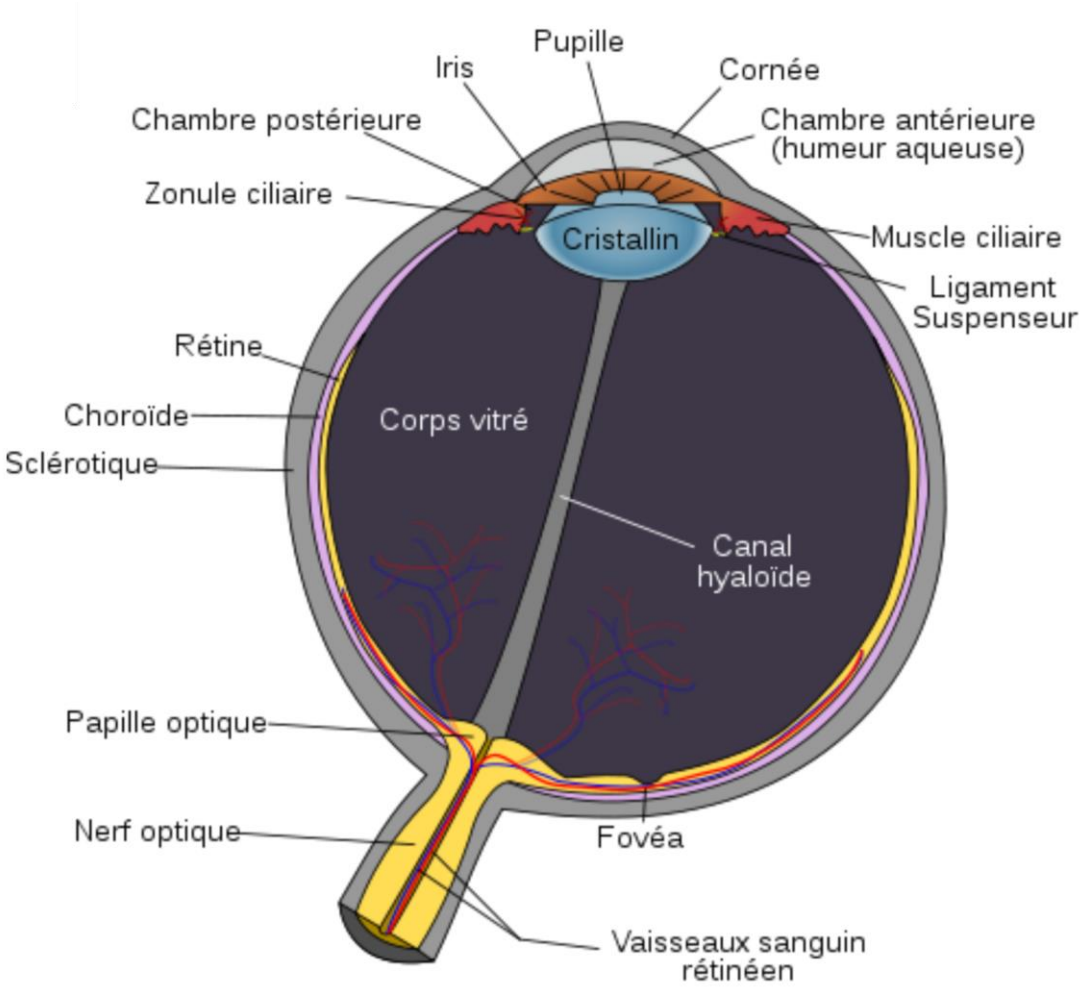
INTRODUCTION AU TRAITEMENT D'IMAGES

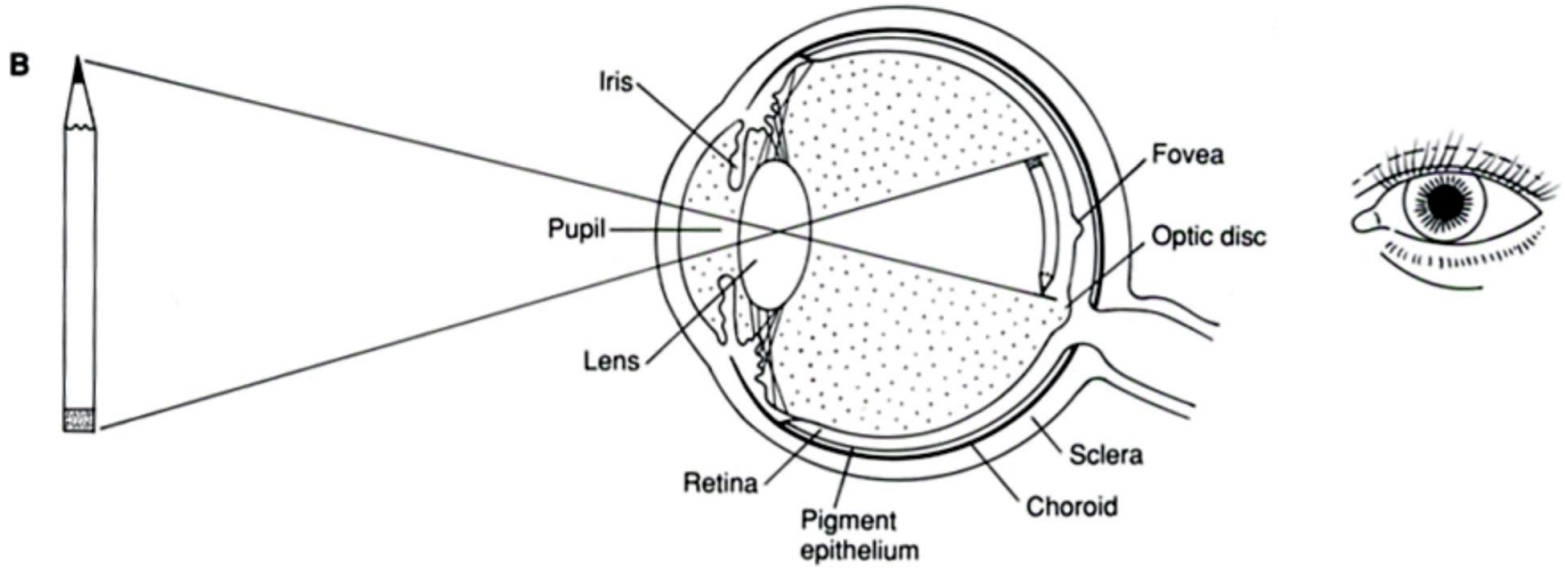
1ere Partie

Donatello Conte

Structure de l'oeil

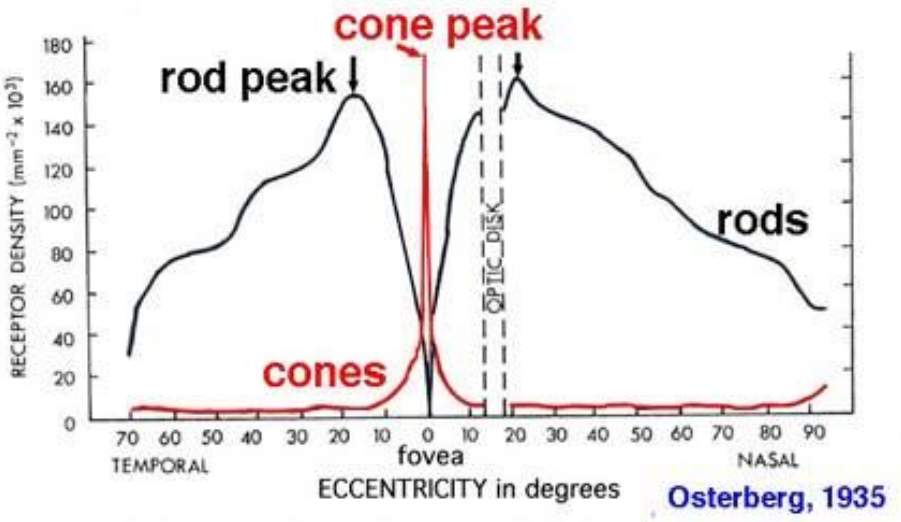


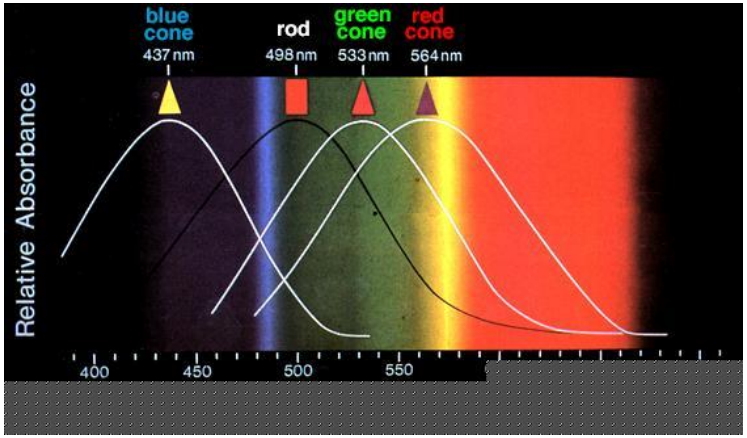
Fonctionnement de l'oeil



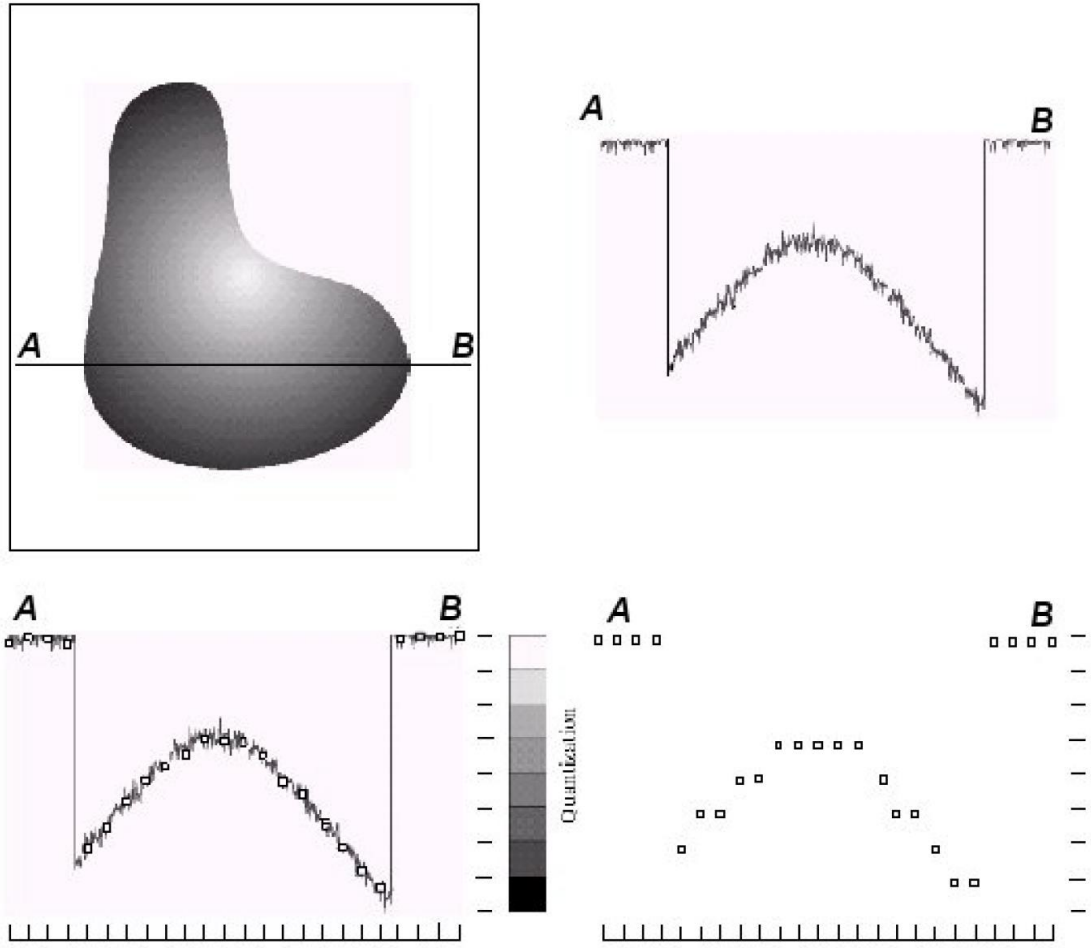
Cônes et bâtonnets

- Cellules photo-réceptrices sensibles spécifiquement à des longueurs d'onde:
  - cônes
  - bâtonnets





