brillance) est définie comme la moyenne de tous les pixels de l'image
- Dans les deux images suivantes, seule la luminance est différente :





# Contraste d'une image

---

- Le contraste peut être défini de plusieurs façons :
  - Ecart-type des variations des niveaux de gris

$$C = \sqrt{\frac{1}{M \times N} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} (f(x,y) - Moy)^2}$$

- Variation entre niveaux de gris min et max

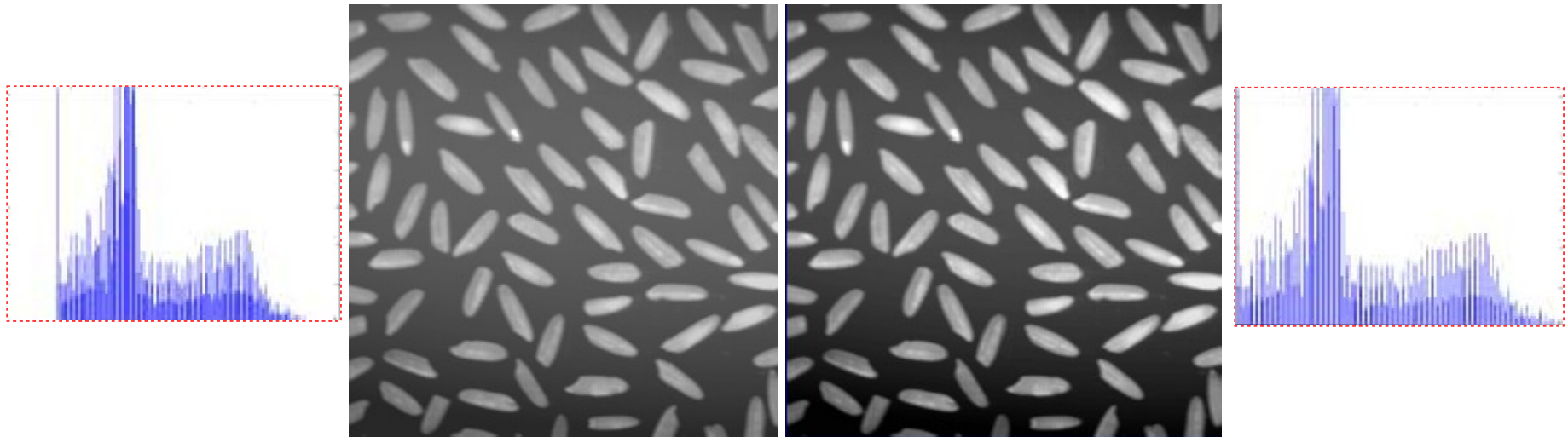
$$C = \frac{\max[f(x,y)] - \min[f(x,y)]}{\max[f(x,y)] + \min[f(x,y)]}$$



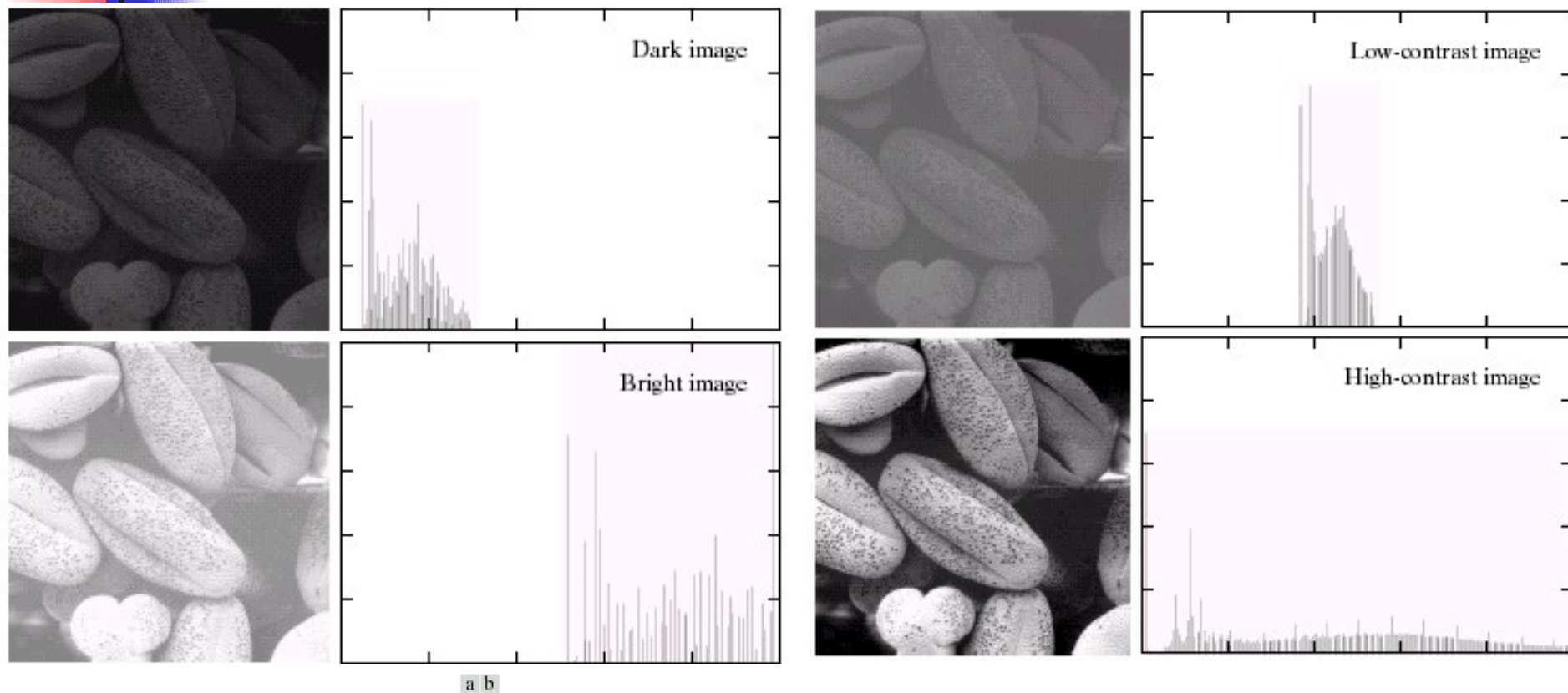
# Contraste d'une image

---

Les deux images suivantes possèdent un contraste différent :



# Exemples de contrastes d'une image



**FIGURE 3.15** Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)



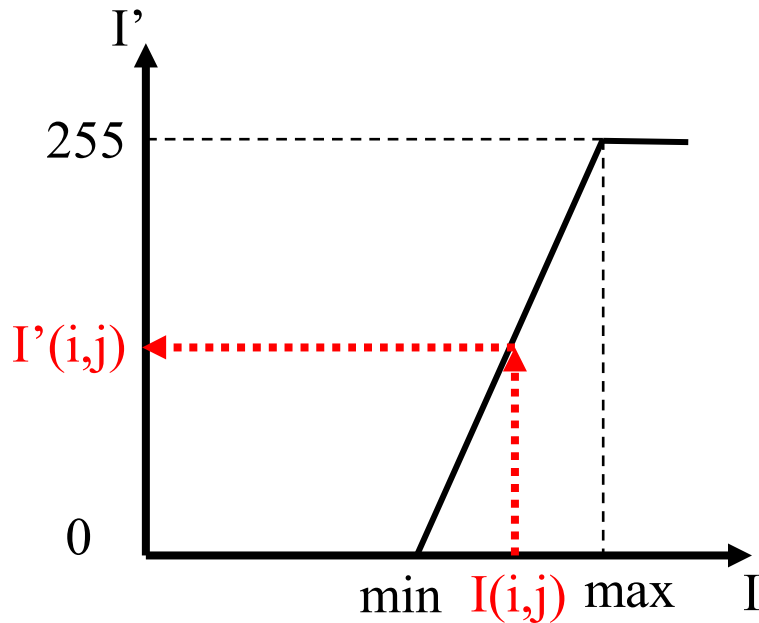


# Amélioration du contraste

---

- Plusieurs méthodes possibles :
  - Transformation linéaire
  - Transformation linéaire avec saturation
  - Transformation linéaire par morceau
  - Transformation non-linéaire
  - Égalisation de l'histogramme

