

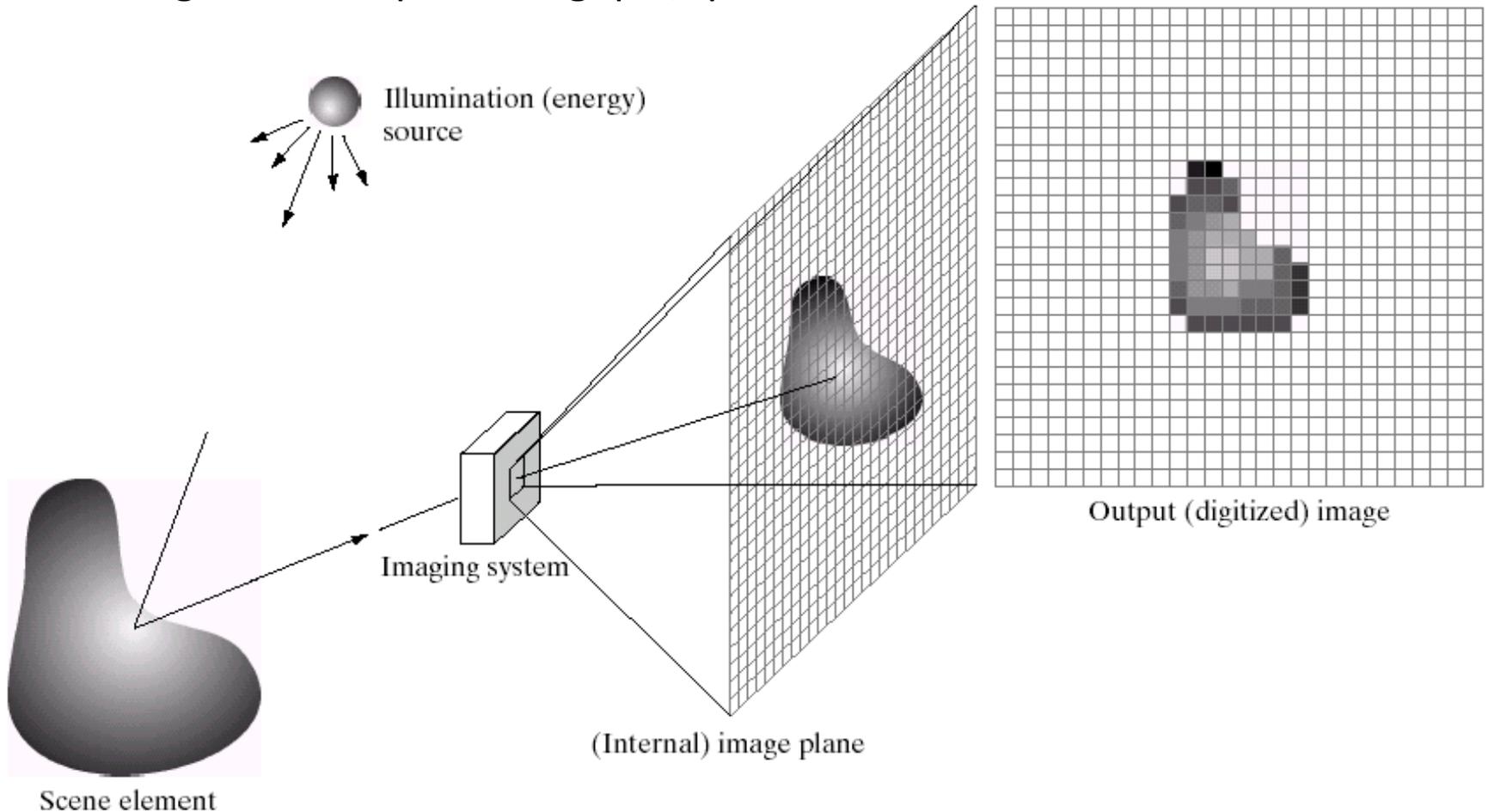
# Traitement d'images

# Qu'est-ce qu'une image numérique?

- Deux types d'images numériques :
  - Images bitmap (*raster*)
  - Images vectorielles
- Les images bitmap peuvent provenir de différentes sources

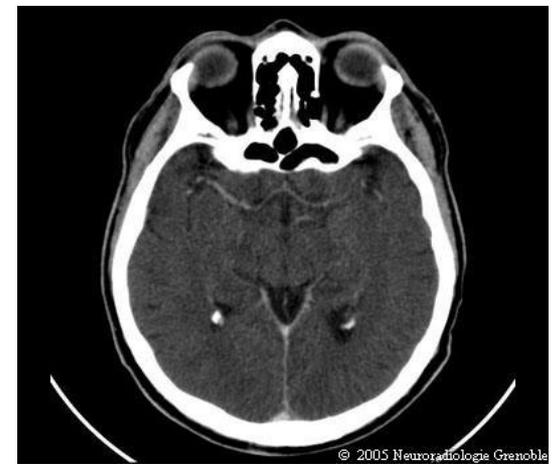
- Image naturelles : utilisation d'un capteur CCD

- Un capteur CCD (Charge-Coupled Device ou détecteur à couplage de charge) est utilisé pour convertir un rayonnement de photons en signal électrique analogique, qui est ensuite discrétisé.

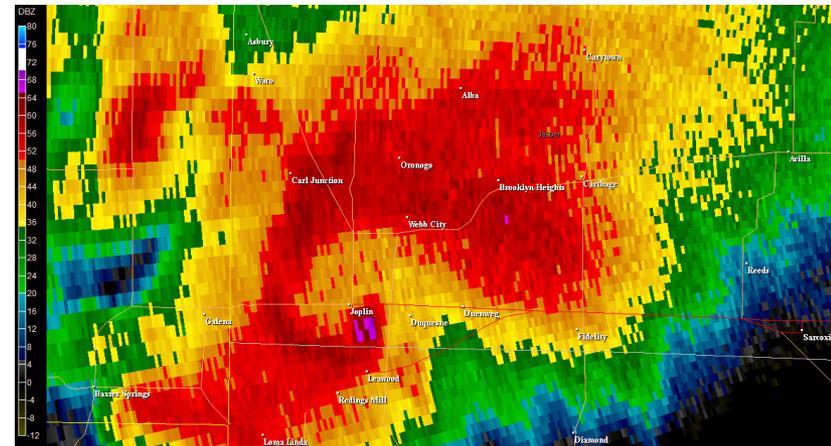


- Ex : appareils photos, scanner, satellites...

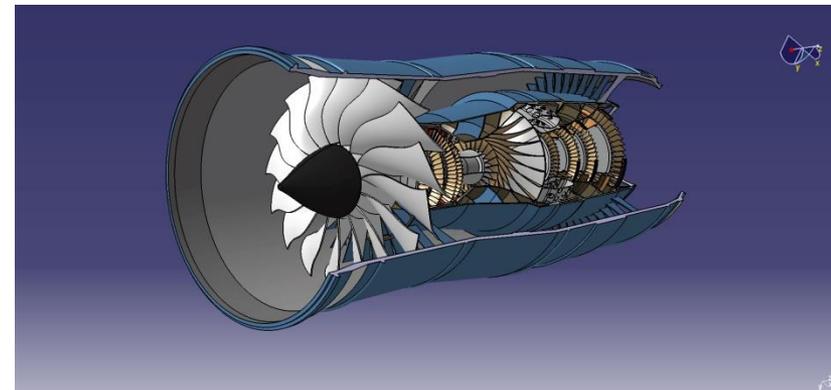
- Image médicales
  - Scanner
  - Imagerie à résonance magnétique
  - Imagerie ultrasonore (échographie)
  - Tomographie à émission de positron
  - Etc.