Acquisition d'images



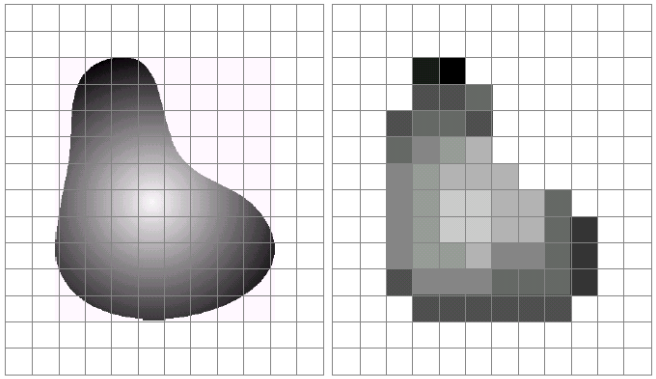
Acquisition d'images

Echantillonnage et quantification

- **Résolution spatiale**
- Le plus petit détail discernable

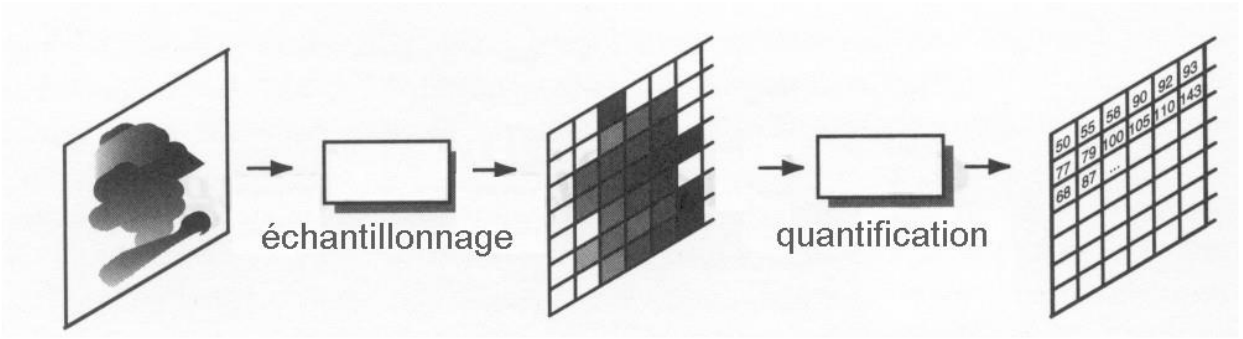
- **Résolution tonale** (de tons de gris)
- Le plus petit changement discernable

- Une image a donc une résolution spatiale de M x N pixels et une résolution de tons de gris de K bits ou de L niveaux ou tons
- Le nombre de bits pour stocker une image est donc : $b = M \times N \times K$



a b

FIGURE 2.17 (a) Continuos image projected onto a sensor array. (b) Result of image sampling and quantization.



Résolution d'une image

Echantillonnage et quantification

- **Résolution spatiale**


256x256


128x128


64x64


32x32

- **Résolution tonale**


6 bits


4 bits

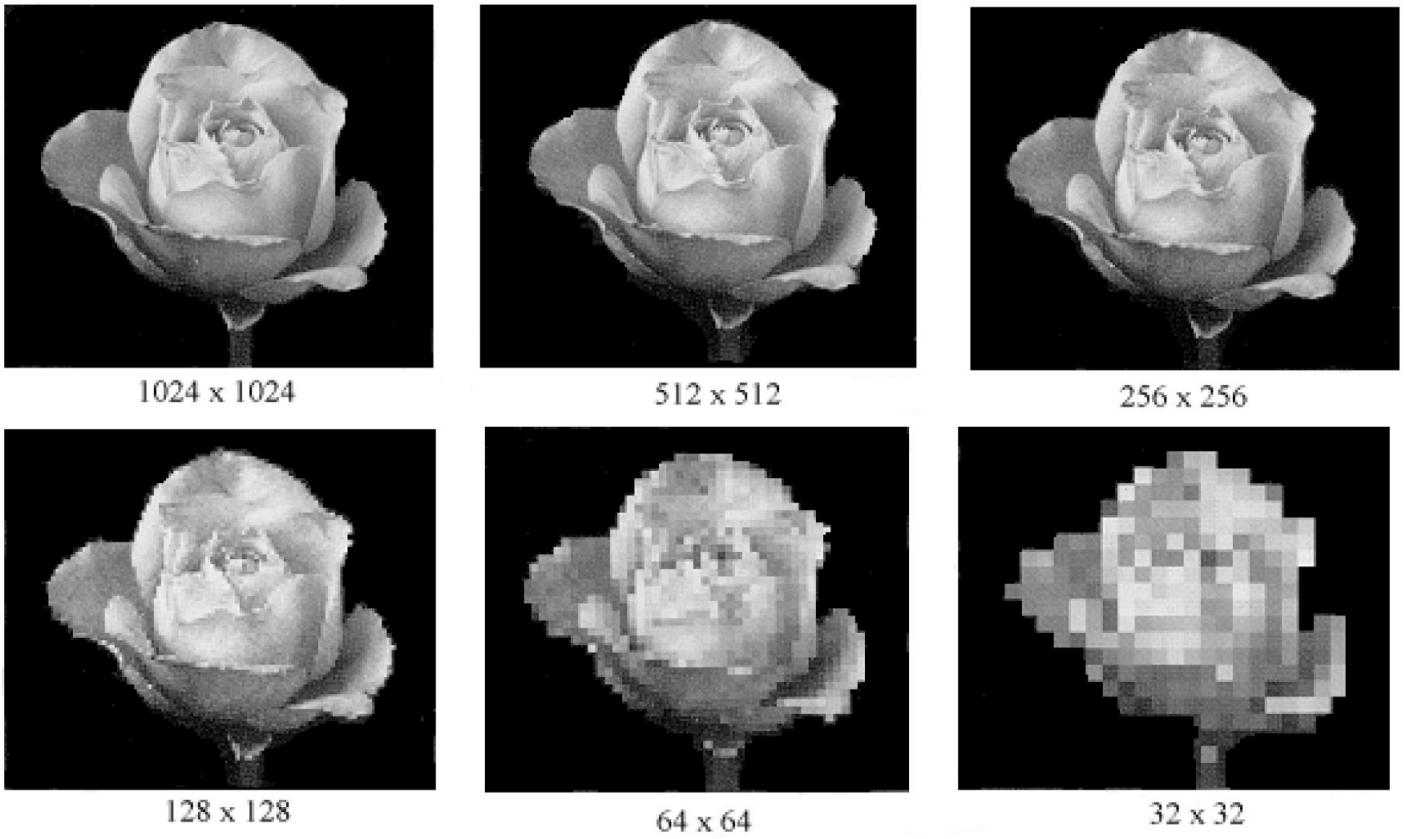

3 bits


2 bits


1 bit

Résolution (spatiale) d'une image

- La dimension du pixel change



Résolution (spatiale) d'une image

- La dimension du pixel ne change pas

Résolution (tonale) d'une image

- Attention aux apparences (dans l'être humain il y a le cerveau qui intègre l'information)

256 couleurs (8 bit par pixel)

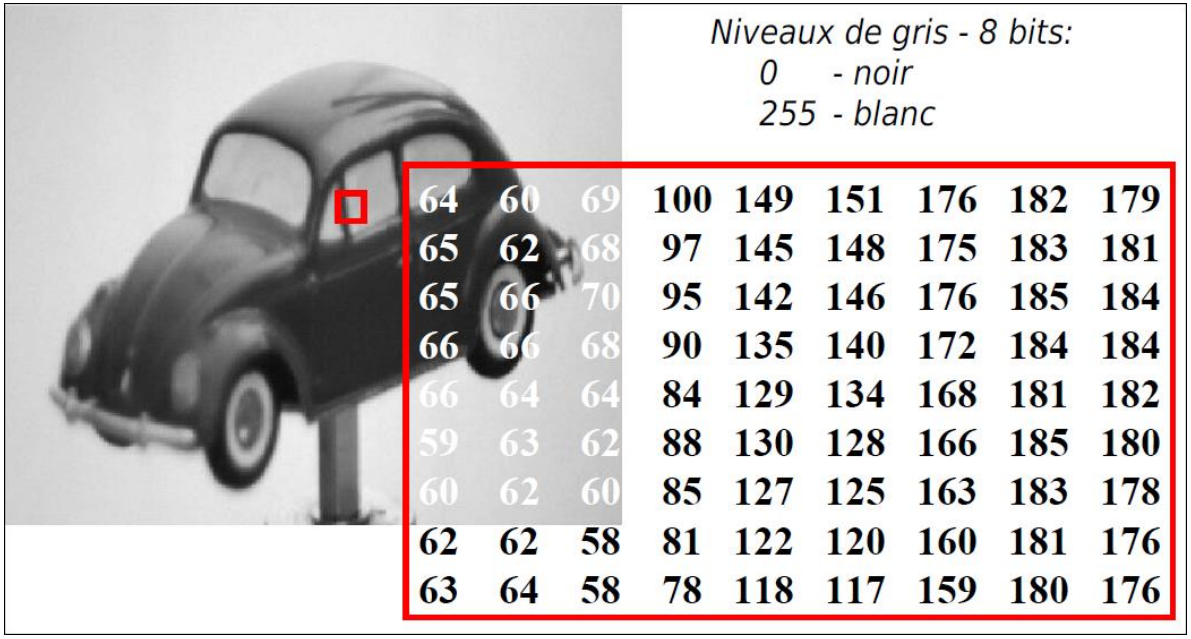
2 couleurs (1 bit par pixel)

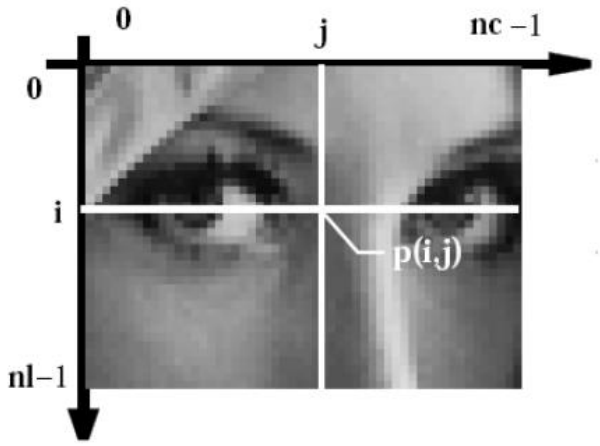
Qu'est ce qu'une image ?

Une image numérique est un tableau de pixel.

Un pixel s est décrit par :

- ses coordonnées dans l'image (i, j)
- sa valeur $I(i, j)$, représentant sa couleur (ou son niveau de gris)





Qu’est ce qu’une image ?

	x = 58	59	60	61	62	63	64	65	66	67
y = 41	210	209	204	202	197	247	143	71	64	80
42	206	196	203	197	195	210	207	56	63	58
43	201	207	192	201	198	213	156	69	65	57
44	216	206	211	193	202	207	208	57	69	60
45	221	206	211	194	196	197	220	56	63	60
46	209	214	224	199	194	193	204	173	64	60
47	204	212	213	208	191	190	191	214	60	62
48	214	215	215	207	208	180	172	188	69	72
49	209	205	214	205	204	196	187	196	86	62
50	208	209	205	203	202	186	174	185	149	71
51	207	210	211	199	217	194	183	177	209	90
52	208	205	209	209	197	194	183	187	187	239
53	204	206	203	209	195	203	188	185	183	221
54	200	203	199	236	188	197	183	190	183	196
55	205	210	202	203	199	197	196	181	173	186

62	64	52	93	52
58	68	61	51	56
75	61	58	60	60
122	63	58	64	66
105	62	57	64	63

Qu'est ce qu'une image ?