# Transformation linéaire



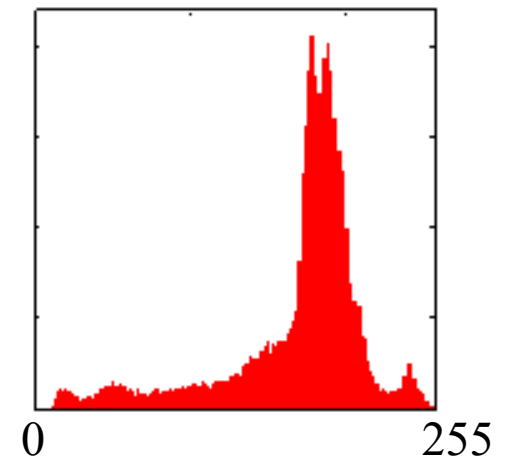
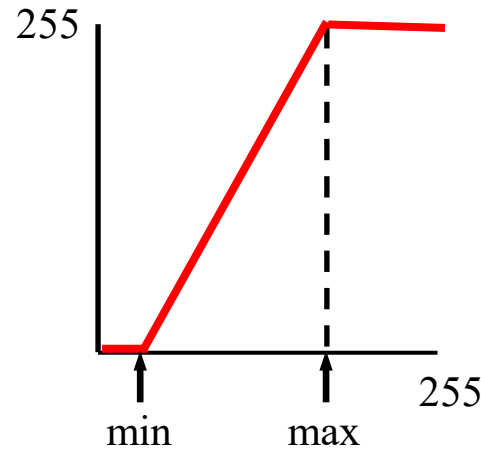
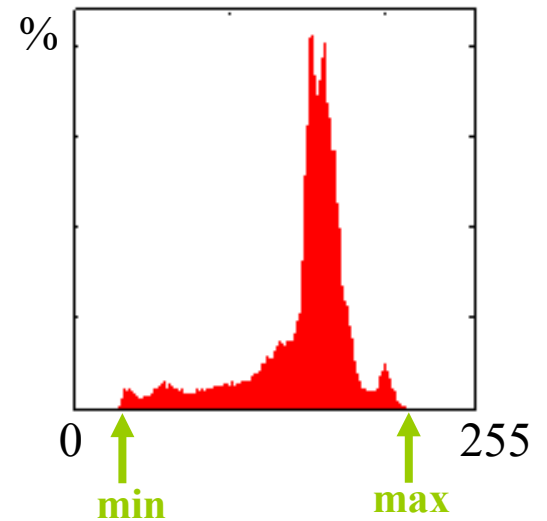
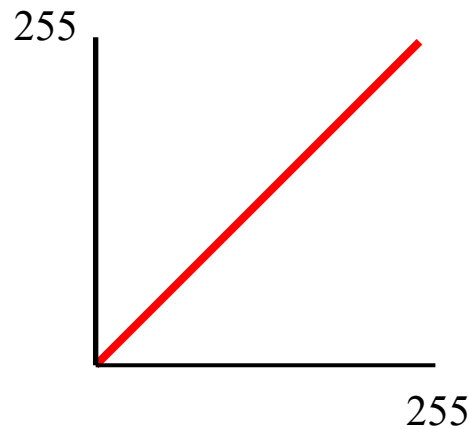
$$\frac{\max - \min}{I(i, j) - \min} = \frac{255 - 0}{I'(i, j) - 0}$$

Alors:

$$I'(i, j) = \frac{255}{\max - \min} (I(i, j) - \min)$$

$$\text{avec } \frac{I(i, j) - \min}{\max - \min} \in [0, 1]$$

# Transformation linéaire





# Implémentation...

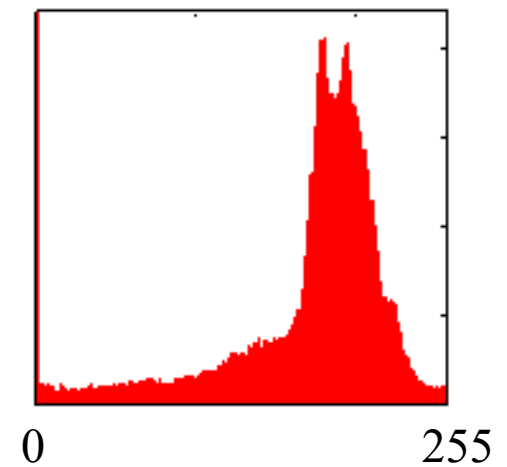
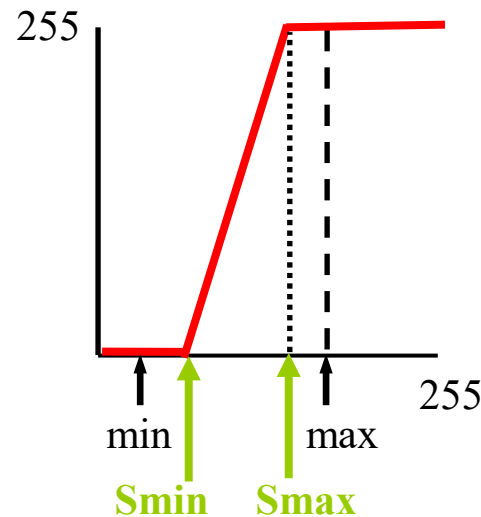
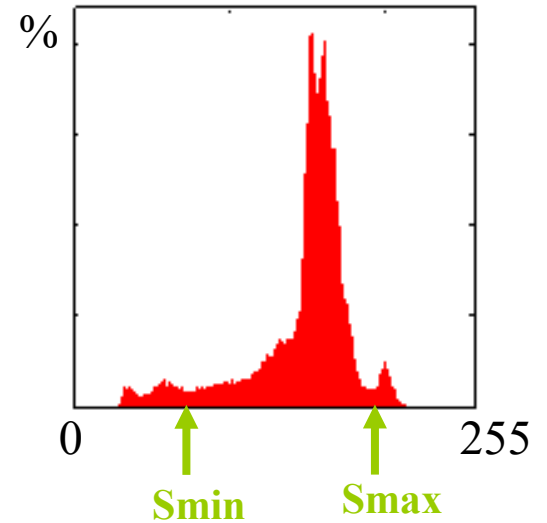
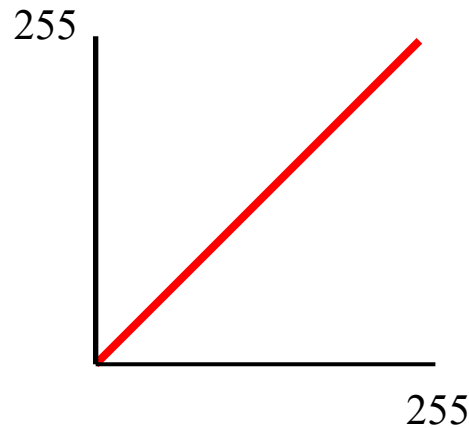
```
Pour i=1 à nblig  
  Pour j=1 à nbcol  
     $I'(i,j) = 255 * (I(i,j) - \min I) / (\max I - \min I);$ 
```

Pas optimal

## ➔ Utilisation d'une LUT (Look Up Table)

```
/* Initialisation de la LUT */  
Pour i=0 à 255  
   $LUT[i] = 255 * (i - \min I) / (\max I - \min I);$   
  
/* Initialisation de la LUT */  
Pour i=1 à nblig  
  Pour j=1 à nbcol  
     $I'(i,j) = LUT[I(i,j)];$ 
```

# Transformation linéaire avec saturation





# Transformation linéaire avec saturation

---

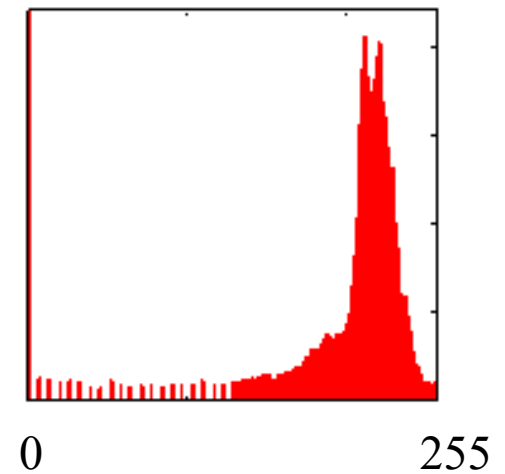
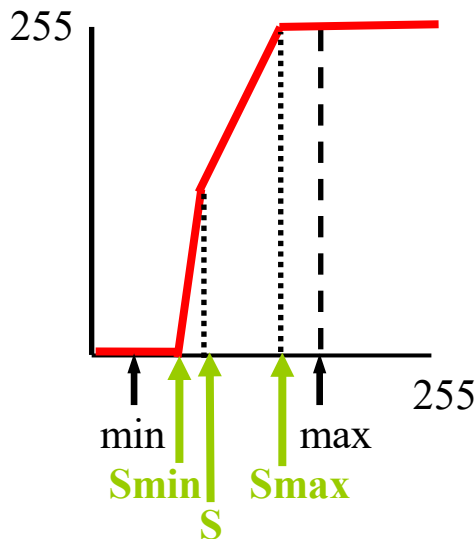
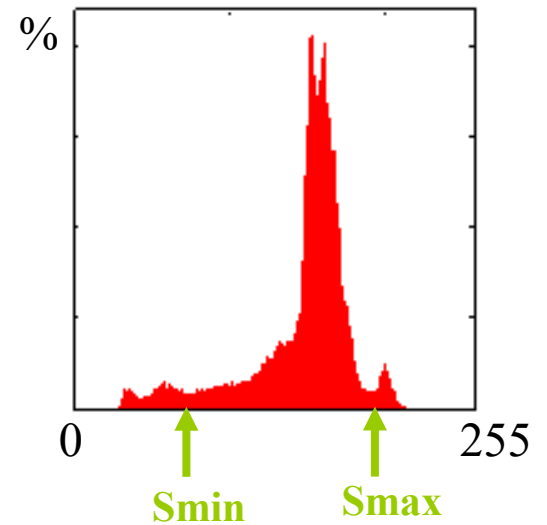
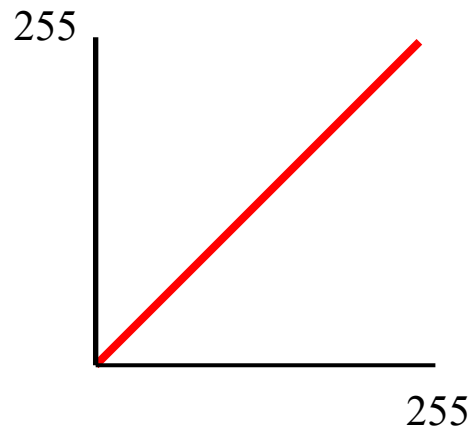
$$I'(i, j) = \frac{255}{S_{\max} - S_{\min}} (I(i, j) - S_{\min})$$