- Télémétrie
  - Sonar
  - Radar
  - Etc.

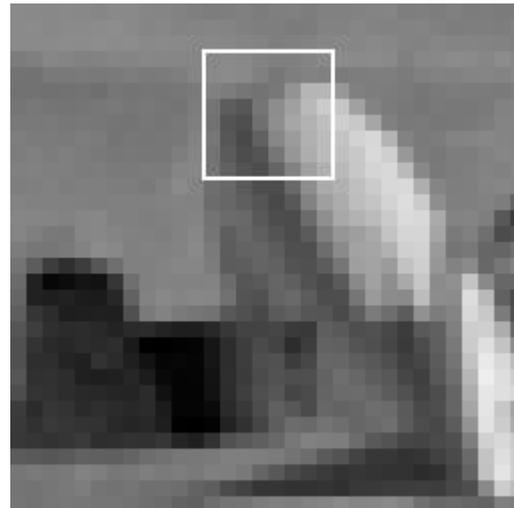


- Images de synthèses
  - CAO
  - Réalité virtuelle
  - Dessin numérique
  - Etc.



# Image bitmap

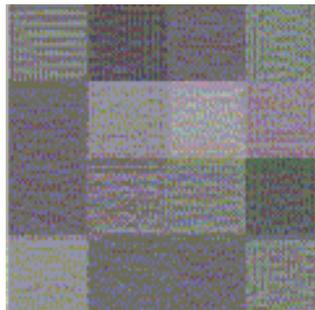
- Une image bitmap est une matrice d'éléments de bases appelés pixels : picture elements
- La taille de la matrice (*ie* le nombre de pixels) définit la **résolution de l'image**
- Chaque pixel possède une valeur représentant une intensité lumineuse ( 0 = noir)
- Le domaine de variation de ces valeurs définit la **quantification de l'image**



137	139	137	128	119	126	130	129
129	124	118	115	120	147	181	182
117	93	87	117	141	160	185	203
115	84	79	111	142	166	178	191
119	87	73	97	135	155	176	187
116	91	77	84	118	150	173	188
117	97	84	78	101	131	160	177
111	104	90	78	78	105	142	170

# Résolution d'une image

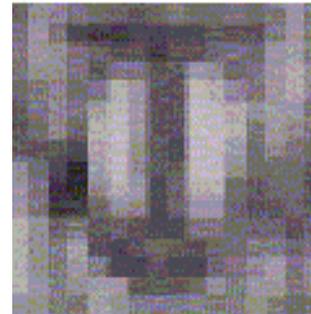
- La résolution d'une image correspond au nombre de pixels qui la composent
- Plus ce nombre est élevé, plus l'image est détaillée



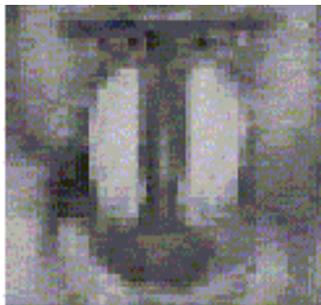
4\*4



8\*8



16\*16



32\*32



64\*64



128\*128

# Quantification d'une image

- La quantification d'une image définit le domaine de variation des valeurs de ses pixels.
- Elle est liée à l'espace mémoire nécessaire pour coder un pixel
- Noir et blanc :
  - 1 bit par pixel
  - $2^1 = 2$  valeurs possibles : {0,1}
- 4 niveaux de gris :
  - 2 bits par pixels
  - $2^2 = 4$  valeurs possibles : [0,3]
- 16 niveaux de gris :
  - 4 bits par pixels
  - $2^4 = 16$  valeurs possibles : [0,15]
- 256 niveaux de gris :
  - 8 bits par pixels
  - $2^8 = 256$  valeurs possibles : [0,255]

# Quantification d'une image



256 NdG



16 NdG



4 NdG



2 NdG

# Les images couleurs

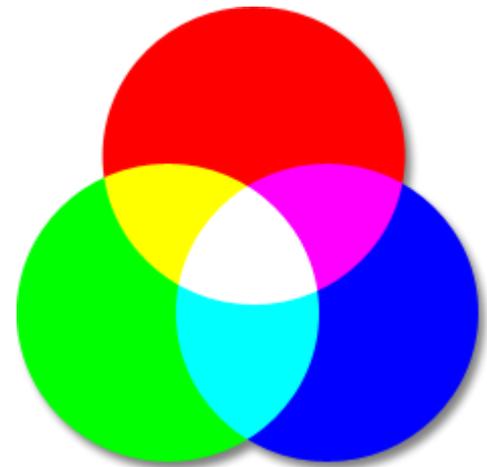
## Le modèle RGB

- Le codage RGB est imposé par les dispositifs d'affichage (écrans)
- En 1859, le physicien écossais James Clerck Maxwell a démontré que toutes les couleurs visibles pouvaient être créées à partir du mélange de trois couleurs primaires projetées sur un fond noir
- Cette découverte est à la base du codage RGB

# Les images couleurs

## Le modèle RGB

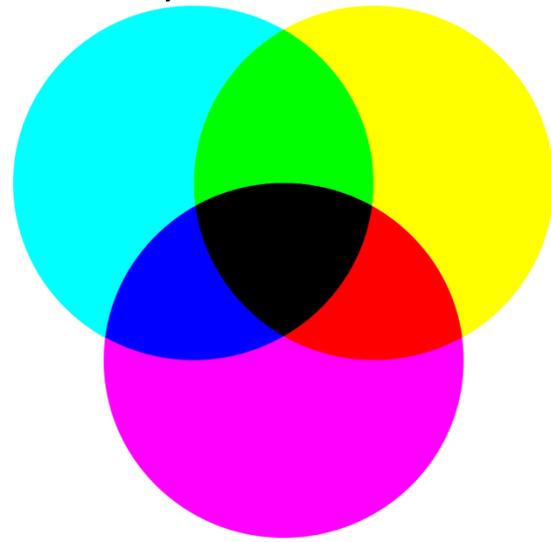
- Principe : on utilise un canal 8bit par couleur primaire :
  - 1 canal Rouge codé sur 8 bits
  - 1 canal Vert codé sur 8 bits
  - 1 canal Bleu codé sur 8 bits
- Tout pixel de l'image est défini par un triplet de valeurs correspondant à la proportion de chaque couleur primaire dans le mélange
- C'est une synthèse additive :
  - $\{0,0,0\}$  = Noir
  - $\{255,255,255\}$  = Blanc