La valeur $I(i,j)$ d'un pixel $s = (i,j)$ représente son intensité lumineuse

En niveau de gris

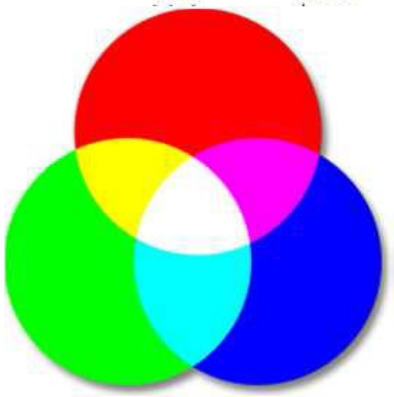
binaire : $I(i,j) = 0$ noir ou $I(i,j) = 1$ blanc

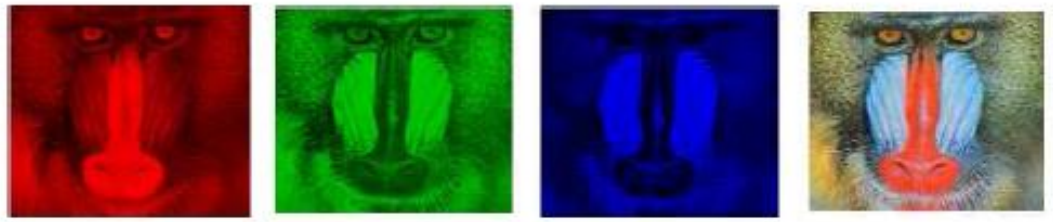
codage 8 bits : (le plus classique) $I(i,j) = 0, ..., 255$ du plus foncé au plus clair

En couleur

codage dans l'espace RGB : trois intensités lumineuses rouge, vert, bleu.

codage 24 bits : $I_R(i,j) = 0, ..., 255$; $I_V(i,j) = 0, ..., 255$; $I_B(i,j) = 0, ..., 255$



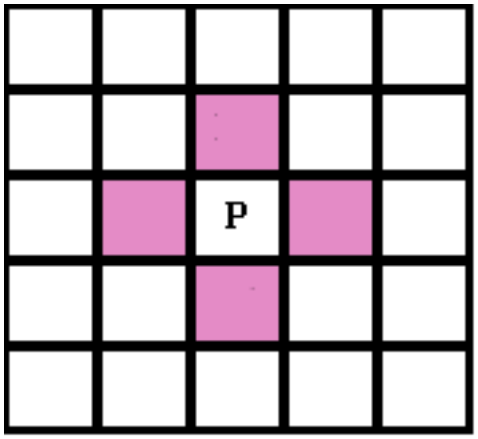


Modèle additif

Voisinage entre pixels

- Pour un pixel p ayant coordonnées (x, y), ces 4-voisins (dénotés $N_4(p)$) ont les coordonnées suivantes

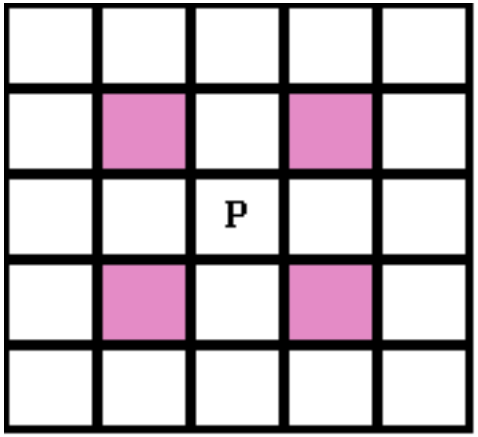
- (x + 1, y)
- (x - 1, y)
- (x, y + 1)
- (x, y - 1)



Voisinage entre pixels

- L’ensemble des 8-voisins de p (dénotés $N_8(p)$) è composé par les 4-voisins plus les pixels de coordonnées suivantes

- (x - 1, y - 1)
- (x - 1, y + 1)
- (x + 1, y + 1)
- (x + 1, y - 1)



Connexité

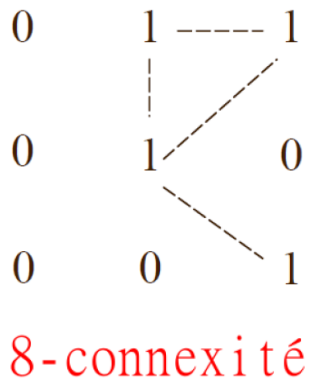
- La relation de voisinage est liée à la notion de connexité entre pixels

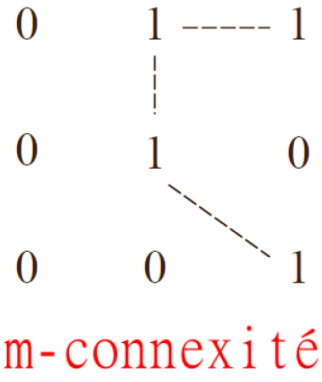
- Deux pixels sont connexes s'ils sont voisins (4- ou 8-voisins) et leur niveau de couleur satisfait un certain critère de similarité

- Le critère de similarité peut par exemple consister à appartenir à un certain ensemble de valeurs d'intensité V
  - Dans une image binaire le critère de similarité est, par exemple, l'appartenance du pixel au niveau noir ou blanc

Connexité

- Sur la base du voisinage on définit les types de connexité suivants:
  - 4-connexité: deux pixels p et q qui son 4-voisins ayant un niveau de couleur dans V
  - 8-connexité: deux pixels p et q qui sont 8-voisins ayant un niveau de vouleur dans V
  - m-connexité: deux pixels p et q qui ont un niveau de couleur dans V et
    - q est dans $N_4(p)$, ou
    - q est dans $N_8(p)$ mais $N_4(p) \cap N_4(q) = \varnothing$
  - La m-connexité a été introduite pour avoir des chemins de connexion uniques entre les pixels



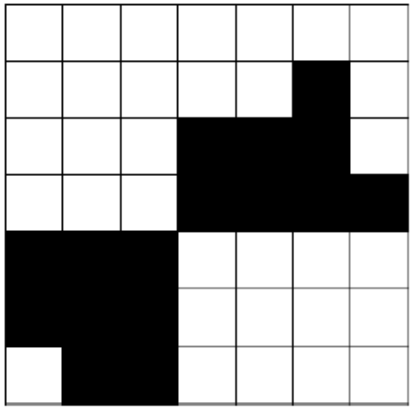


Autres définitions

- Deux pixels 4-,8- ou m-connexes sont entre eux *adjacents*
- Deux ensemble S1 et S2 de pixels sont adjacents s'il existe au moins un pixel dans S1 qui est adjacent à un pixel de S2
- Un chemin du pixel p(x,y) au pixel q(s,t) est une séquence de pixels de coordonnées:
  - (x_0,y_0), (x_1,y_1), …, (x_n,y_n)
  - $(x_0,y_0)=(x,y)$
  - $(x_n,y_n)=(s,t)$
  - (x_i,y_i) est voisin de (x_{i+1},y_{i+1})
- *n* est la longueur du chemin
- Etant donné deux pixels p et q d'un ensemble S, p est connexe à q dans S s'il existe un chemin entre p et q entièrement composé de pixels de S

Autres définitions

- Pour chaque pixel p de S, l'ensemble de pixels de S connexes à p est une composante connexe de S
- Toutes couples de pixels d'une composante connexe est composé de pixels connexes
- Dans la définition de composante connexe il est important de définir quel type de voisinage on utilise

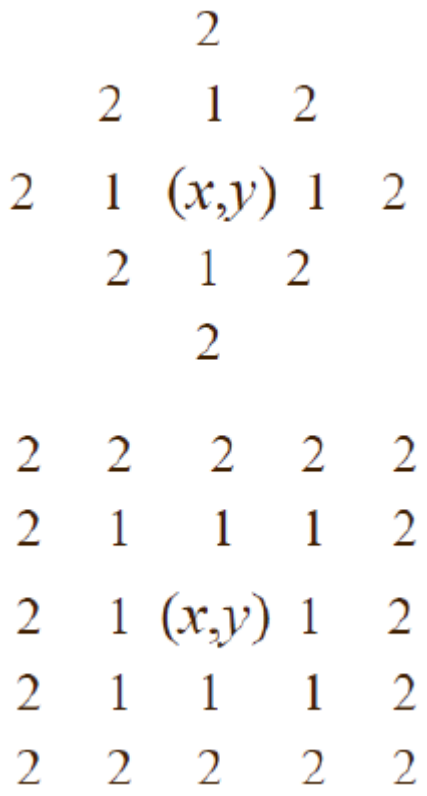


Régions et bords

- Un sous-ensemble R d'une image est une *région* de l'image si R est une composante connexe
- Son *contour* est l'ensemble de pixels de la région qui ont un ou plusieurs pixels adjacent qui n'appartiennent pas à R
- Si R est tout l'image, le contour est formée par la première et la dernière ligne, et par la première et la dernière colonne
- Le contour d'une région (finie) est un chemin fermé (caractéristique globale de la région)
- Un *bord* (*edge* en anglais) est une caractéristique locale, étant la mesure de discontinuité d'un niveau de gris dans un point

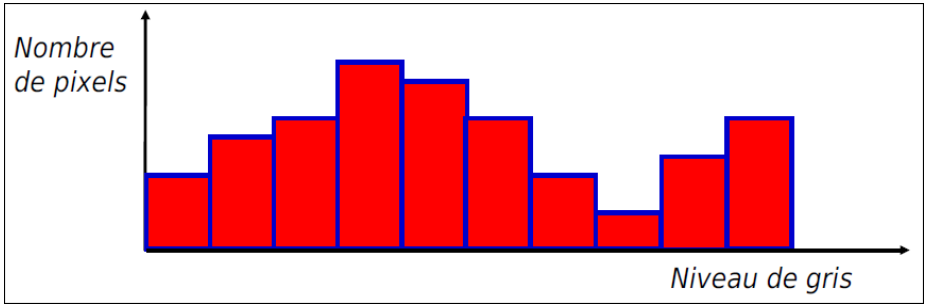
Distance entre pixels

- Considérons les pixels p(x,y) et q(s,t)
- La distance euclidienne est définie comme suit:
  - $D_e(p,\mathrm{q}) = \sqrt{(x-s)^2 + (y-t)^2}$
- La distance D_4 (city block) est définie comme suit:
  - $D_4(p,\mathrm{q}) = |x-s| + |y-t|$
- La distance D_8 (chessboard) est définie comme suit:
  - $D_8(p,\mathrm{q}) = \max(|x-s| + |y-t|)$
- La distance D_4 entre deux points coïncide avec la longueur du plus petit 4-chemin entre les deux points
- La distance D_8 entre deux points coïncide avec la longueur de plus petit 8-chemin entre les deux points



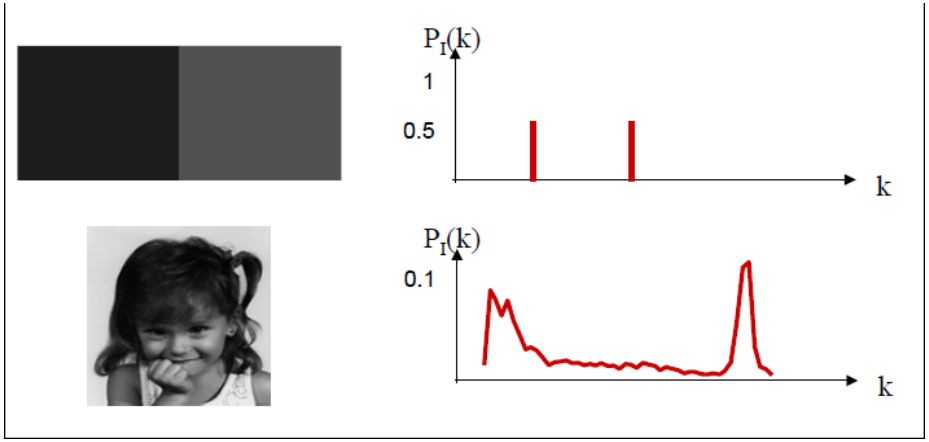
Histogramme des niveaux de gris

- 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.
- *Dynamique d'une image = [valeur_min,valeur_max]*