$$I'(i, j) \leq 0 \Rightarrow I'(i, j) = 0$$

$$I'(i, j) \geq 255 \Rightarrow I'(i, j) = 255$$

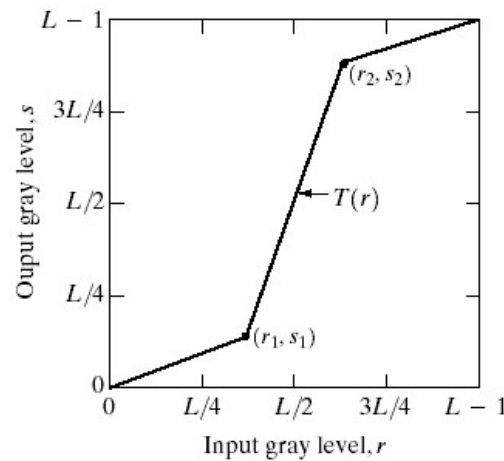
$$\min(I(i, j)) \leq S_{\min} < S_{\max} \leq \max(I(i, j))$$

# Transformation linéaire par morceaux



# Transformation linéaire par morceaux

*Exemple d'une fonction ad-hoc de modification du contraste de l'image.*



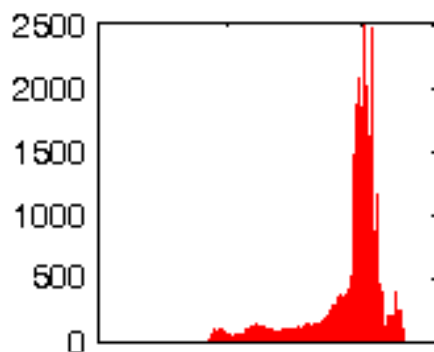
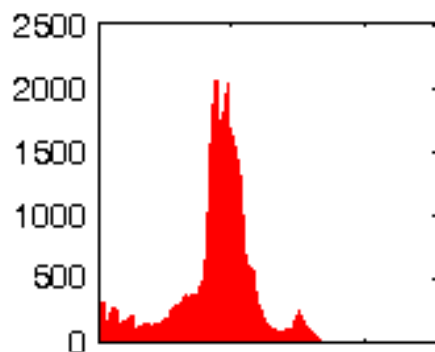
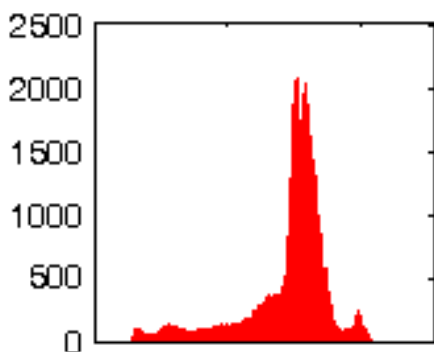
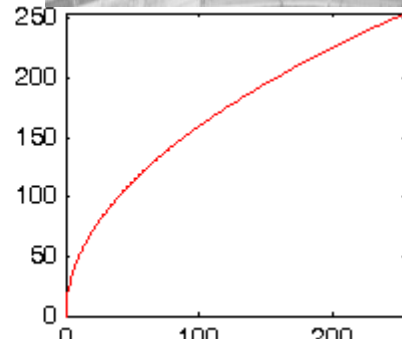
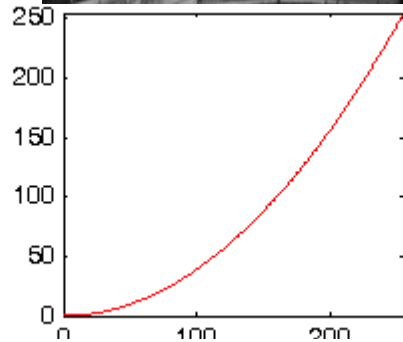
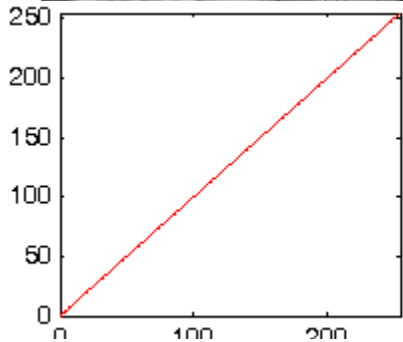
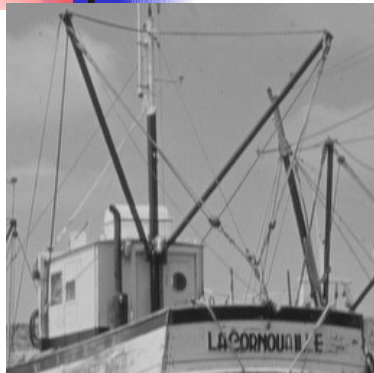
a b  
c d

**FIGURE 3.10**

Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)



# Transformation non-linéaire



Ex: Correction  
gamma (g)

$$I_{\text{émise}} = k \cdot n^{\gamma}$$

$$\Rightarrow n' = n^{\frac{1}{\gamma}}$$

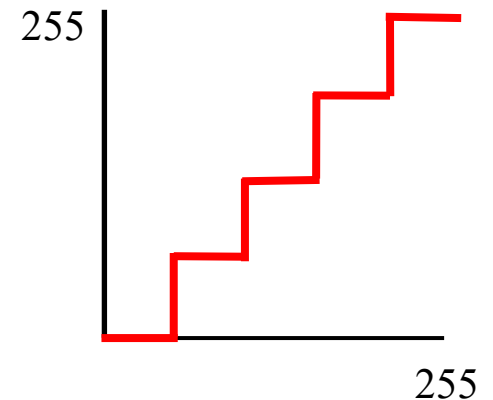
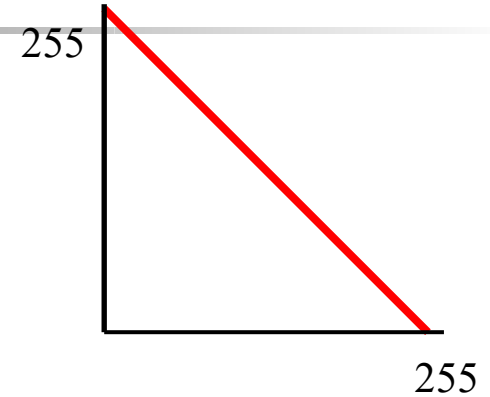
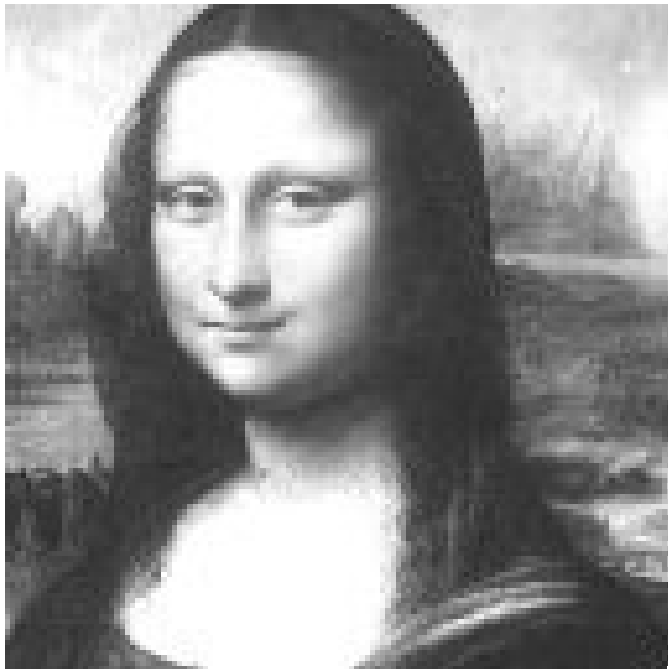
$$\gamma_{\text{écran}} \in [1.3, 3.0]$$