# Les images couleurs

## Le modèle CMY

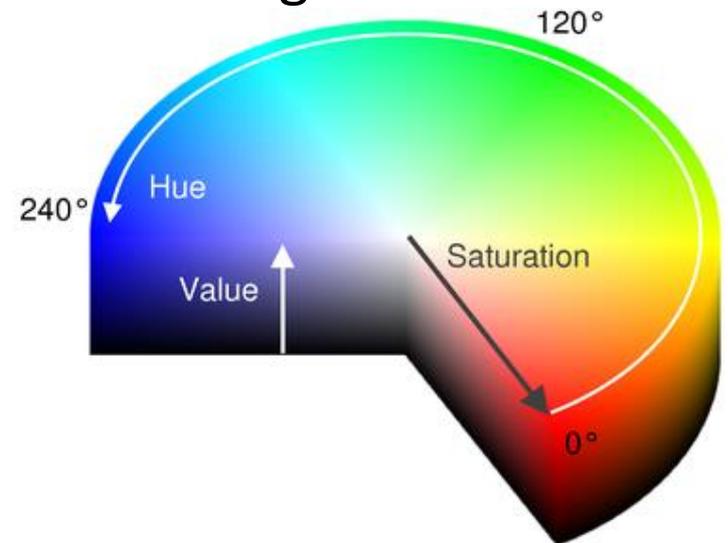
- C'est une synthèse utilisée en impression
- Le codage CMY est basé sur trois couleurs composées :
  - Cyan
  - Magenta
  - Jaune
- Alors que le codage RGB part du Noir et ajoute les couleurs RGB, le codage CMY part du blanc et les soustrait.
- C'est une synthèse soustractive : la couleur perçue est blanche si les composantes n'ont aucune intensité, et noire si leurs intensités sont maximales.



# Les images couleurs

## Le modèle HSV

- C'est une synthèse basée sur la perception humaine
- Utilisée car plus intuitive que RGB
- Trois canaux :
  - Teinte (Hue)
  - Saturation
  - Valeur (Value)
- Teinte : valeur codée sur le cercle des couleurs
- Saturation : distance par rapport au niveau de gris de même couleur
- Valeur : brillance de la couleur



# Les images couleurs

- Autre modèle : HSL (Hue saturation lightness), proche de HSV
- Des fonctions de transfert  $RGB \rightarrow HSV$  et  $HSV \rightarrow RGB$  existent
- Attention : La conversion  $RGB \rightarrow NdG$  n'est pas une simple moyenne des canaux !
- Des études psychovisuelles ont établi que :

$$NdG = \alpha R + \beta G + \gamma B$$

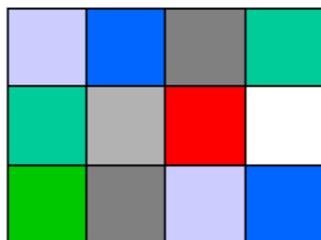
avec  $\alpha + \beta + \gamma = 1, \alpha = 0.2989, \beta = 0.5870, \gamma = 0.1140$

# Les formats d'images

- Composition d'un fichier image
  - En-tête
  - Palette
  - Buffer
- En-tête :
  - Identificateur de format
  - Résolution en x
  - Résolution en y
  - Information optionnelle
  - Nombre de bits/pixel
  - Type de compression s'il y a lieu
  - Nombre d'octets
  - Longueur de l'entête si elle n'est pas fixe

- Principe de la palette :

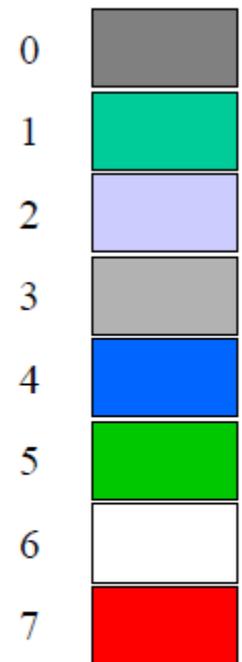
- On a accès à un grand nombre de niveaux
- Seuls certains sont sélectionnés
- La palette fait la correspondance entre le numéro d'indice et la couleur réelle.
- Intérêt : on peut n'utiliser que 8 bits (donc 256 couleurs), mais les choisir parmi  $2^{24}$ .
- Cela permet d'augmenter le contraste
- Ex :



*Image*

2	4	0	1
1	3	7	6
5	0	2	4

*Représentation en mémoire de  
l'image*



*Palette*

# Les formats d'images

## Le buffer

- Les valeurs des pixels sont stockées dans un tableau 1D (ligne après ligne) d'octets
- Si un seul canal (images 8bits) : un octet = un pixel.
- Si plusieurs canaux (ex RGB) : un pixel est représenté par 3 octets successifs, un par canal.

# Les formats d'images

## BMP

- Format d'image bitmap Windows
- Quantification mono canal : 1-, 4-, and 8-bit
- Quantification couleur : RGB 24-bit
- Compressions RLE (facultative)
- Bien supporté sous Windows

# Les formats d'images

## TIFF

- Tagged image file
- Quantification mono canal : 1-, 4-, and 8-bit
- Quantification couleur : 24-bit RGB and 32-bit RGB+alpha
- Compressions Mac Packbits LZW ou aucune
- Multi-plateforme
- Meilleur choix pour l'échange de données

# Les formats d'images

## GIF

- Graphics Interchange Format
- Licence unisys, gratuit si utilisation non commerciale
- Quantification mono canal : 8-bit couleur (palette)
- Compression LZW
- Fomat multi-images
- Multi-plateformes
- 256 couleurs simultanées max

# Les formats d'images

## JPEG

- Joint Photographic Experts Group
- Images compressées
- 8-bit, 24-bit RGB
- Méthode de compression avec perte
- Meilleure compression pour les photographies

# Opérations sur images

- Opérations arithmétiques et logiques sur les images
  - Addition, Soustraction, Multiplication, Division
  - ET logique, OU Logique, Négation

# Addition

- L'opérateur s'applique à 2 images de taille identique et en produit une troisième de même taille, où chaque pixel est la somme des pixels correspondants dans les deux images initiales.

$$I_3(i, j) = I_1(i, j) + I_2(i, j)$$

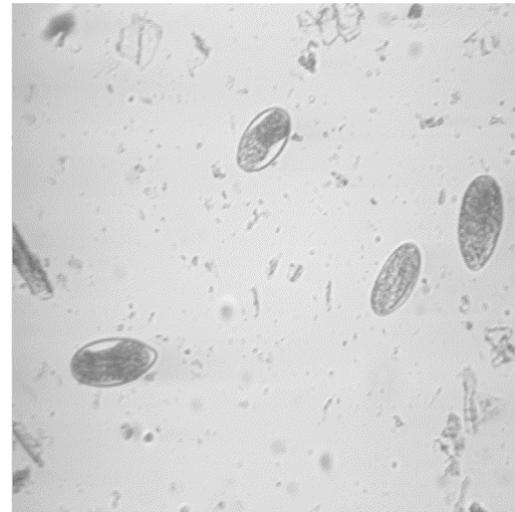
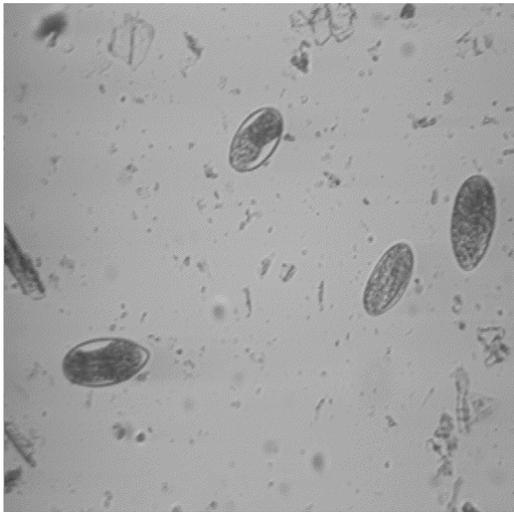
- On peut également effectuer la somme d'une image et d'une constante.