Histogramme normalisé

P(k)=H(k) / Nb_pixels



Exemples

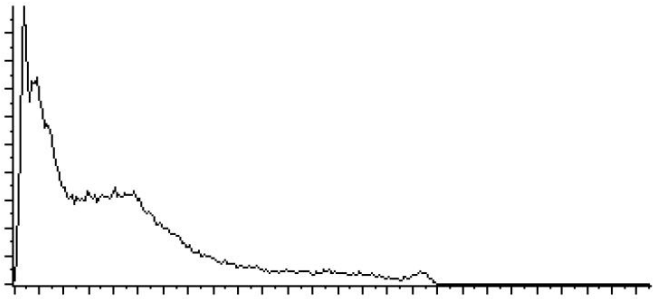


Image sombre

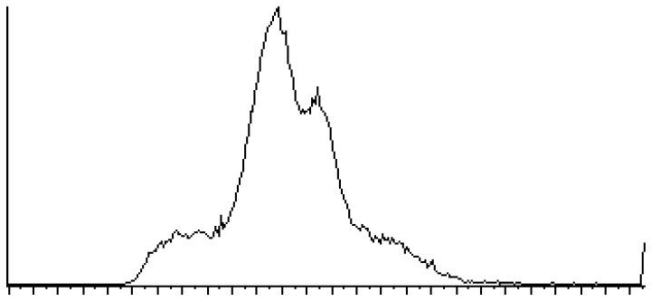
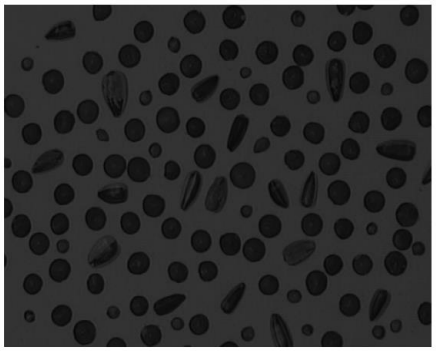


Image peu contrastée

Exemples



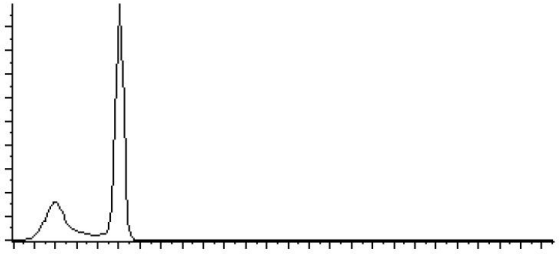


Image sombre et peu contrastée

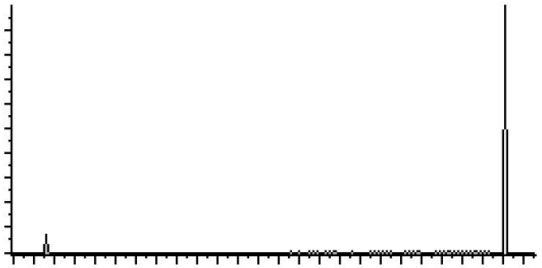
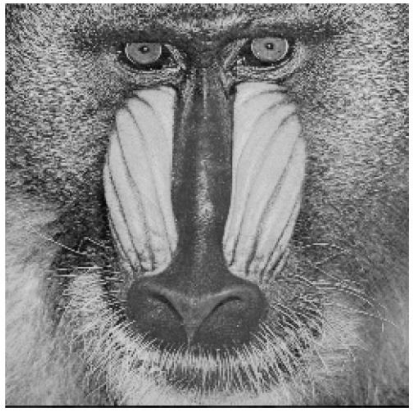


Image claire et peu contrastée

Exemples

Image équilibrée



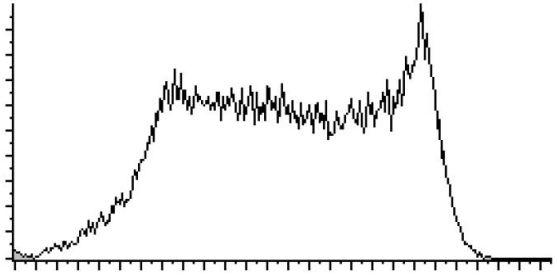


Image équilibrée mais un peu moins contrastée

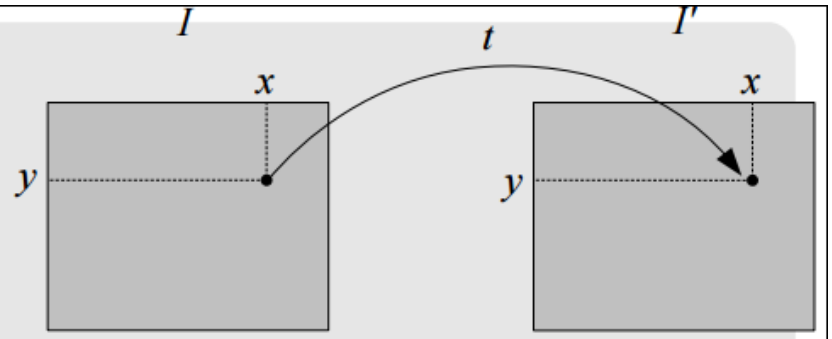
Types de traitement

- **Transformations ponctuelles**

$$I(x,y) \xrightarrow{t} I'(x,y)=t\big(I(x,y)\big)$$

Ex. seuillage, ajustement luminosité/contraste

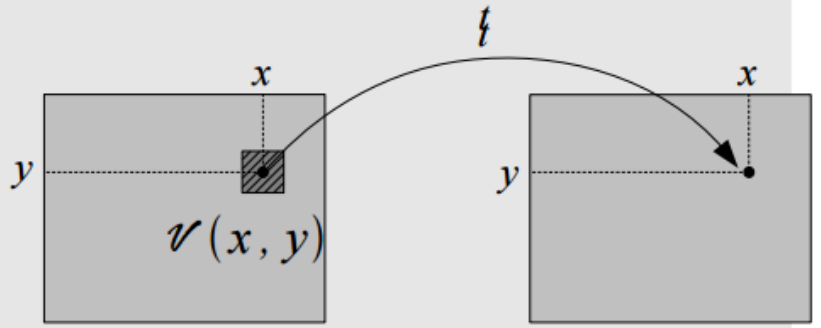
opérations algébriques, manip. d'histogramme



- **Transformations locales**

$$I(x,y) \xrightarrow{t} I'(x,y)=t\Big(I\big(\mathscr{V}(x,y)\big)\Big)$$

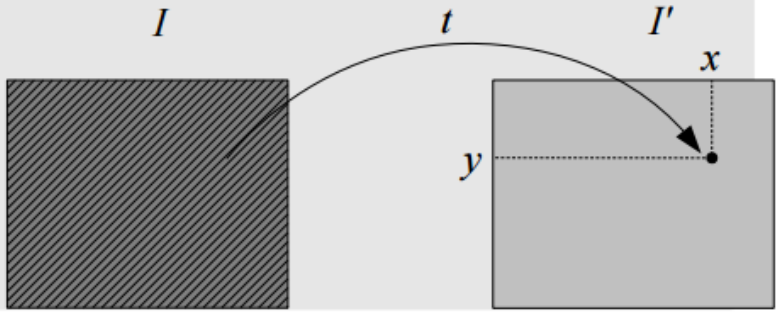
Ex. filtrage



- **Transformations globales**

$$I(x,y) \xrightarrow{t} I'(x,y)=t\big(I\big)$$

Ex. transformation dans l'espace de Fourier



Traitements ponctuels

- Une opération générique peut s'écrire:
  - g(x,y)=t[f(x,y)]
  - g est l'image de sortie, f l'image en entrée, T est un opérateur sur f, calculé à partir de la valeur du pixel (x,y)

- Les opération ponctuels sont divisées en
  - Homogènes: les operateurs ne dépendent pas de la position du pixel
    - Ex. : ajustement de luminosité
  - Non-homogènes: les operateurs dépendent de la position du pixel
    - Ex. : somme entre images

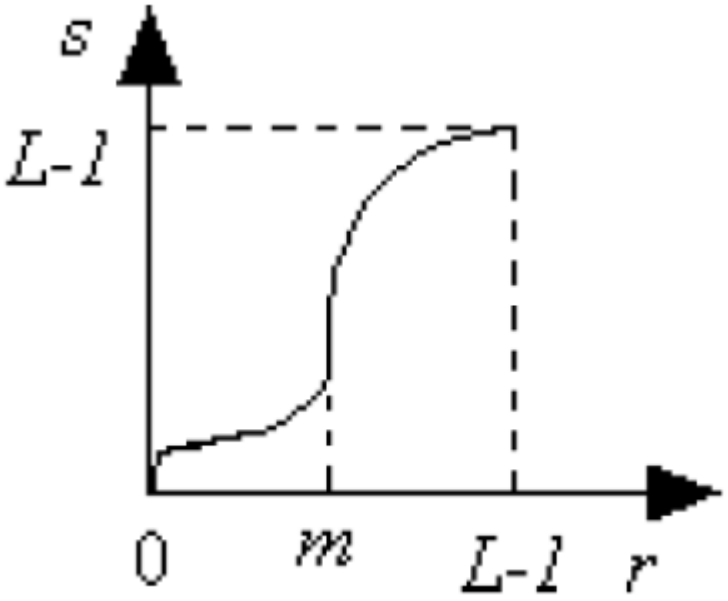
Opérations homogènes

- Le traitement s'applique à chaque pixel de l'image

- Etant l'opérateur dépendant que du niveau de gris (ou couleur) du pixel, la transformation peut s'écrire
  - s = t(r)
  - r est le niveau de gris (ou couleur) du pixel dans l'image d'entrée et s est le niveau de gris (ou couleur) du pixel dans l'image traitée

Amélioration du contraste

- Les valeurs basses des pixels sont baissées et les valeurs hautes sont rendues plus claires



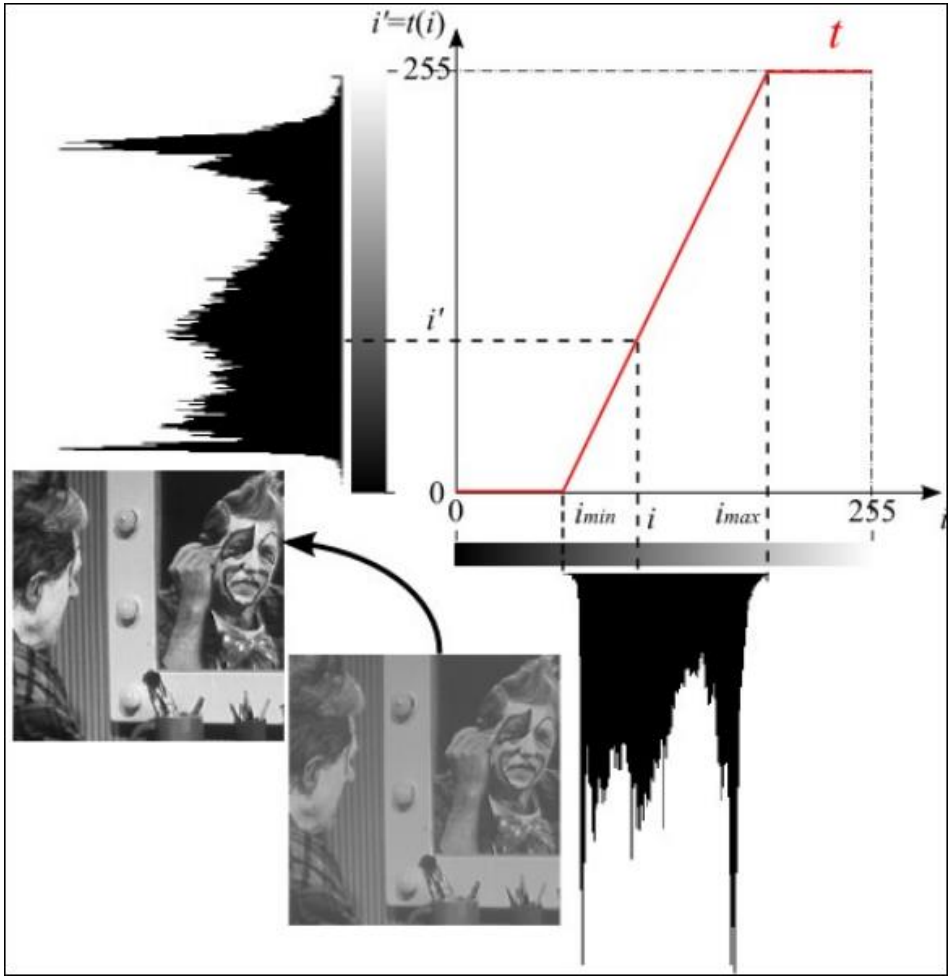
Expansion de la dynamique

- Effet : rehaussement du contraste par expansion de la dynamique
- Remarque : pas d'effet si i_{min} =0 et i_{max} =255