$$\gamma_{\text{oeil}} \approx \frac{1}{2} \text{ à } \frac{1}{3}$$

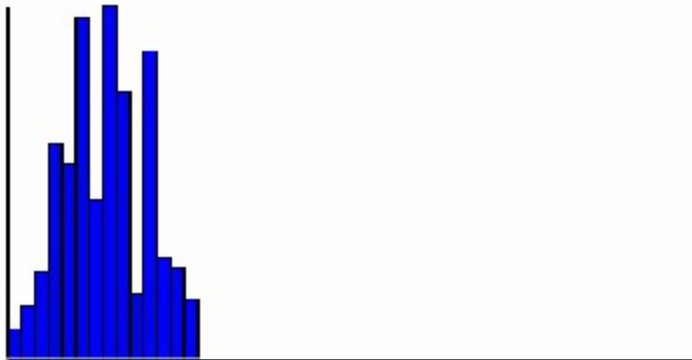
# Ex: gamma correction



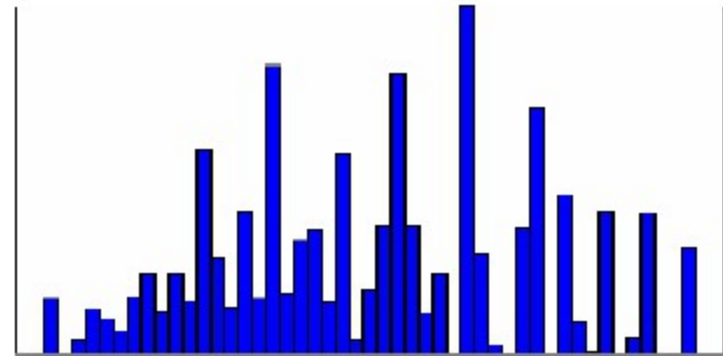
# Autres fonctions possibles



# Correction de la dynamique de l'image

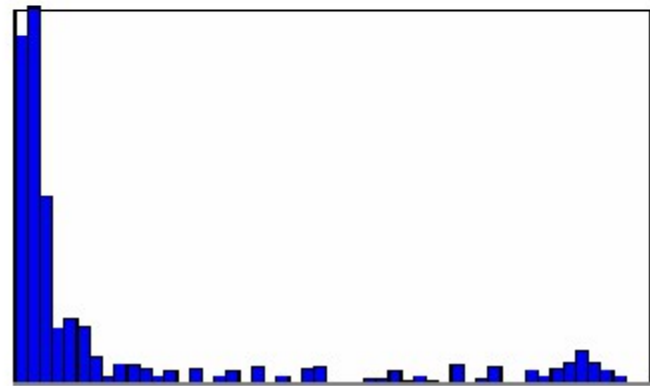
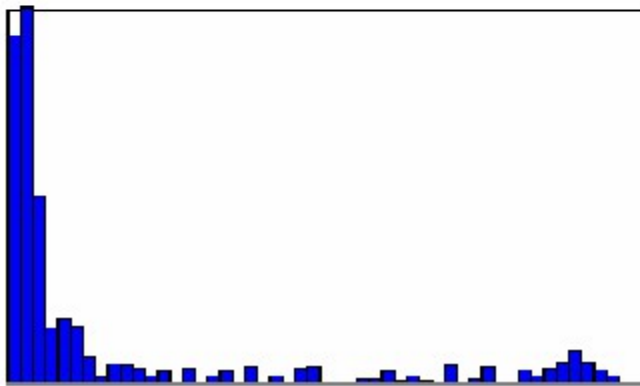


*Image originale*



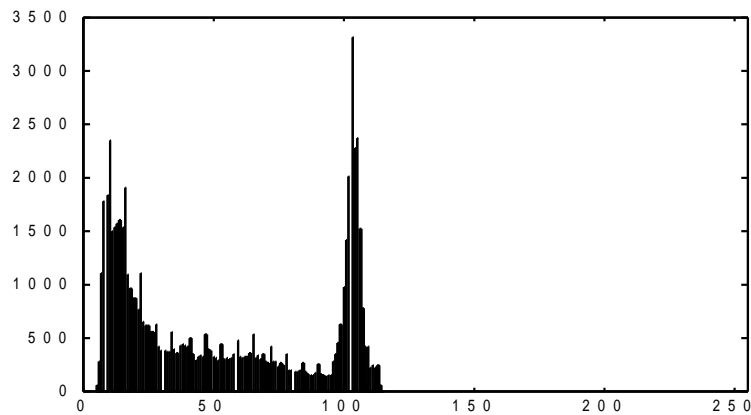
*Image restaurée*

# Correction de la dynamique de l'image

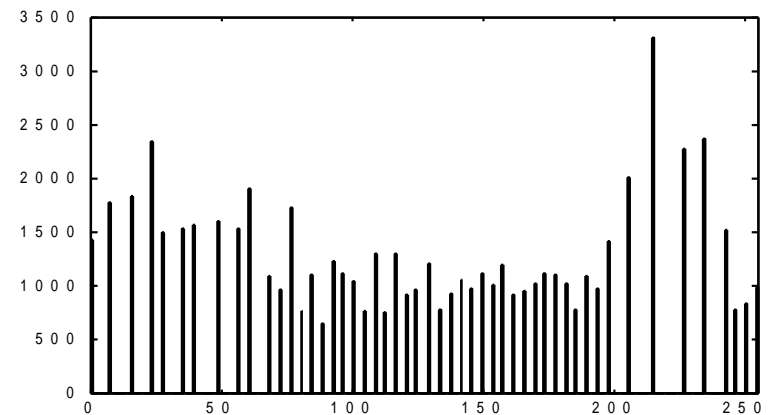


*Dans le cas où l'histogramme initial occupe **toute la plage de dynamique**, aucun changement n'est visible.*

# Egalisation de l'histogramme



*Image originale*



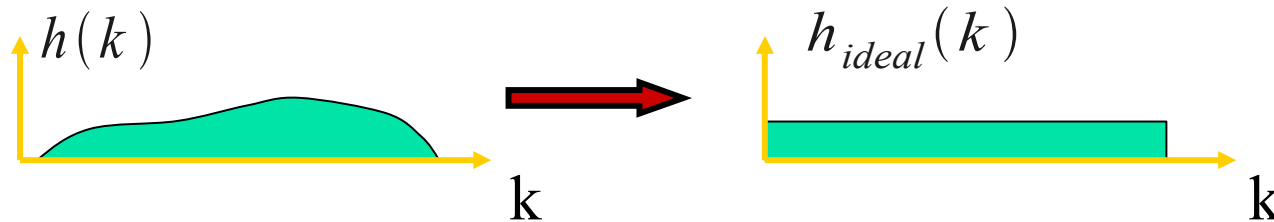
*Image plus contrastée*

*CVIPTools : Utilities>Enhance>Histogram Equalization*

*Source : Tal Hassner. Computer Vision. Weizmann Institute of Science (Israel).*

# Egalisation de l'histogramme

- Pour améliorer le contraste, on cherche à aplanir l'histogramme

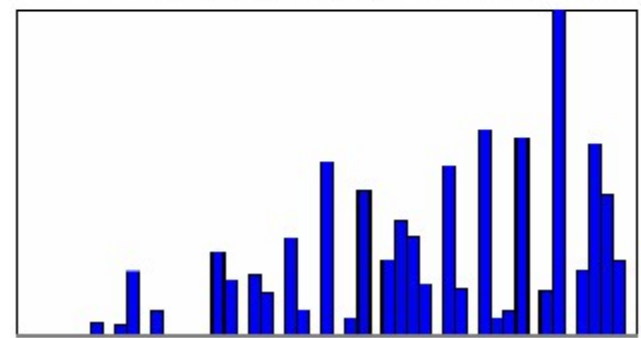
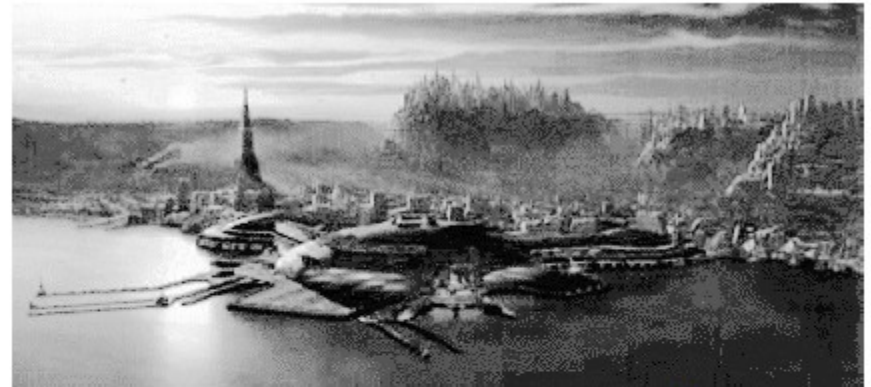
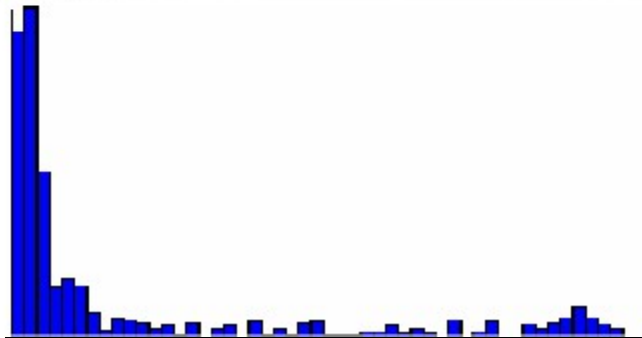