$$I_2(i, j) = I_1(i, j) + C$$

# Addition

- Pour une image couleur, l'addition est réalisée séparément sur chaque composante (rouge, vert, bleu).
- Attention le résultat de l'addition peut dépasser le nombre  $G$  de niveaux d'intensité. Le résultat final dépend de l'implémentation : on peut obtenir la valeur maximale d'intensité (on parle de saturation) ou un modulo peut être effectué avec la valeur maximale.

# Addition



Ajout de la constante 50

# Soustraction

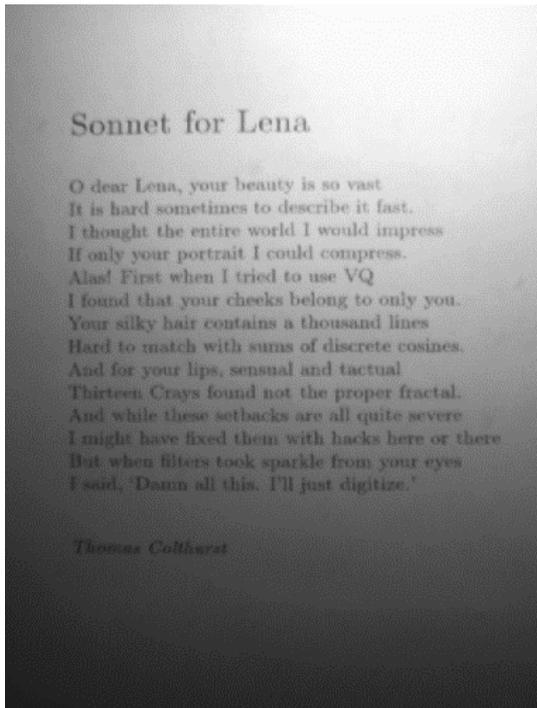
- Soustraction d'une image à une autre ou d'une constante à une image

$$I_3(i, j) = I_1(i, j) - I_2(i, j) \quad \text{ou} \quad I_3(i, j) = |I_1(i, j) - I_2(i, j)|$$

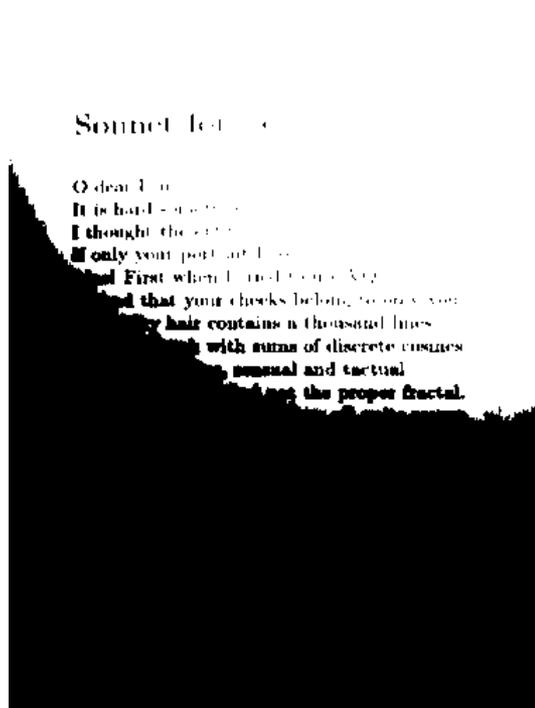
$$I_2(i, j) = I_1(i, j) - C$$

- Mêmes précautions que pour l'addition
- Utilisation classique : soustraction de la luminosité du fond d'une scène pour analyser plus facilement l'avant plan.

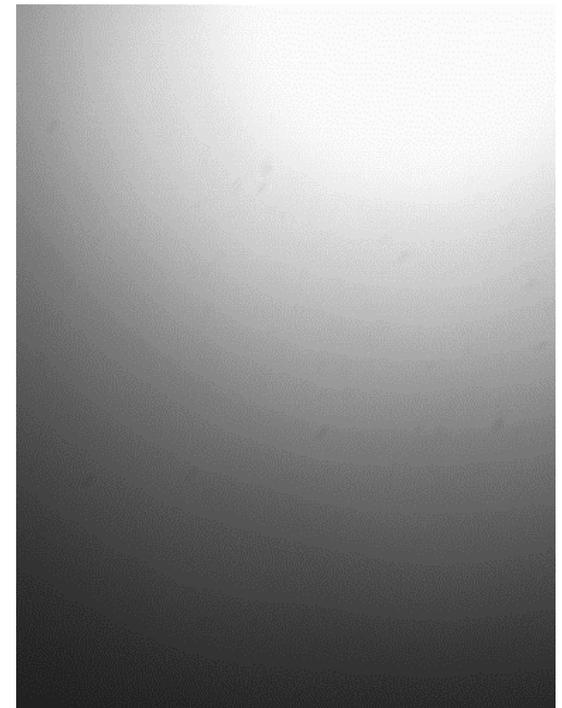
# Soustraction



Problème de luminosité pendant la capture



Seuillage de l'image



Capture d'une page blanche

# Soustraction

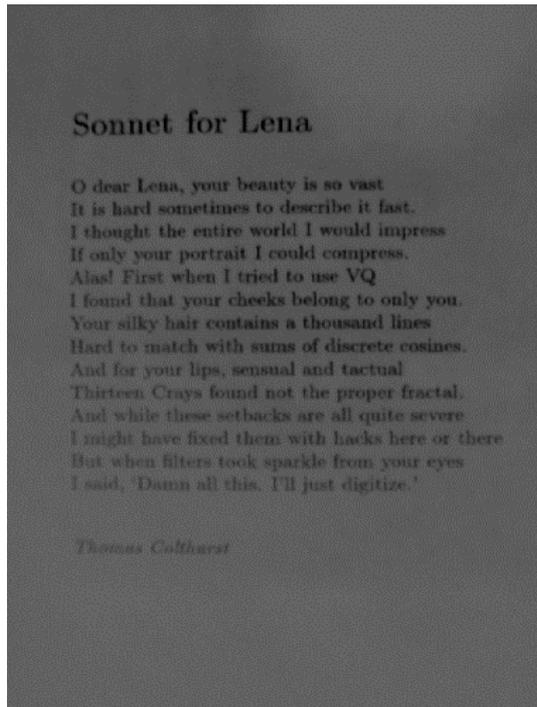
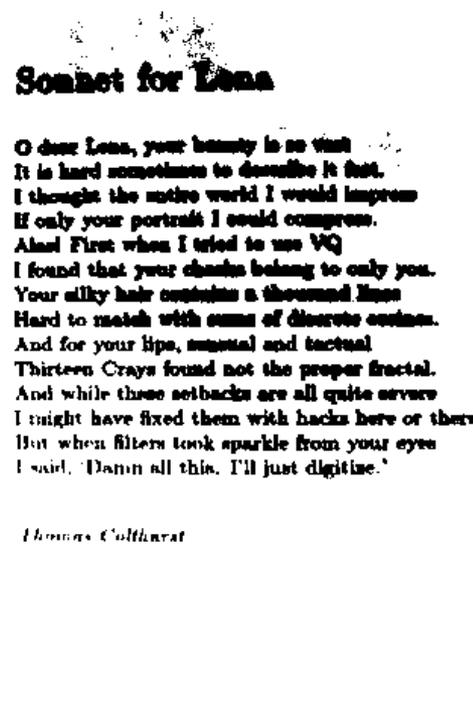