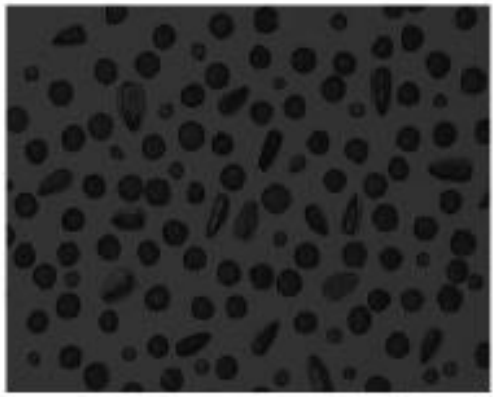
$$i' = \frac{255}{i_{max} - i_{min}}(i - i_{min})$$

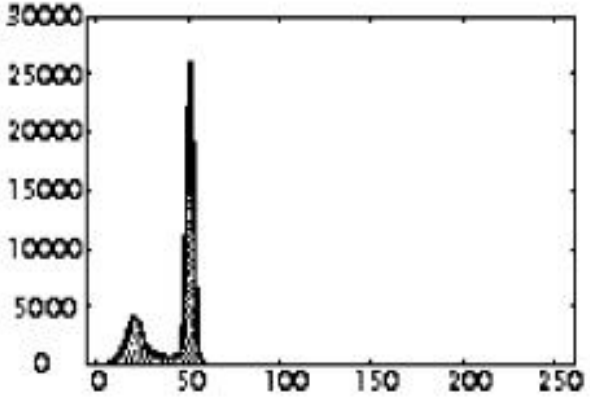
avec

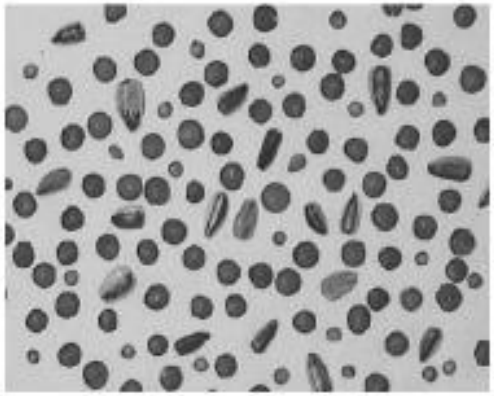
$$\frac{i - i_{min}}{i_{max} - i_{min}} \in [0,1]$$

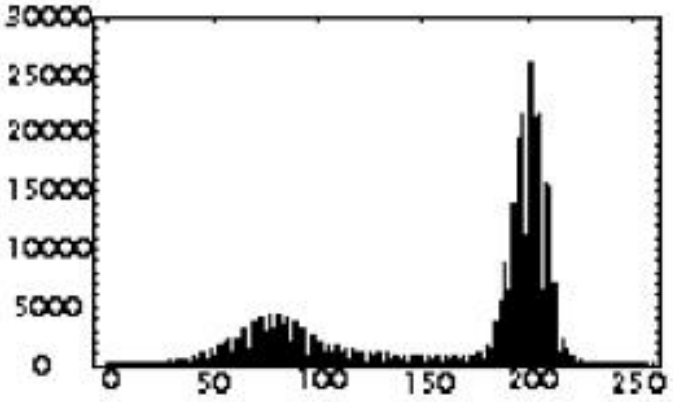


Expansion de la dynamique





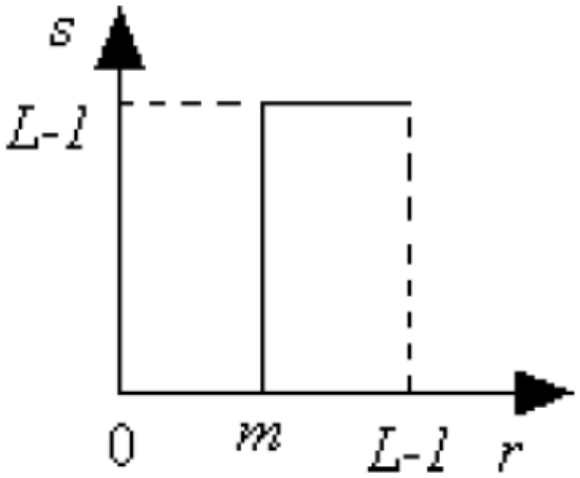




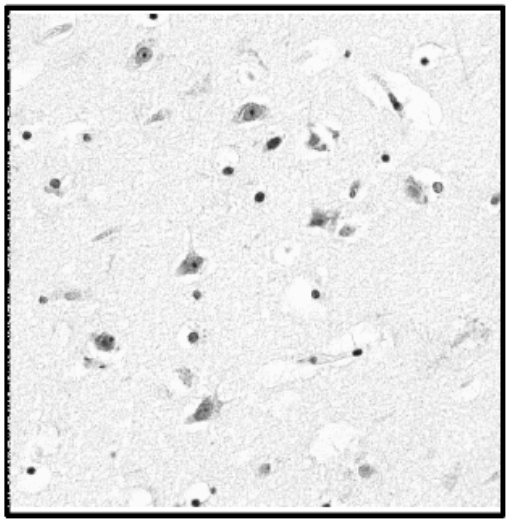
Seuillage

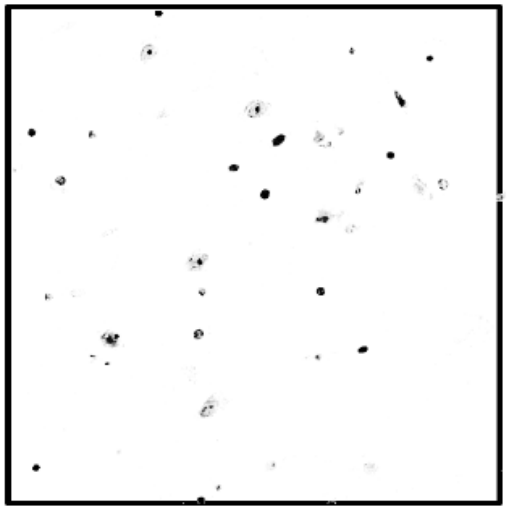
- Dans le cas limite on a l'operation de seuillage

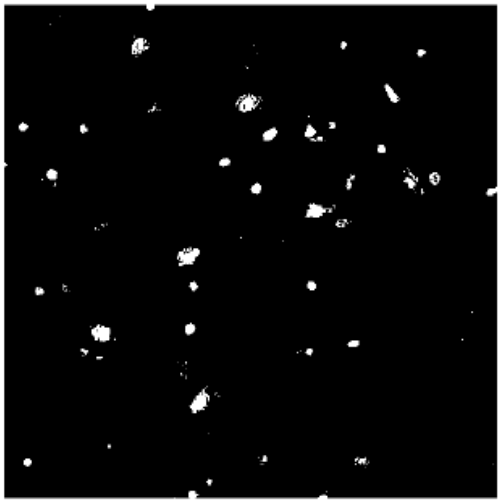
$$s = \begin{cases} 0 & r < m \\ L-1 & r \geq m \end{cases}$$



Exemple





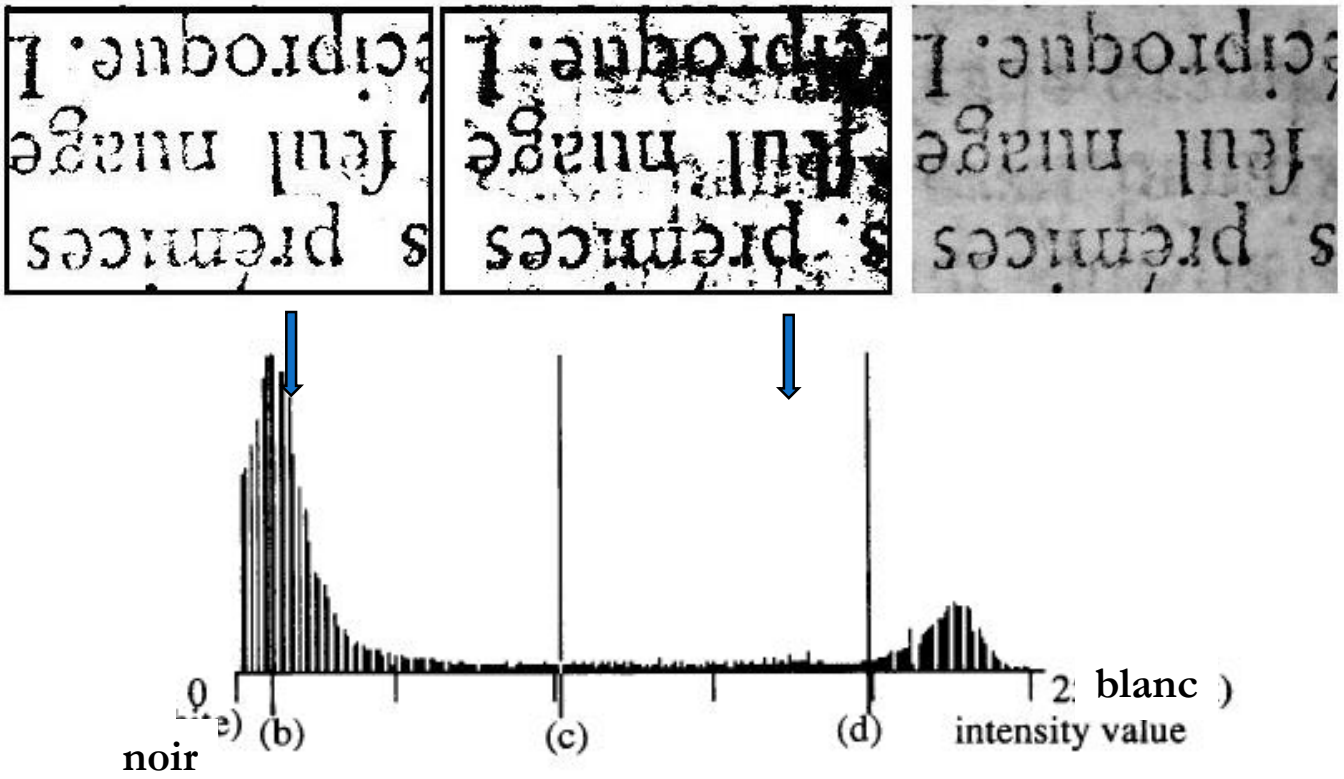


Binarisation par seuillage

- **Seuillage manuel** ➔ C'est la méthode la plus simple et la plus utilisée
- Il y a une relation entre les niveaux de gris d'un pixel et son appartenance ou non à une forme