- Etape 1 : Calcul de l'histogramme  $h(i) \quad i \in [0, 255]$
- Etape 2 : Normalisation de l'histogramme  
(Nbp : nombre de pixels de l'image)  $h_n(i) = \frac{h(i)}{Nbp} \quad i \in [0, 255]$
- Etape 3 : Densité de probabilité normalisé  $C(i) = \sum_{j=0}^i h_n(j) \quad i \in [0, 255]$
- Etape 4 : Transformation des niveaux de gris de l'image  $f'(x, y) = C(f(x, y)) \times 255$

*Note : voir ImAnalyse pour des exemples d'histogrammes cumulés*



# Egalisation de l'histogramme

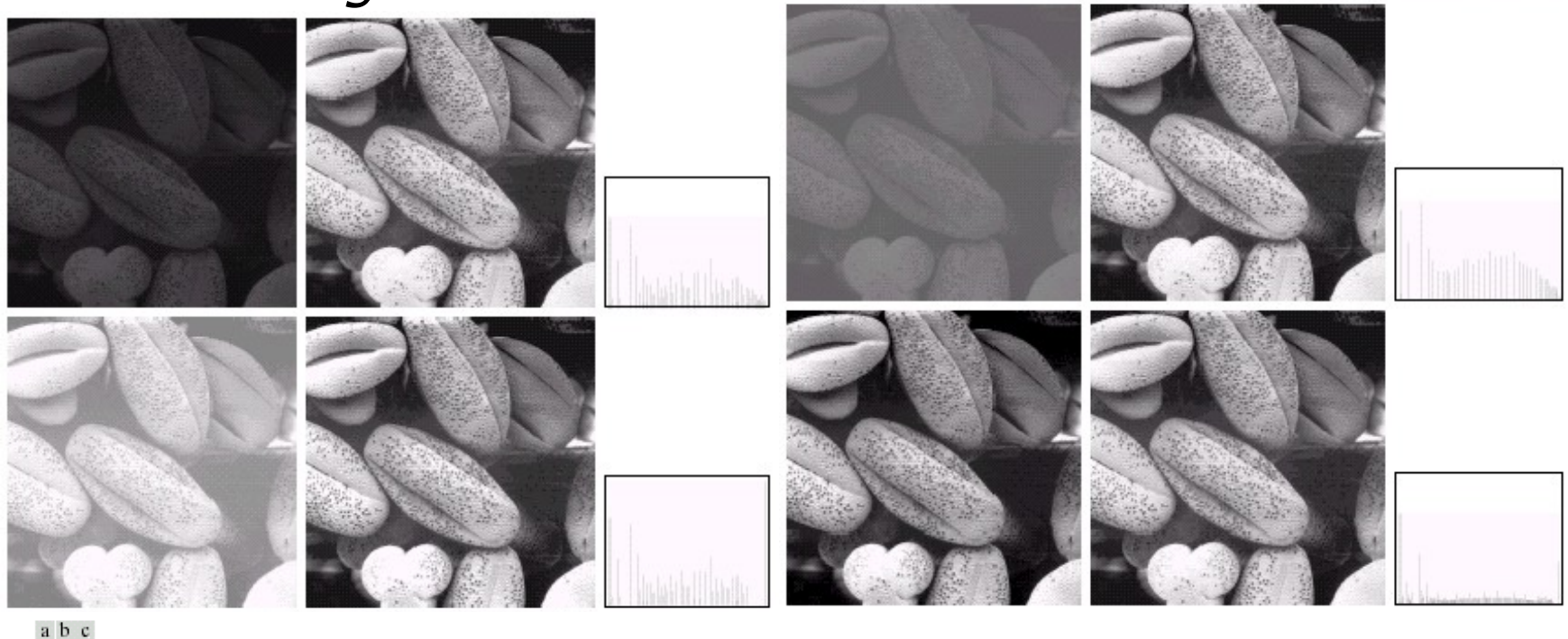


*L'égalisation d'histogramme peut améliorer une image là où la correction de dynamique de l'histogramme est inefficace.*



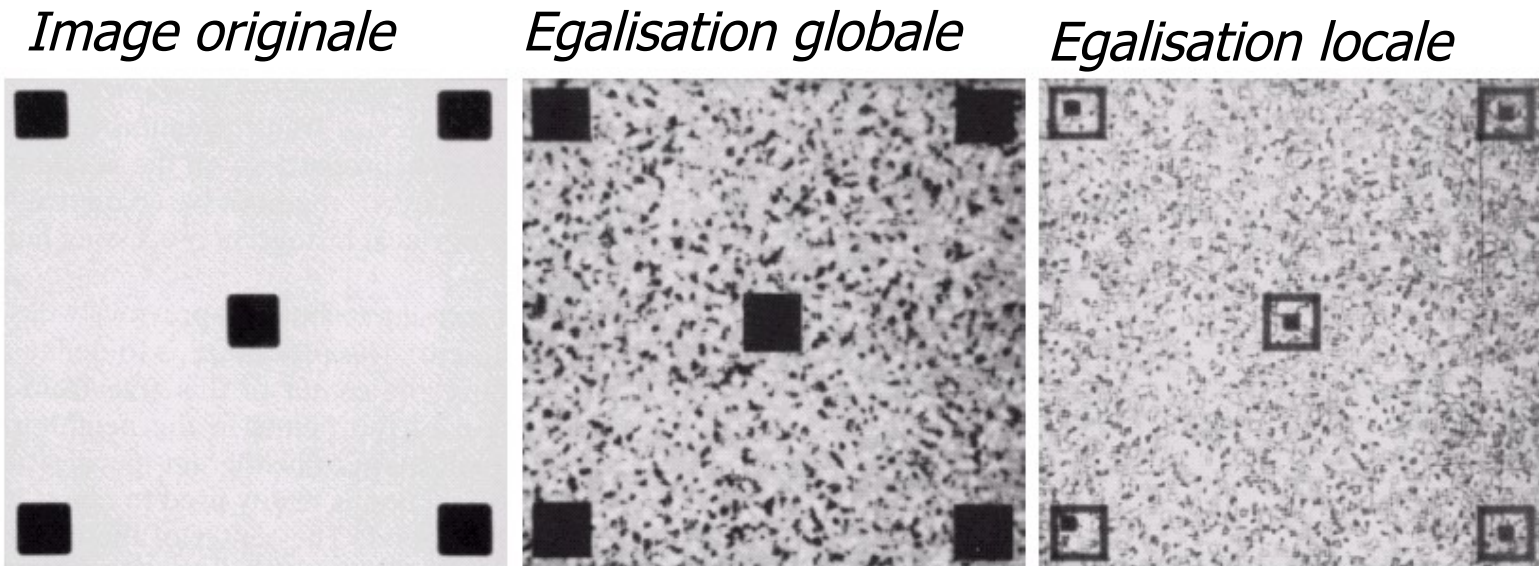
# Egalisation de l'histogramme

*Si on prend la **même image** avec des **contrastes différents**, l'égalisation d'histogramme donne le **même résultat** pour toutes les images.*