Image d'origine + 100  
– image page blanche



Après seuillage 80

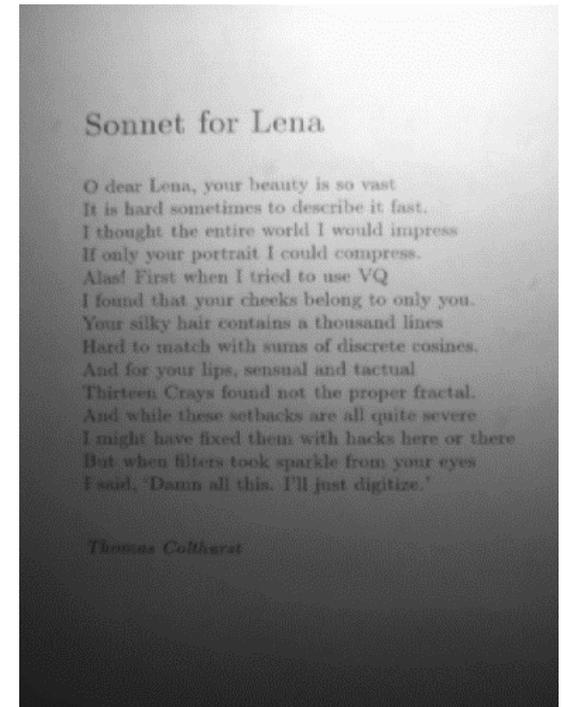
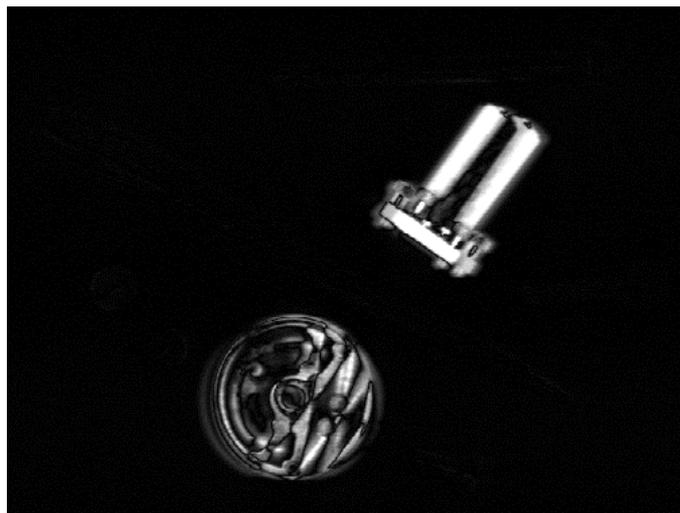
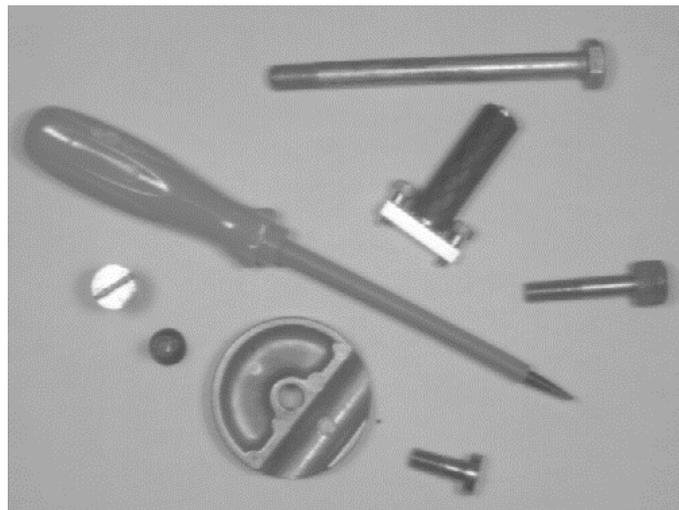
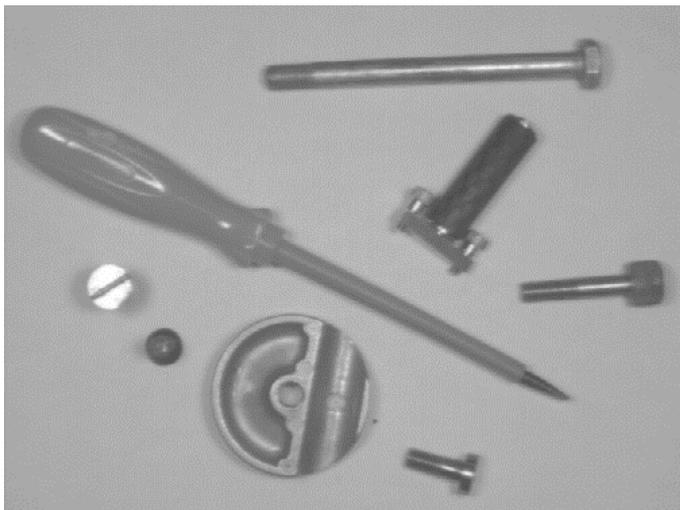


Image d'origine

# Soustraction

- Une autre application est la détection de changements entre deux images d'une séquence vidéo.
- En cas de changement, une différence absolue conduira à des valeurs d'intensité des pixels significativement différentes de 0 dans la région de l'image où le mouvement a eu lieu et à des valeurs nulles ailleurs .

# Soustraction



Différence absolue des deux images

# Multiplication

- Obtenir une troisième image par multiplication de deux images ou par multiplication d'une image par une constante.

$$I_3(i, j) = I_1(i, j) * I_2(i, j) \qquad I_2(i, j) = I_1(i, j) * C$$

- La multiplication par une constante réelle (scaling) est la plus utilisée.
- Avec une constante supérieure à 1, on illumine l'image alors qu'avec une constante inférieure à 1 on l'assombrit.