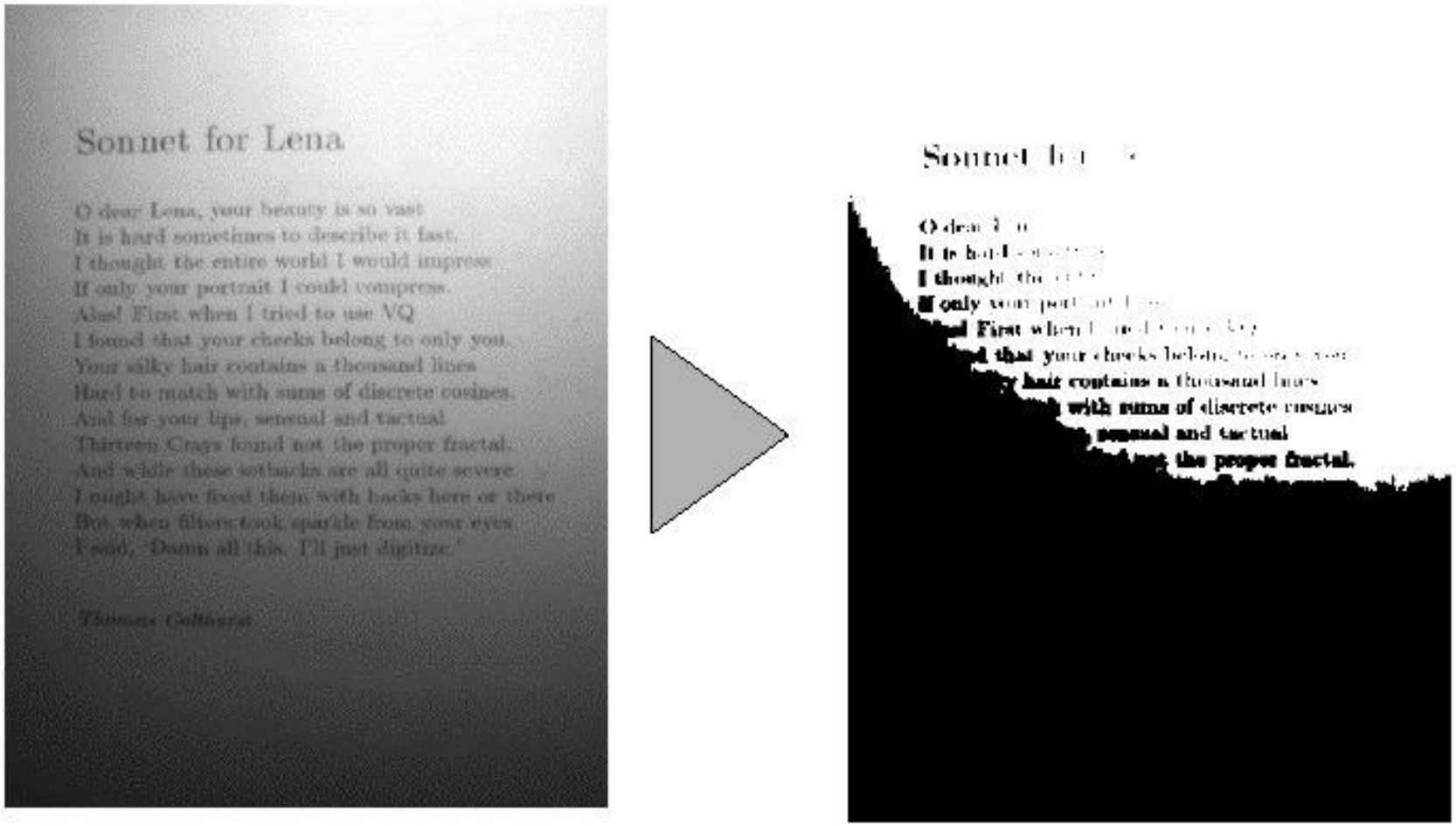
Binarisation par seuil global fixe

- Comment choisir le bon seuil ➔ une multitude de méthodes
  - Différents objectifs : tramage, segmentation, …
  - Otsu, …



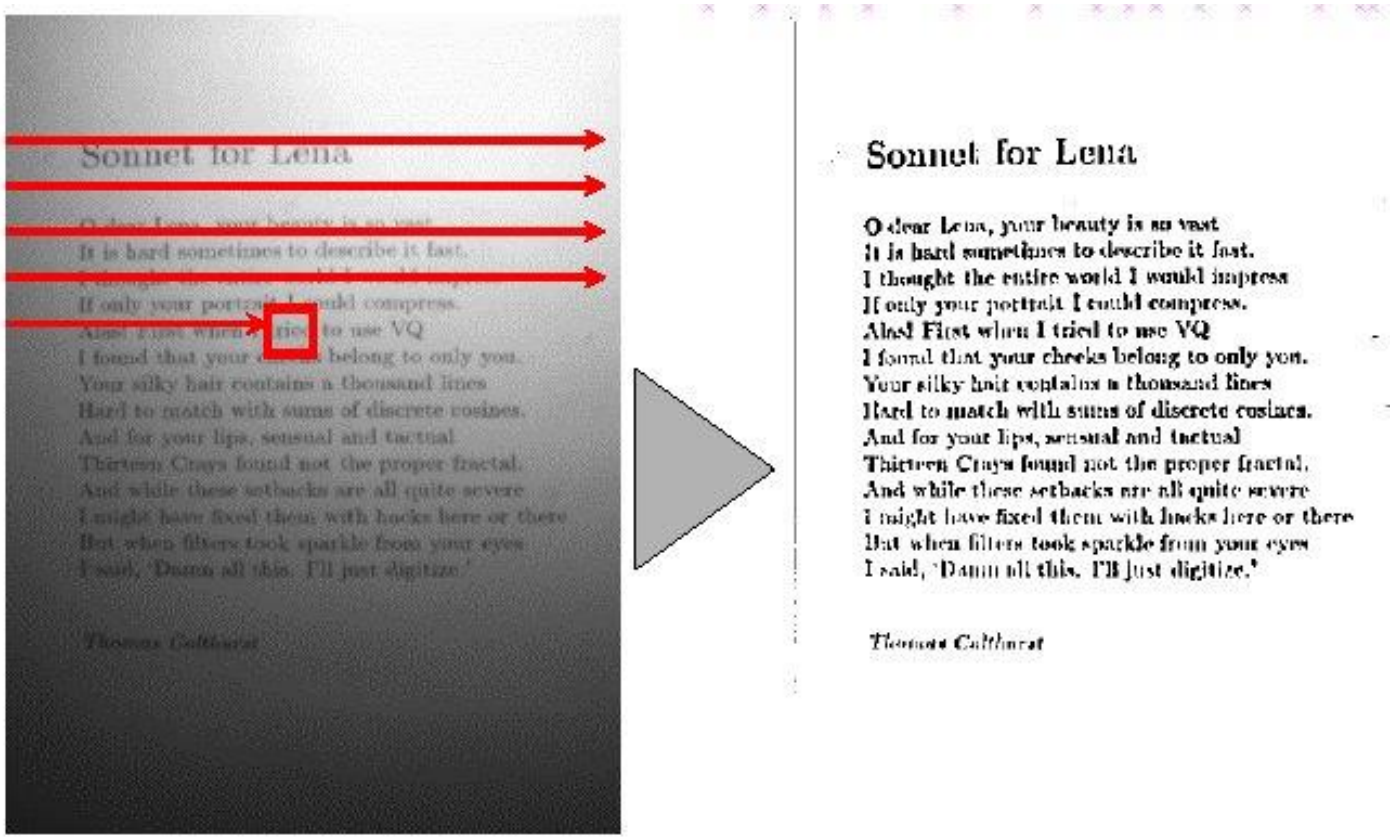
Inconvénient d'un seuil global

- **Idées de solution ?**



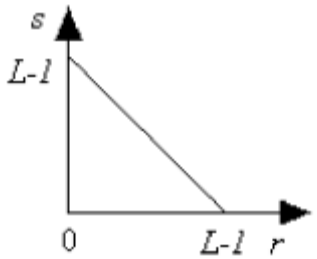
Seuillage adaptatif

- On définit un seuil pour chaque pixel en fonction de son voisinage
- **Le Niblack :** $S = m + ks^2$ avec k= -0,2 | m : moyenne et s : l'écart-type

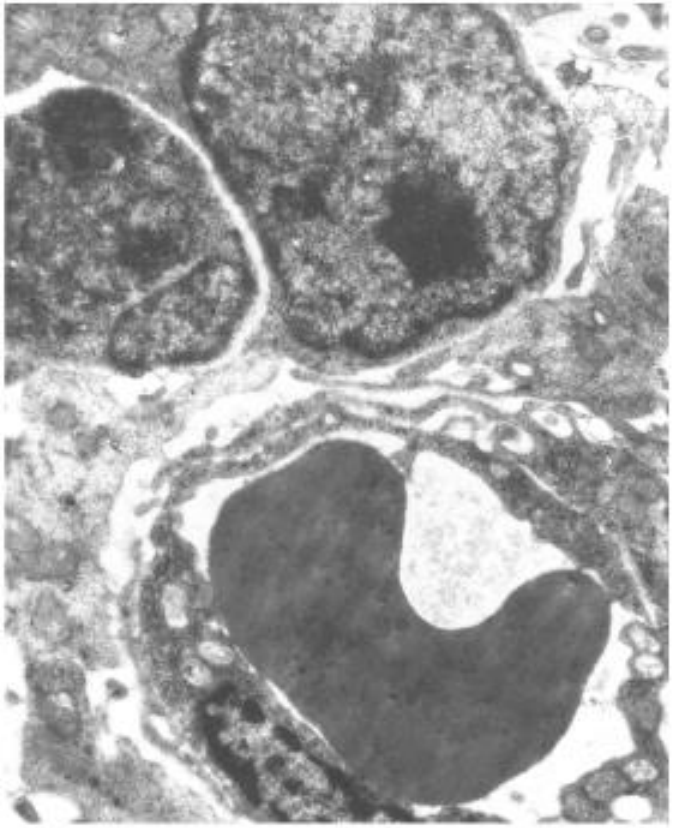


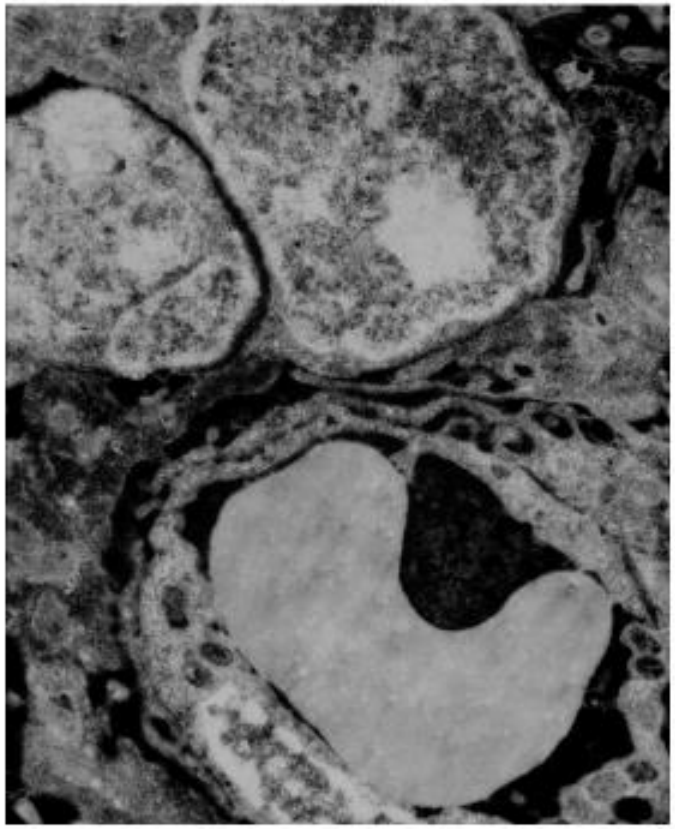
Inverse

- C’est un exemple d’opération inversible
  - s = L – 1 - r



Inverse

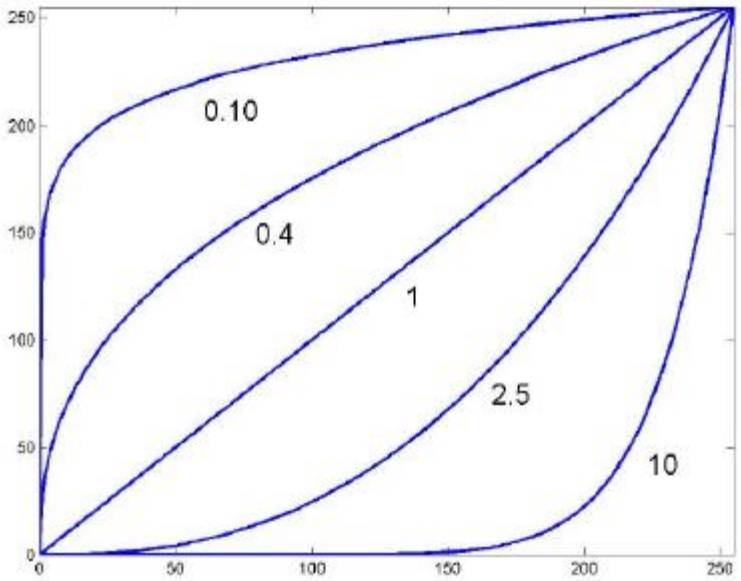




Opération Puissance

$$s = cr^{\gamma}$$

- γ est un paramètre et c est un facteur d'échelle
- Si γ est inférieur à 1 l'opération équivaut à augmenter les valeurs basses de niveau de gris en baissant les valeurs élevées
- Si γ est supérieur à 1 le comportement est inversé