**FIGURE 3.17** (a) Images from Fig. 3.15. (b) Results of histogram equalization. (c) Corresponding histograms.

# Egalisation (locale) de l'histogramme



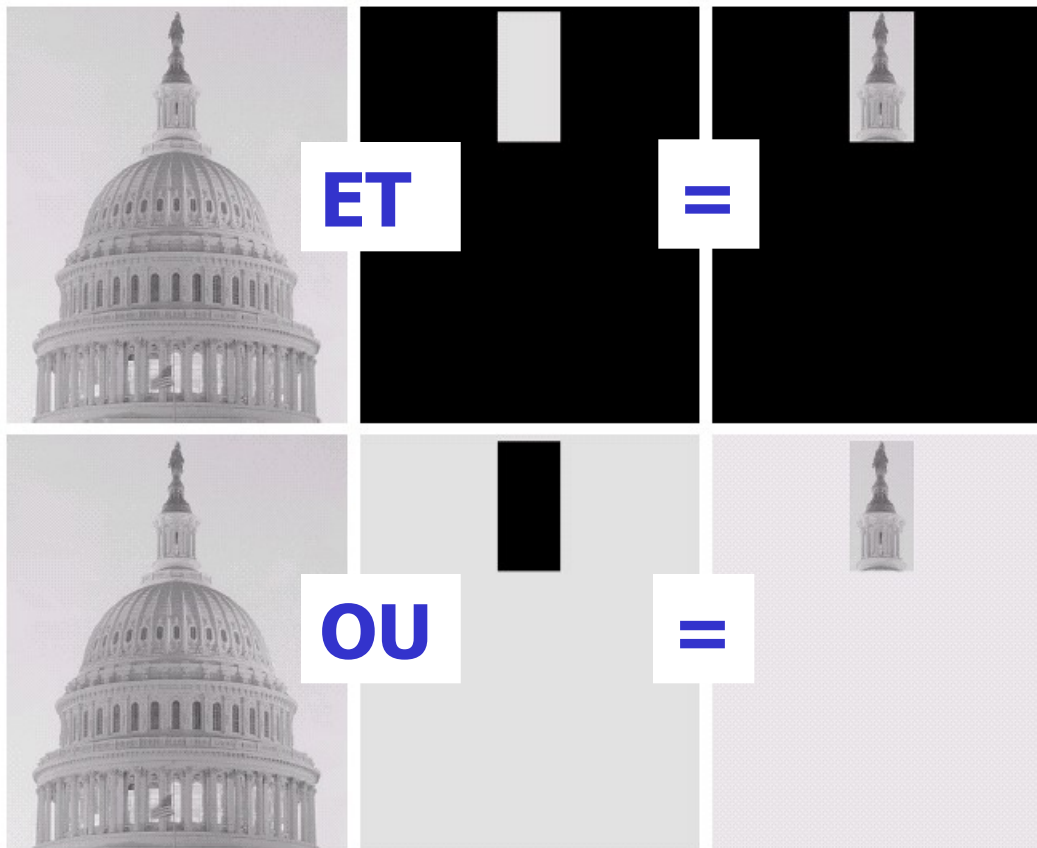
a b c

**FIGURE 3.23** (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization using a  $7 \times 7$  neighborhood about each pixel.

*L'égalisation **locale** de l'histogramme est faite en prenant une fenêtre de 7x7 autour de chaque pixel.*

# Opérations sur les images (ET,OU)

Les opérations logiques fonctionnent aussi sur les images.

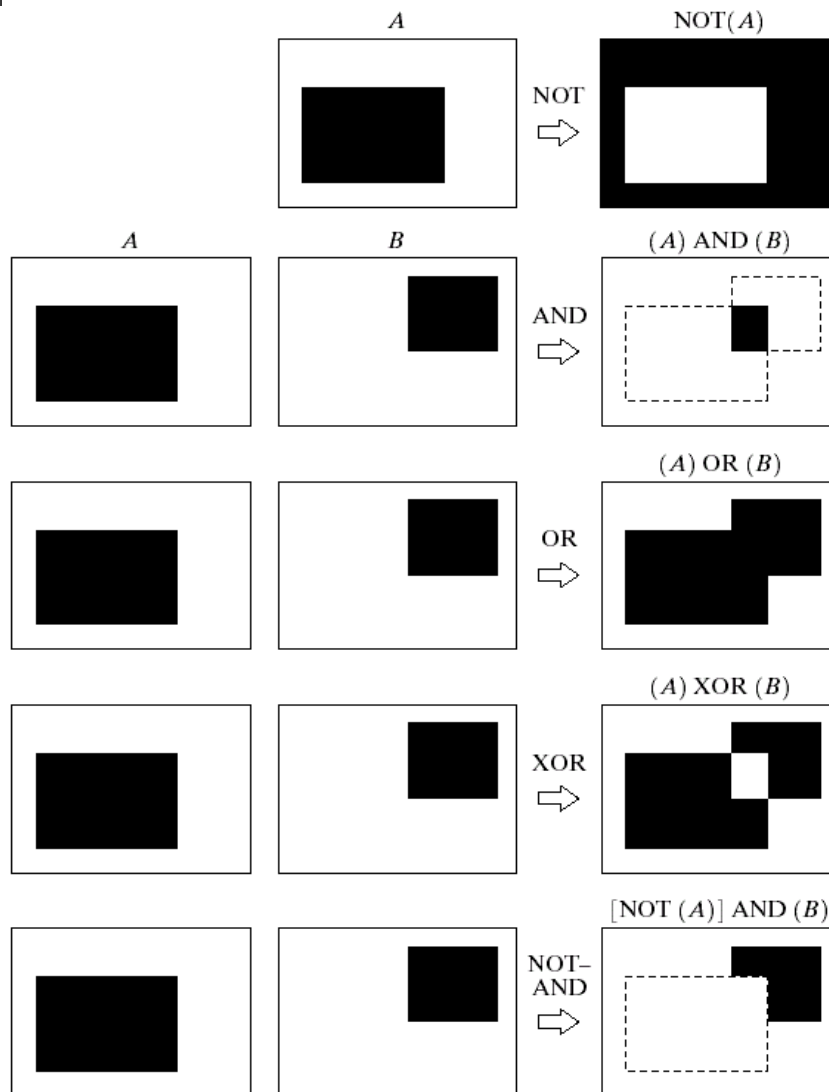


a	b	c
d	e	f

**FIGURE 3.27**

(a) Original image. (b) AND image mask. (c) Result of the AND operation on images (a) and (b). (d) Original image. (e) OR image mask. (f) Result of operation OR on images (d) and (e).

# Opérations sur les images (ET,OU)



**FIGURE 9.3** Some logic operations between binary images. Black represents binary 1s and white binary 0s in this example.

# Addition d'images

- Si  $f$  et  $g$  sont deux images, on peut définir l'addition  $R$  pixel à pixel de ces deux images par :

$$R(x,y) = \text{Min}( f(x,y)+g(x,y) ; 255 )$$

- L'addition d'images peut permettre
  - De diminuer le bruit d'une vue dans une série d'images
  - D'augmenter la luminance en additionnant une image avec elle-même



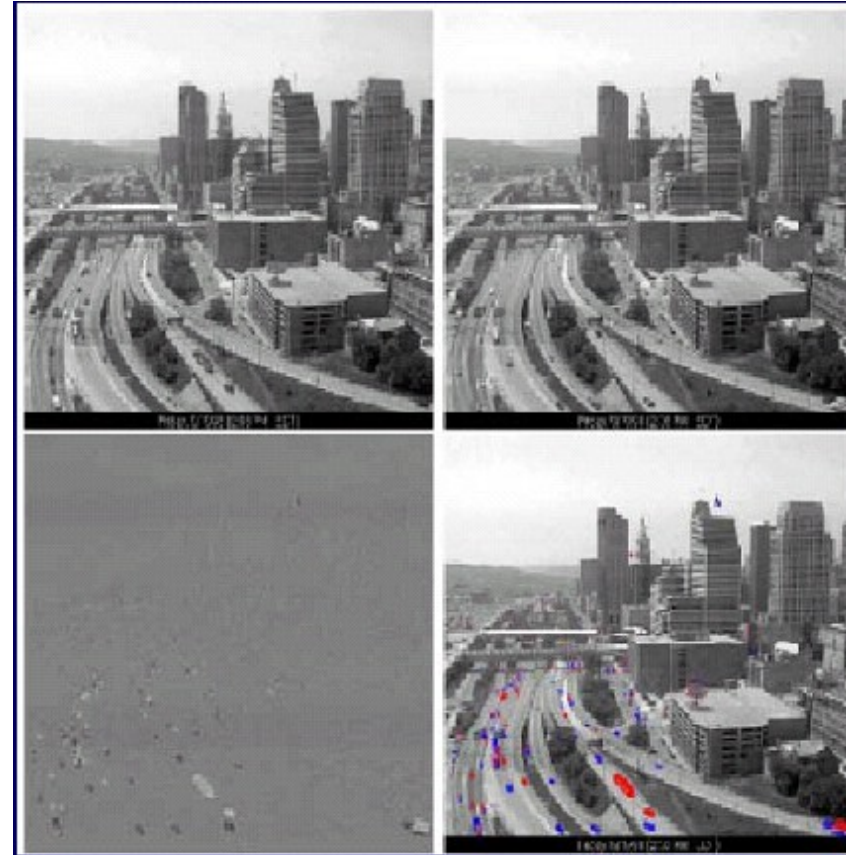


# Soustraction d'images

- On peut définir la soustraction  $S$  pixel à pixel de deux images  $f$  et  $g$  par :