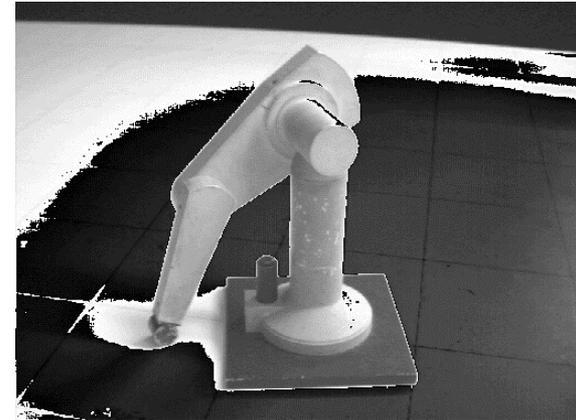
# Multiplication



Image d'origine



Multiplication par 3



Multiplication par 8

Attention au dépassement de capacité

# Multiplication

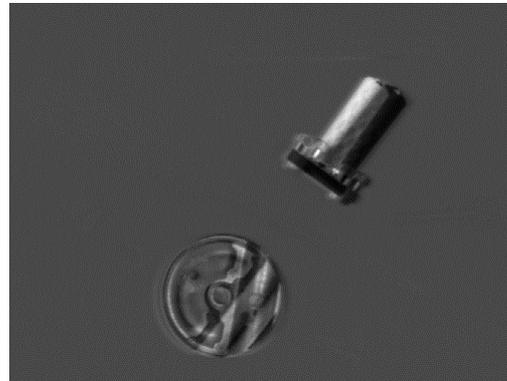
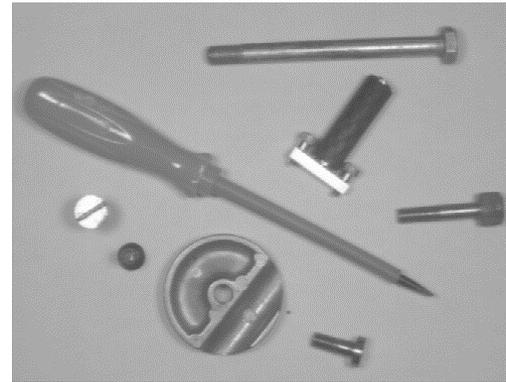
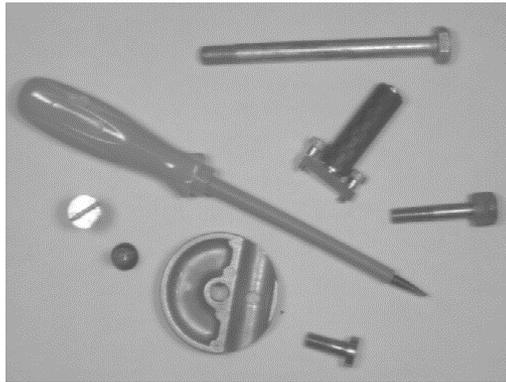
- La multiplication entre pixels est moins utilisée.
- On peut multiplier par une image binaire pour obtenir un masquage. On multiplie par 1 les pixels que l'on veut conserver et par 0 les autres. Il est cependant plus facile et rapide d'utiliser l'opérateur logique ET.

# Division

$$I_3(i, j) = I_1(i, j) / I_2(i, j) \quad I_2(i, j) = I_1(i, j) / C$$

- Le plus souvent, si la division donne un résultat non entier, le résultat est arrondi à l'entier immédiatement inférieur.
- Utilisation pour la détection de changement (comme la soustraction)

# Division



Division de la première image par la deuxième puis amélioration du contraste

# Division

- Suppression de l'influence de l'illumination sur l'image. Comme pour la soustraction, on capture une page blanche et l'on divise l'image d'origine par cette page blanche. Le résultat est gardé en format réel puis étalé entre 0 et 255 et convertit en entiers. Il suffit ensuite d'effectuer un seuillage à 160.
- Attention : il faut d'abord étalé le résultat réel de la division entre 0 et 255 PUIS effectuer l'arrondi.

# Division

## Sonnet for Lena

O dear Lena, your beauty is so vast  
It is hard sometimes to describe it fast.  
I thought the entire world I would impress  
If only your portrait I could compress.  
Alas! First when I tried to use VQ  
I found that your cheeks belong to only you.  
Your silky hair contains a thousand lines  
Hard to match with sums of discrete cosines.  
And for your lips, sensual and tactual  
Thirteen Crays found not the proper fractal.  
And while these setbacks are all quite severe  
I might have fixed them with hacks here or there  
But when filters took sparkle from your eyes  
I said, 'Damn all this. I'll just digitize.'

*Thomas Colthurst*

## Sonnet for Lena

O dear Lena, your beauty is so vast  
It is hard sometimes to describe it fast.  
I thought the entire world I would impress  
If only your portrait I could compress.  
Alas! First when I tried to use VQ  
I found that your cheeks belong to only you.  
Your silky hair contains a thousand lines  
Hard to match with sums of discrete cosines.  
And for your lips, sensual and tactual  
Thirteen Crays found not the proper fractal.  
And while these setbacks are all quite severe  
I might have fixed them with hacks here or there  
But when filters took sparkle from your eyes  
I said, 'Damn all this. I'll just digitize.'

*Thomas Colthurst*

# ET logique

- Deux images de même taille en N&B ou en NdG
- Chaque valeur de pixel de l'image résultat est obtenu par un ET logique sur les pixels correspondant de l'image 1 et de l'image 2.

# ET logique

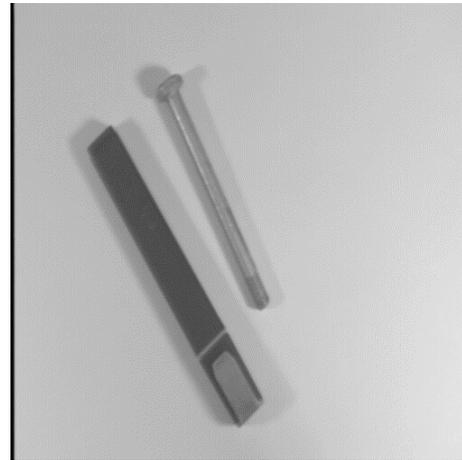
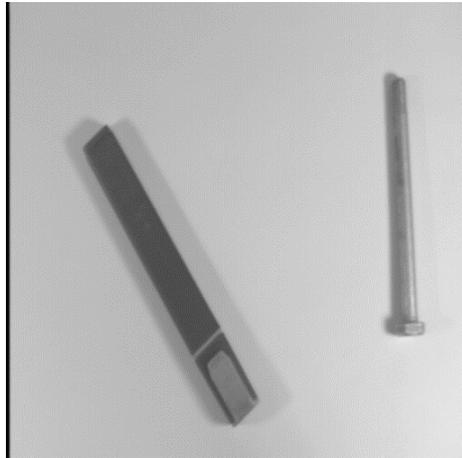
- Pour les images en niveaux de gris, l'opérateur logique est appliqué sur la représentation binaire des niveaux de gris, en comparant les bits correspondants.
- 2 niveaux de gris 47 et 252 codés sur 8 bits,  
47 en binaire 00101111  
252 en binaire 11111100  
Résultat        00101100 soit 44

# ET logique

- Attention aux différentes implémentations des opérateurs logiques
- Certaines considèrent 0 comme faux et toute valeur différente de 0 comme vraie. Le résultat d'un ET sur 2 images NdG est donc une image binaire.