Opération Puissance

$\gamma = 1$

$\gamma = 3$

$\gamma = 4$

$\gamma = 5$

Opérations non homogènes

- Les opérations non homogènes dépendent de la valeur de niveau de gris, mais aussi de la position
- Les plus importantes opérations non homogènes sont les opérations arithmétiques et logiques sur les images
- Ces opérations ont en entrée deux images ou bien le deuxième opérande est une constante

Opérations logiques & arithmétiques

Principe :

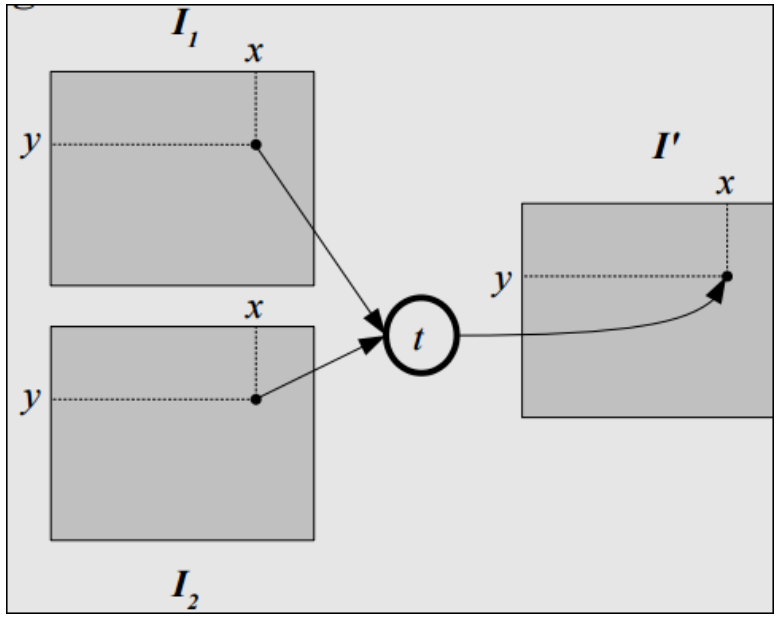
- Appliquer, pixel à pixel, les opérations logiques et arithmétiques classiques à deux (ou plusieurs) images
- Les images opérandes doivent être de même taille
- peuvent être des images constantes.

Exemples :

- Addition, soustraction, ...
- ET logique, OU logique, ...

Problèmes :

- débordements de [0, 255]
- normalisation ...



$$I_1(x,y), I_2(x,y) \xrightarrow{t} I'(x,y) = t\left(I_1(x,y), I_2(x,y)\right)$$

Opérations logiques & arithmétiques

- Addition : $g(x, y) = f_1(x, y) + f_2(x, y)$
- Soustraction : $g(x, y) = f_1(x, y) - f_2(x, y)$
- Multiplication : $g(x, y) = f_1(x, y) \times f_2(x, y)$
- Division : $g(x, y) = f_1(x, y) / f_2(x, y)$
- ET logique : $g(x, y) = f_1(x, y)$ ET $f_2(x, y)$
- OU logique : $g(x, y) = f_1(x, y)$ OU $f_2(x, y)$

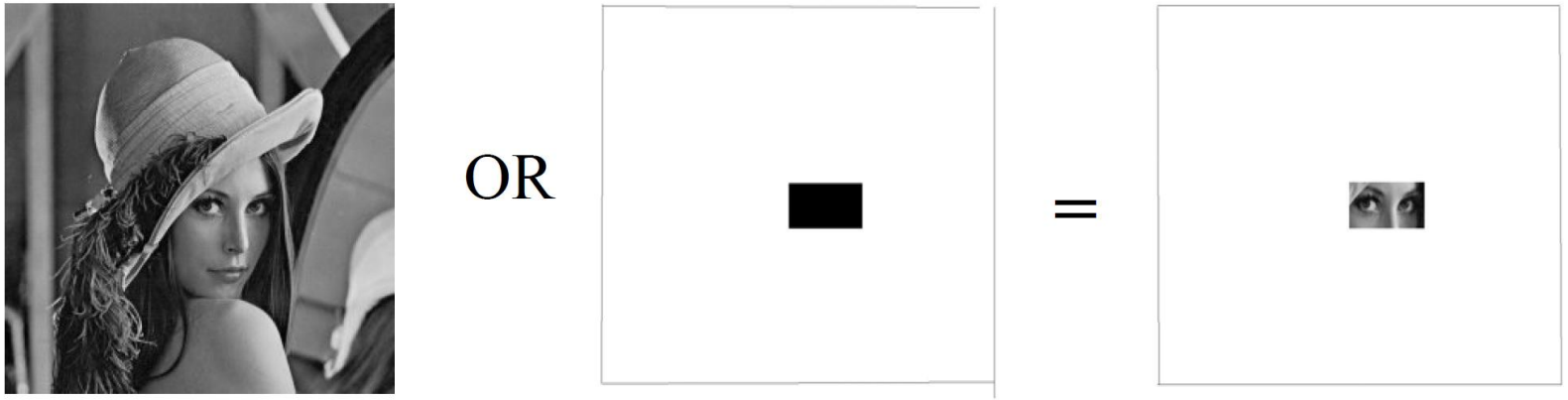
Opérations logiques

- Elle sont utilisées surtout pour l'extraction de régions d'intéret

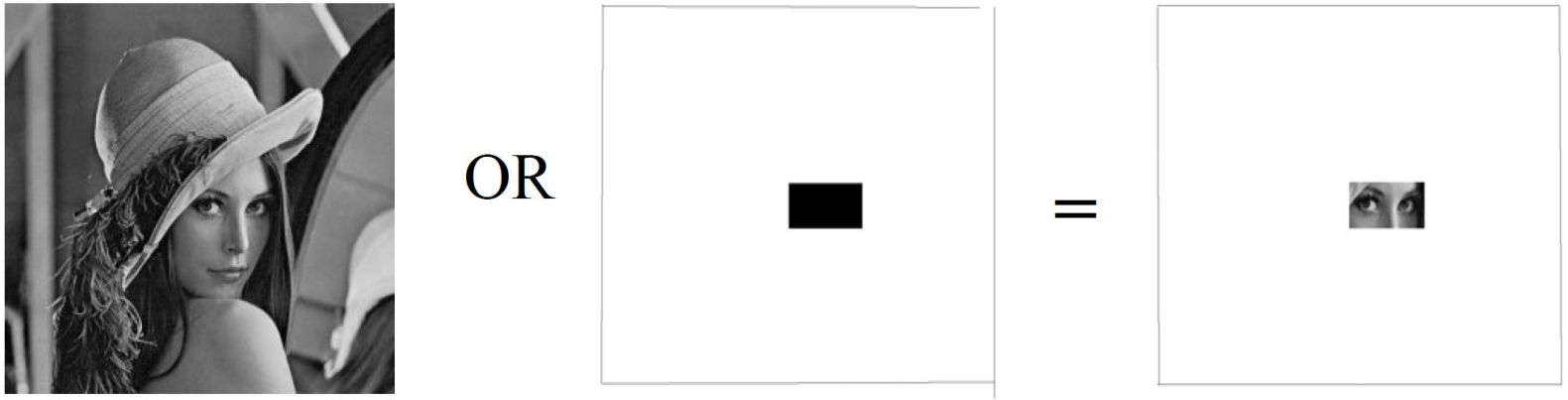
10010011 AND 11111111 = 10010011
10010011 AND 00000000 = 00000000