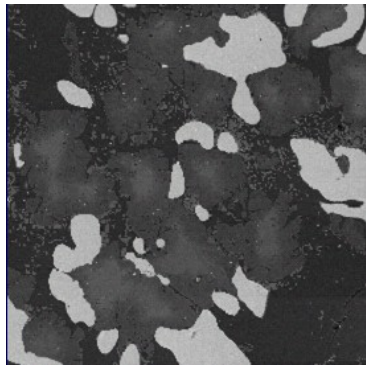
$$S(x,y) = \text{Max}( f(x,y)-g(x,y) ; 0 )$$

- La soustraction d'images peut permettre
  - Détection de défauts
  - Détection de mouvements

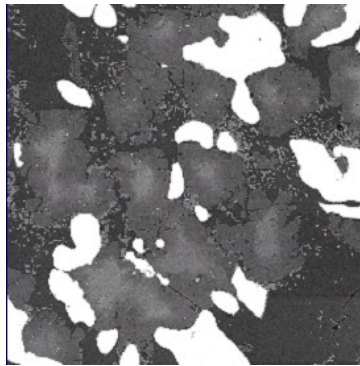


# Multiplication d'images

- La multiplication  $S$  d'une image  $f$  par un ratio (facteur) peut se définir par :  
$$S(x,y) = \text{Min}( f(x,y) \cdot \text{ratio} ; 255)$$
- La multiplication d'images peut permettre d'améliorer le contraste ou la luminosité



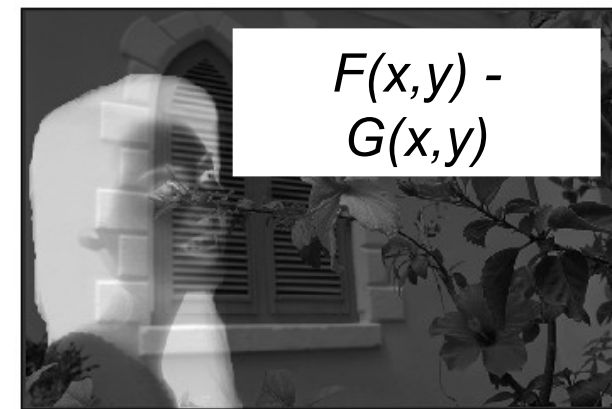
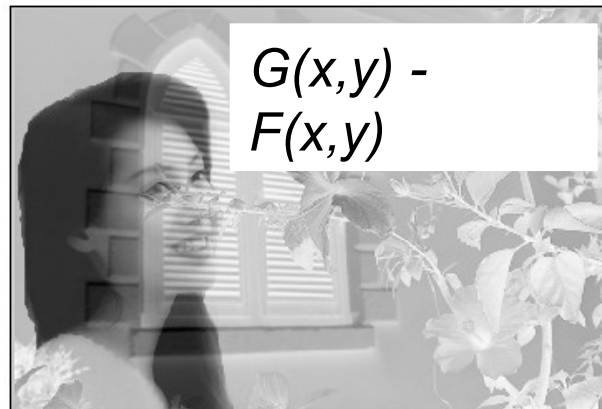
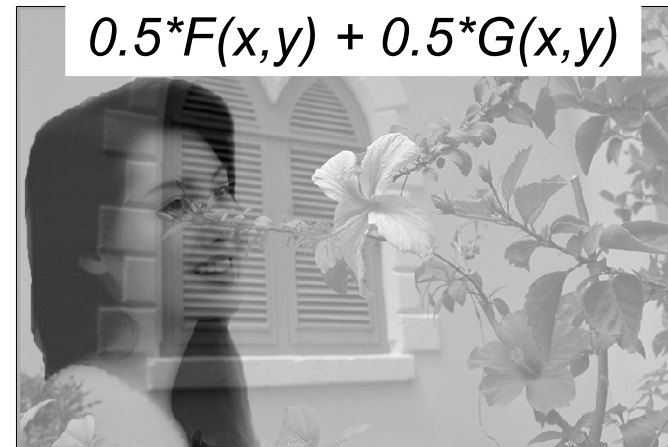
**x1,5 =**



**x1,2 =**



# Opérations sur les images (+,-)







# Interpolation d'images

---

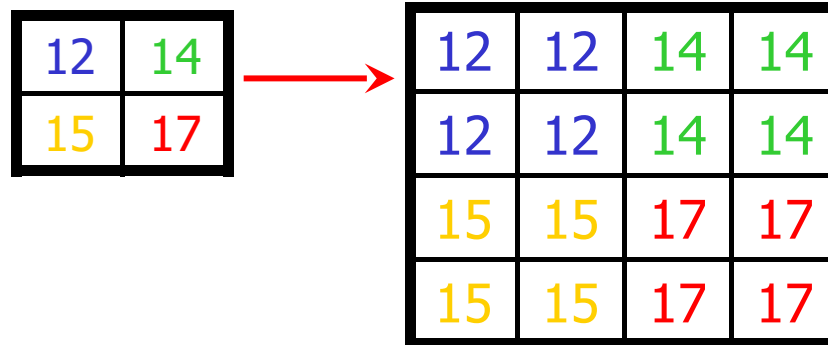
# Changement d'échelle (interpolation)



**FIGURE 2.19** A  $1024 \times 1024$ , 8-bit image subsampled down to size  $32 \times 32$  pixels. The number of allowable gray levels was kept at 256.

# Interpolation du plus proche voisin

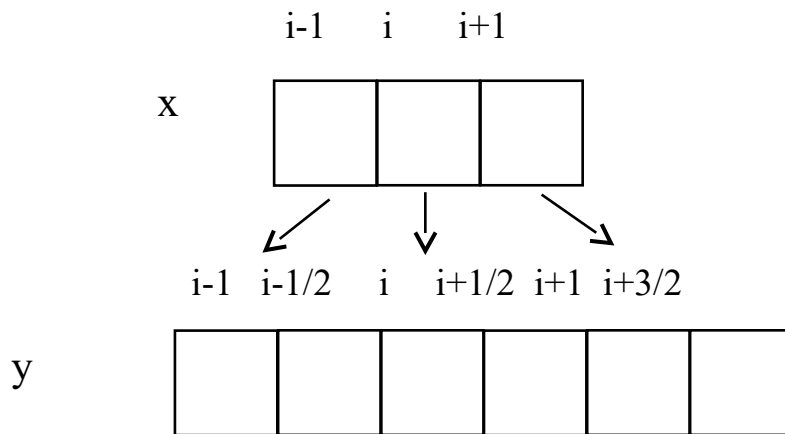
- Interpolation du plus proche voisin par copie des pixels
  - Copie de chaque colonne et de chaque rang



# Interpolation bilinéaire

- Zoom !
- Interpolation bilinéaire
- Fonction bilinéaire de 4 pixels voisins (en 2D)

## Principe 1D



$$y_i = x_i$$

$$y_{i+1/2} = (x_i + x_{i+1}) / 2$$

Problème pour le dernier point

$\Rightarrow$  extrapolation linéaire de :

$$= 2 * x_N - x_{N-1}$$



# Interpolation bicubique

## Principe 1D

- Même raisonnement que précédemment sauf que

$$y_{i+1/2} = (-x_{i-1} + 9x_i + 9x_{i+1} - x_{i+2}) / 16$$

- Problèmes pour  $y_{1/2}$ , ainsi que pour les 2 derniers points

- $\Rightarrow$  2 extrapolations :
  - une parabolique à distance 1  $x_{-1} = 3x_0 - 3x_1 + x_2$
  - une linéaire à distance 2  $x_N = 3x_{N-1} - 3x_{N-2} + x_{N-3}$   
 $x_{N+1} = 2x_N - x_{N-1}$

- Le facteur zoom peut être plus élevé, par exemple 3, ce qui donne pour une interpolation cubique :