# ET logique

- Application : Intersection de 2 images.



# ET logique

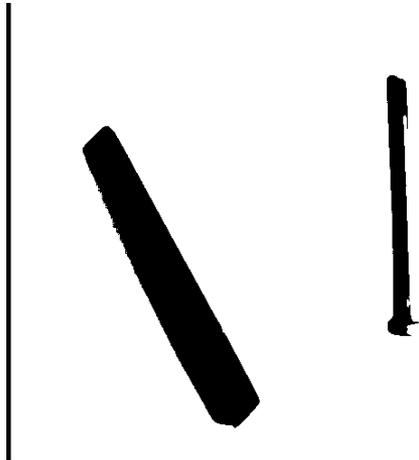


Image N&B par seuillage

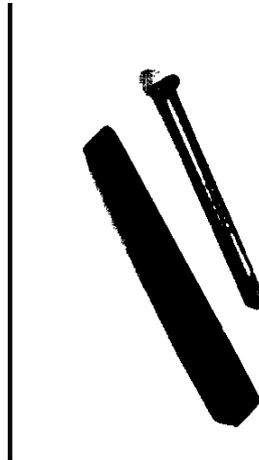
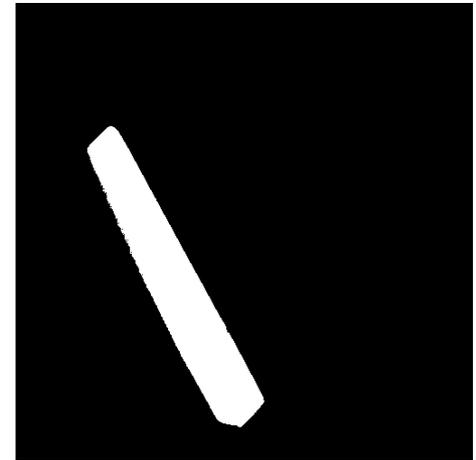