10010011 OR 11111111 = 11111111
10010011 OR 00000000 = 10010011

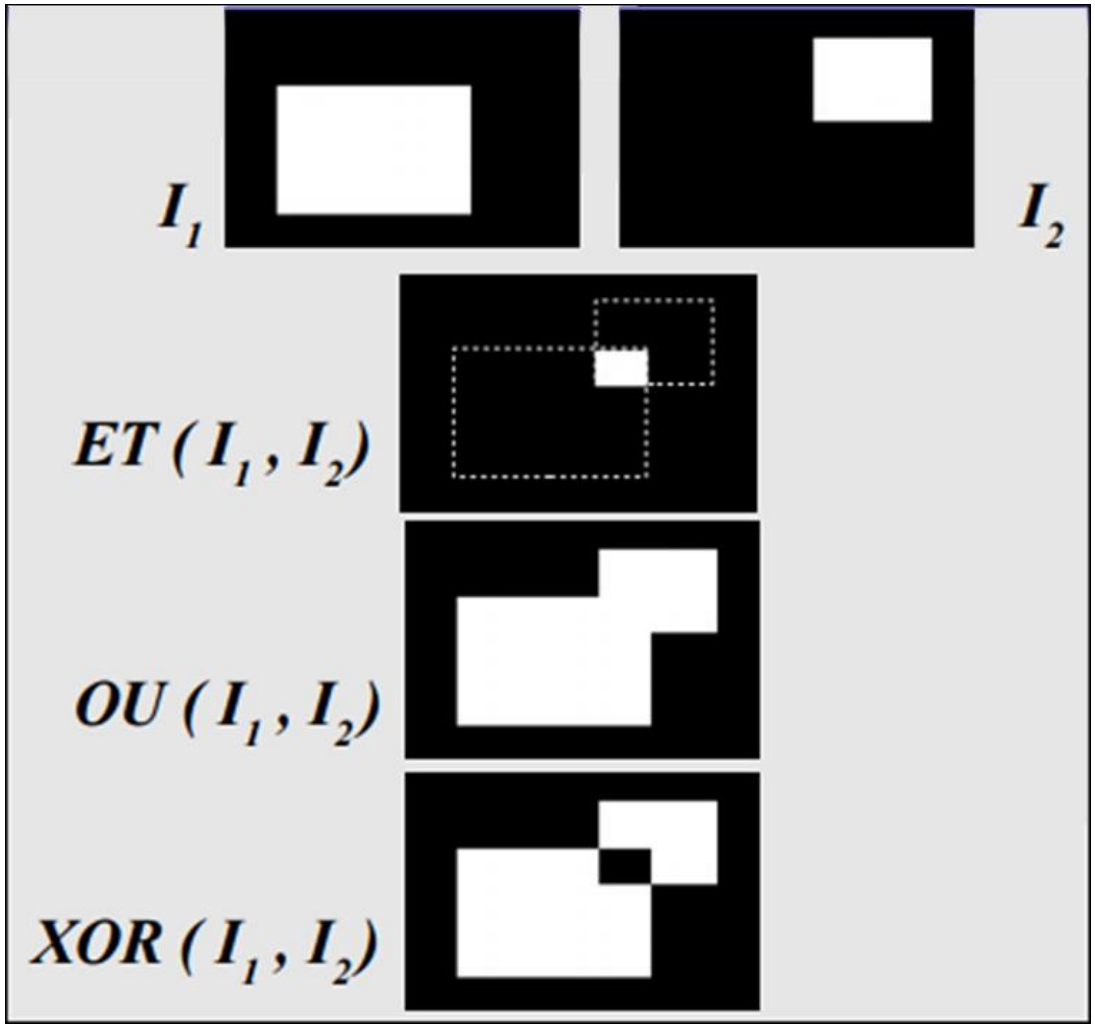
Opérations logiques



Opérations logiques



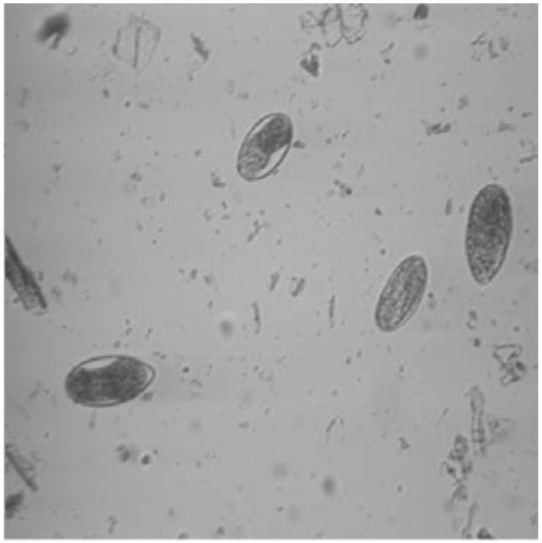
Opérations logiques



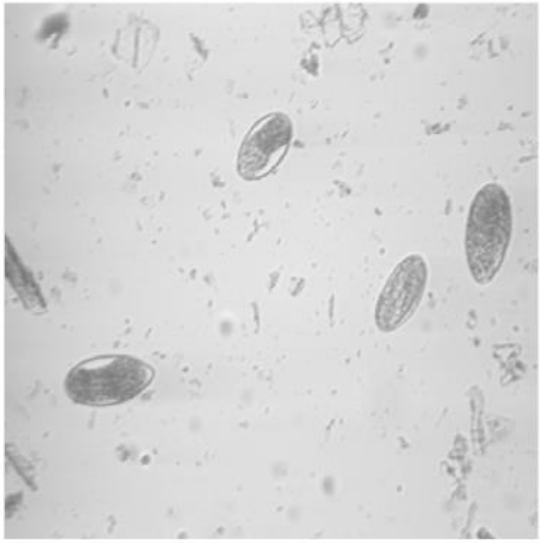
Stratégies de dépassement

- Wrapping (retour circulaire à zero)
- Saturation (remplacer les valeurs qui dépassent par la limite supérieur ou inferieur)
- Décalage des valeurs avant l'opération
- Pré-calcul des valeurs finales (théoriques) minimale et maximale puis recadrage de la dynamique

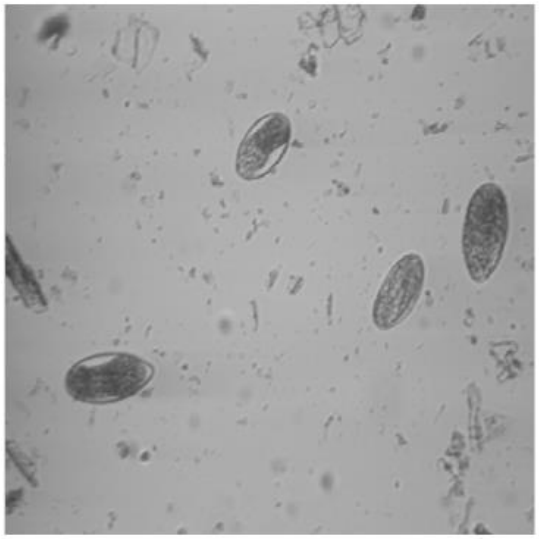
Exemples



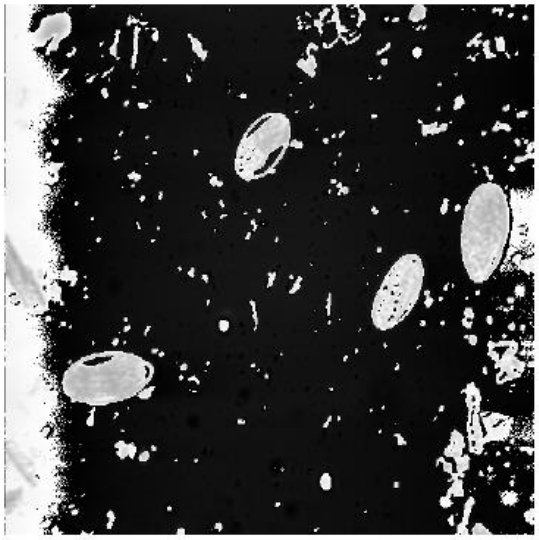
+ 50



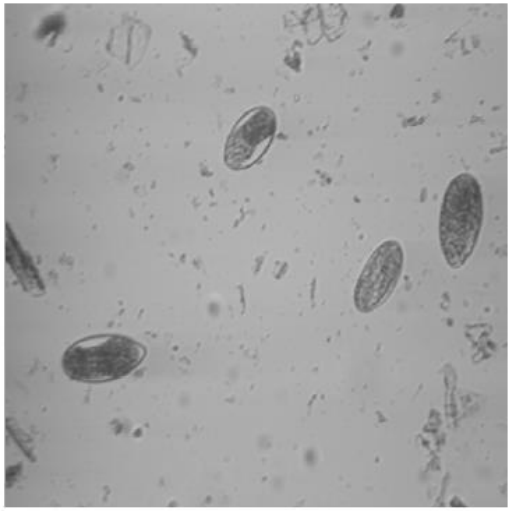
Exemples



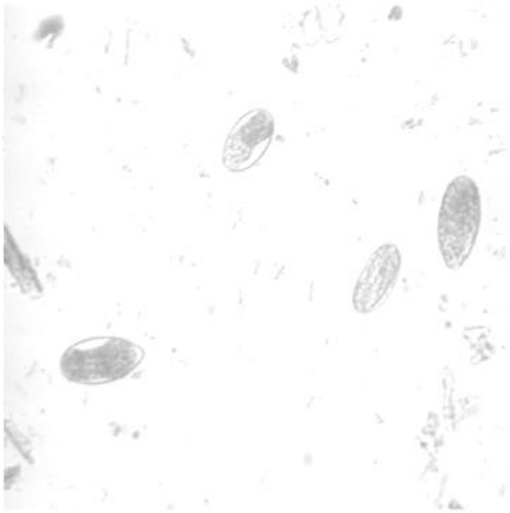
+ 100
(wrapping
overflow)



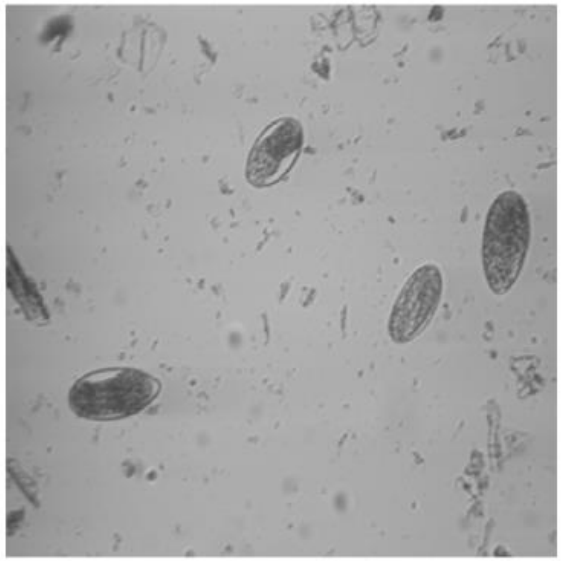
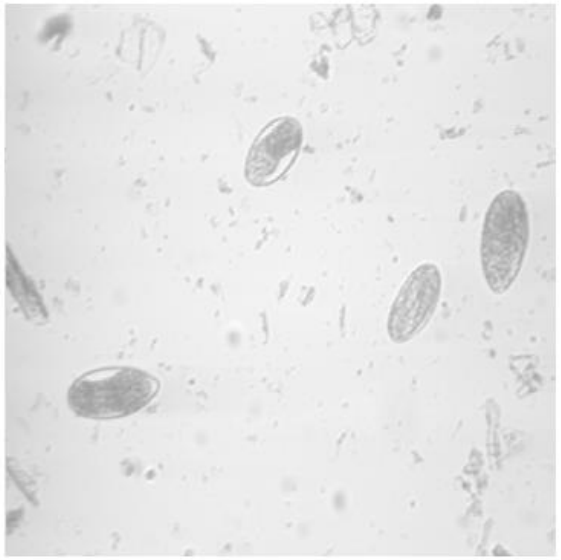
Exemples



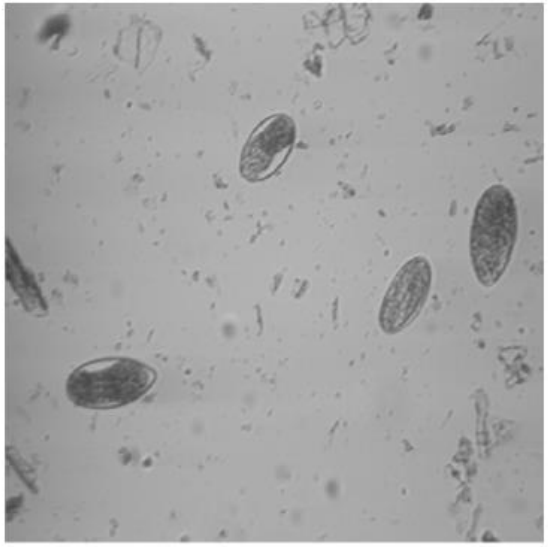
+ 100
(saturating
overflow)



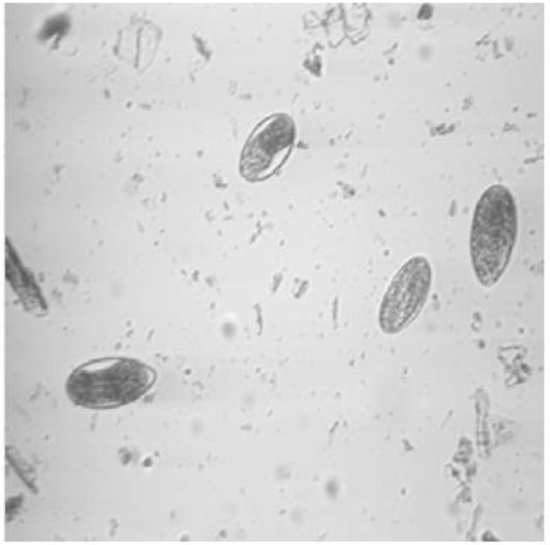
Exemples

( *0.8)+100 

Exemples



*1.3



Opérations arithmétiques

Addition d'images

- **Principe :** $I'(x,y)=I_1(x,y)+I_2(x,y)$ pour tout pixel de coordonnées (x,y)
- Stratégies si dépassement de capacité
  - Décalage des valeurs dans [0, 127] avant addition
  - Saturation : $I'(x,y) = \min (I_1(x,y) + I_2(x,y), 255)$
  - Pré-calcul des valeurs finales (théoriques) minimale et maximale puis recadrage de la dynamique
- Utilisations principales :
  - Augmentation de la luminance d'une image (par addition d‘1 constante ou d‘1 image avec elle-même)
  - Diminution du bruit dans une série d'images

Soustraction d'images

- **Principe :** $I'(x,y) = I_1(x,y) - I_2(x,y)$ pour tout pixel de coordonnées (x,y)
- Stratégies si dépassement de capacité
  - Saturation : $I'(x,y) = \max (I_1(x,y) - I_2(x,y), 0)$
  - Différence absolue : $I'(x,y) = | I_1(x,y) - I_2(x,y) |$
- Utilisations principales
  - Diminution de la luminance d'une image
  - Détection de changements entre images
    - défauts (par comparaison avec une image de référence)
    - mouvements (par comparaison avec une autre image de la séquence)

Opérations arithmétiques

-

Opérations arithmétiques

Multiplication

- **Principe :** $I'(x,y) = I_1(x,y) \times I_2(x,y)$ ou, plus souvent, $I'(x,y) = k \times I(x,y)$
- Stratégies si dépassement de capacité : saturation
- Utilisations principales : amélioration du contraste et de la luminosité d'une image

Combinaison linéaire

- **Principe :** $I'(x,y) = k \times I_1(x,y) + (1-k) \times I_2(x,y)$
- Le facteur k définit la contribution relative de I1 et de I2
- Utilisation principale : superposition d'images

Division

- **Principe :** $I'(x,y)=I_1(x,y) / I_2(x,y)$ ou, plus souvent, $I'(x,y)=I(x,y) / k$
- Problème : éviter la division par 0 ; comment normaliser ?
- Utilisation principale : détection des changements et de leur amplitude