$$y_{i+1/3} = (-5x_{i-1} + 60x_i + 30x_{i+1} - 4x_{i+2}) / 81$$

$$y_{i+2/3} = (-4x_{i-1} + 30x_i + 60x_{i+1} - 5x_{i+2}) / 81$$

# Exemples d'interpolation

*Plus proche  
voisin*



*Bilinéaire*



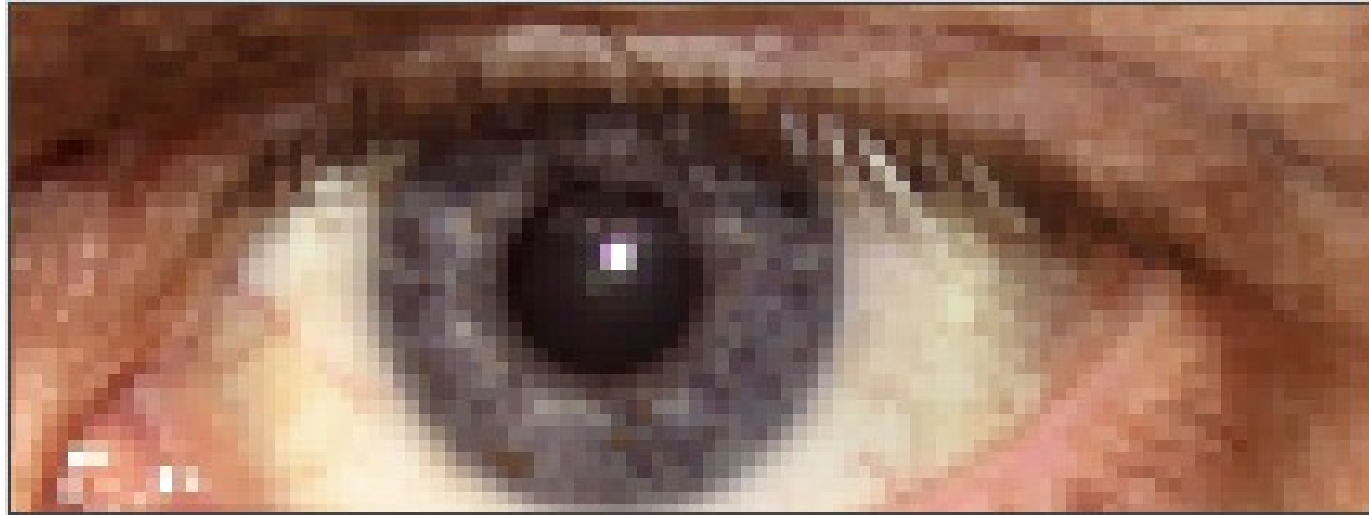
a	b	c
d	e	f

**FIGURE 2.25** Top row: images zoomed from  $128 \times 128$ ,  $64 \times 64$ , and  $32 \times 32$  pixels to  $1024 \times 1024$  pixels, using nearest neighbor gray-level interpolation. Bottom row: same sequence, but using bilinear interpolation.

# Changement d'échelle



*Plus proche  
voisin*



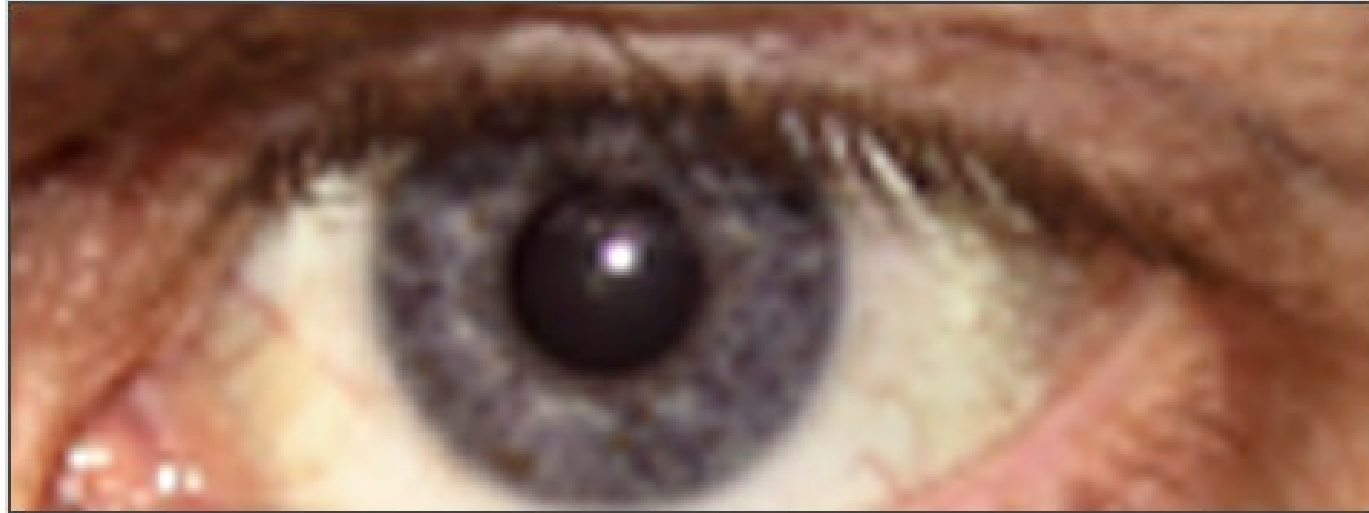
*Bilinéaire  
(4 voisins)*



# Changement d'échelle



*Bicubique  
(16 voisins)*

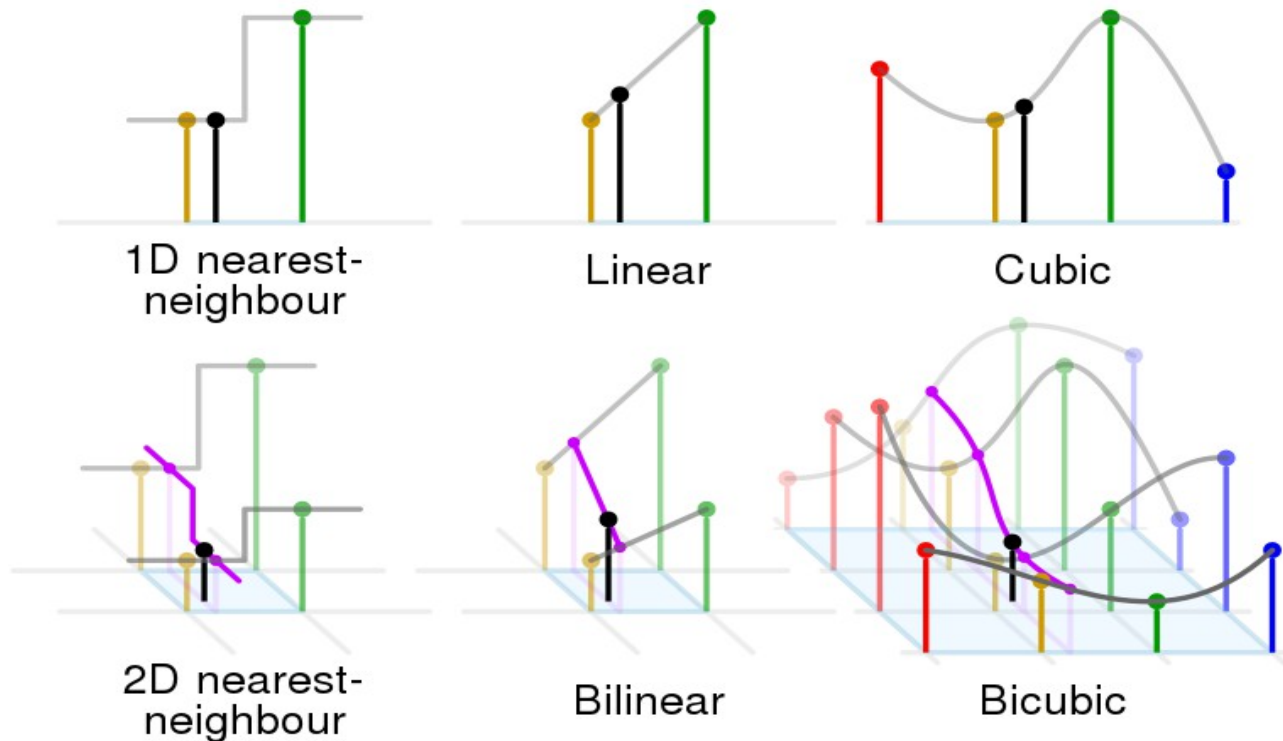


*Bilinéaire  
(4 voisins)*





# Changement d'échelle





# Références

---

- Introduction au Traitement d'Images (Lingrand), chapitre 4.
- Digital Image Processing 2ed (Gonzalez & Woods), chapitres 2 et 3.
- Wikipédia : Histogramme, Brightness (luminance), Contrast, Lookup table, Color balance (amélioration du contraste), [Correction gamma](#), Egalisation d'histogramme, Image scaling (interpolation d'images),
- Gamma correction:  
<http://www.cambridgeincolour.com/tutorials/gamma-correction.htm>



# Références

---

- Eric Favier. Cours sur L'analyse et le traitement des images ; Les principes de la vision assistée par ordinateur. ENISE (France). *(n'est plus disponible en ligne)*
  - <http://www.enise.fr/perso/favier/vision/indexvision.htm>
  - Cours sur les traitements ponctuels :  
[http://www.enise.fr/perso/favier/vision/traitements\\_ponctuels.pdf](http://www.enise.fr/perso/favier/vision/traitements_ponctuels.pdf)
- Caroline Rougier. Cours de Traitement d'images (IFT2730). Université de Montréal (Canada) *(n'est plus disponible en ligne)*
  - <http://www-etud.iro.umontreal.ca/~rougierc/ift2730/>
  - Chap 9 : Histogramme : étirements, égalisation, manipulation, seuillage :  
[http://www-etud.iro.umontreal.ca/~rougierc/ift2730/cours/Cours9\\_IFT2730\\_2008\\_2.pdf](http://www-etud.iro.umontreal.ca/~rougierc/ift2730/cours/Cours9_IFT2730_2008_2.pdf)