Image N&B par seuillage

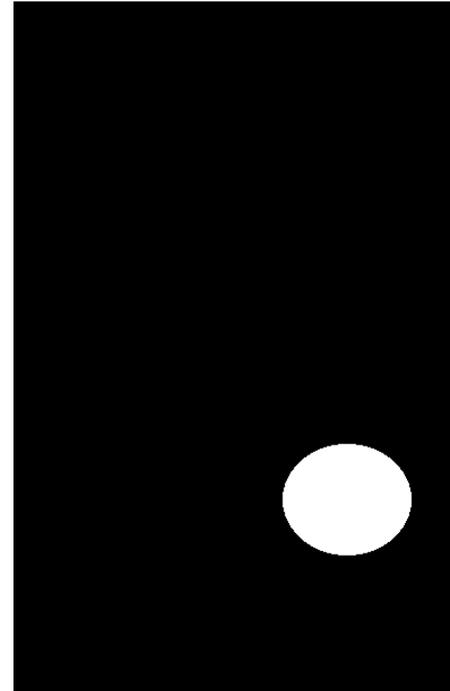


Et logique

Forme des objets immobiles

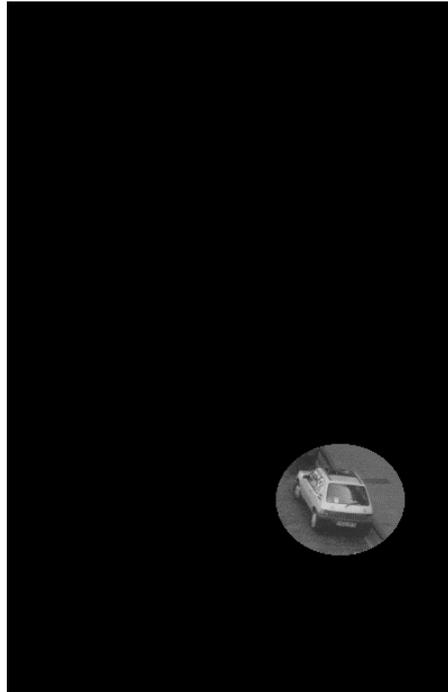
# ET logique

- Application : Masque pour isoler une région



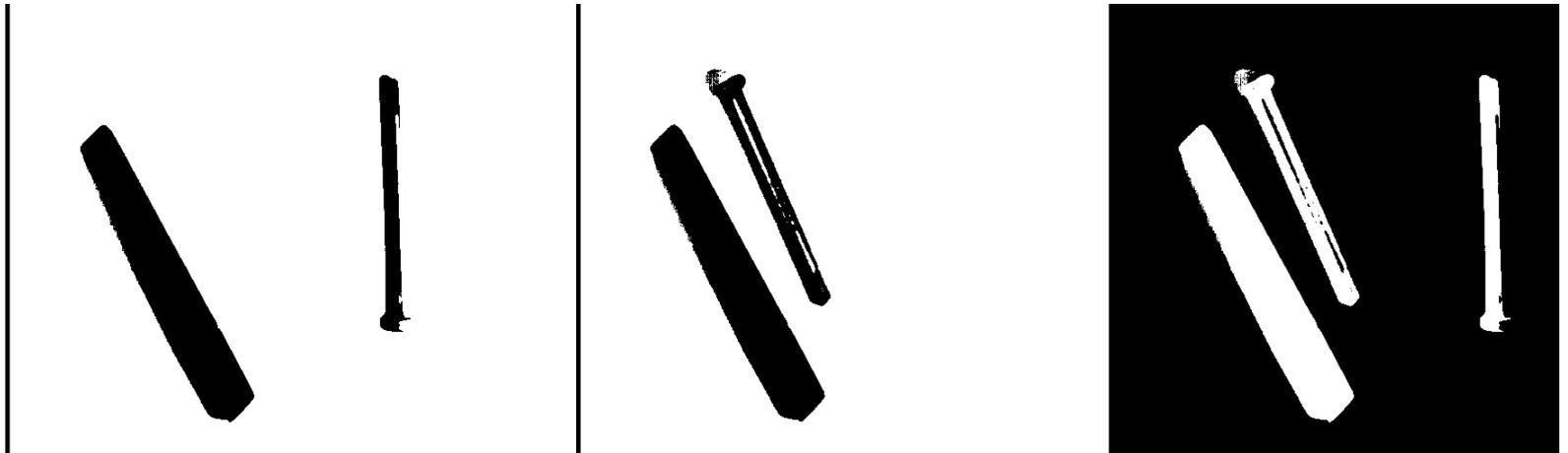
# ET logique

- Application : Masque pour isoler une région



# OU logique

- Application : Fusion de 2 images N&B



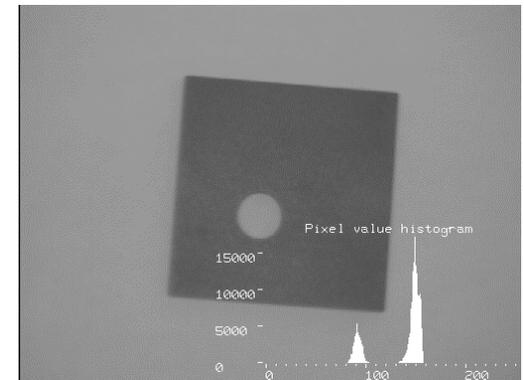
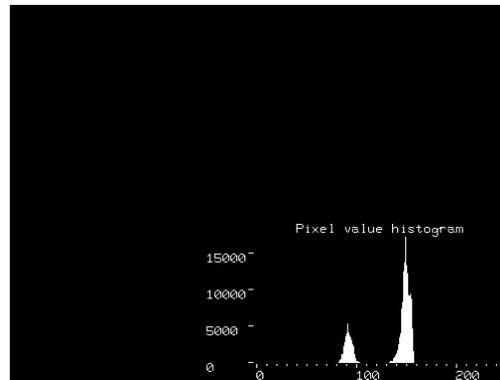
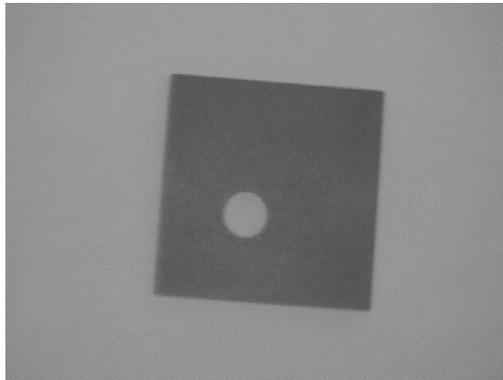
Les objets sont codés 0 et le fond 1. Inverser les images puis faire le ou

# OU logique

- Attention : si les 2 images en NdG, le résultat du OU peut grandement varier avec un changement minime dans les valeurs d'entrée.
- Par exemple 127 OU 128 donne 255 mais 127 OU 126 donne 127.

# OU logique

- Application : Fusion images NdG et N&B



# NON logique

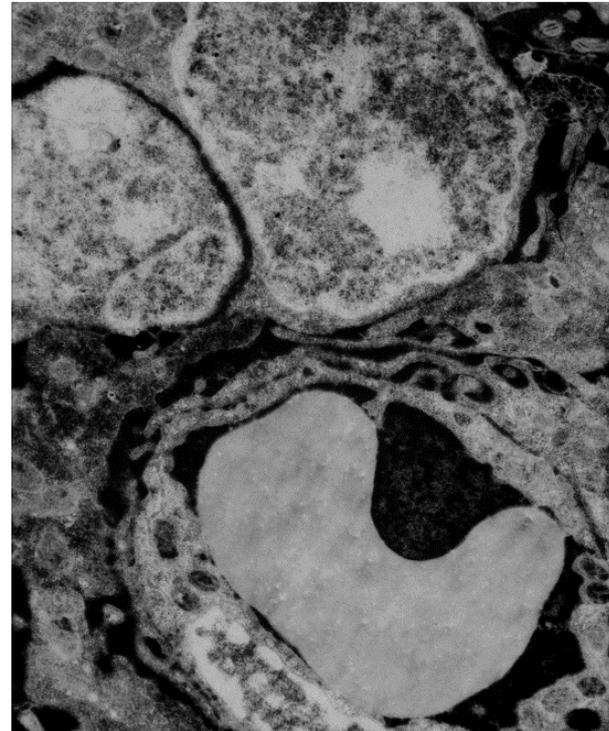
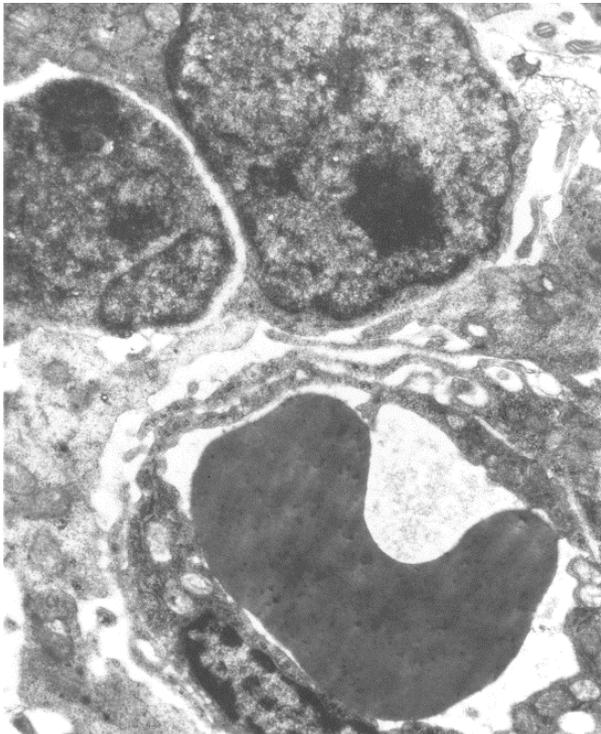
- Aussi appelé opérateur d'inversion.
- Pour une image binaire, change la polarité
- Pour les images en niveaux de gris,

$$I_2(i, j) = 255 - I_1(i, j)$$

- En NdG, la négation fait apparaître plus clairement des caractéristiques.
- Les objets apparaissent en blanc sur fond noir ce qui est plus approprié à la vision humaine.

# NON logique

- Application : Négatif d'une coupe de tissu biologique



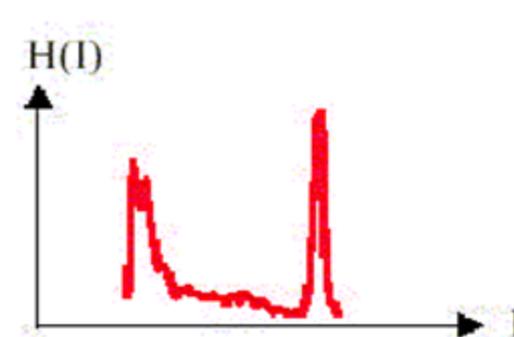
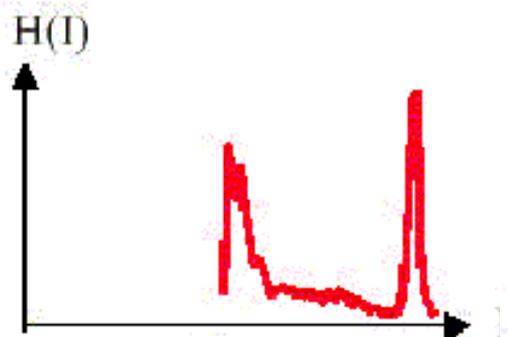
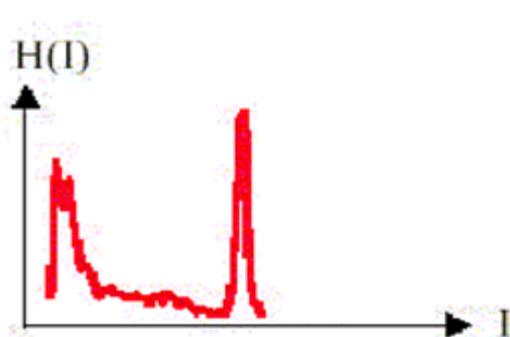
# Traitement par histogramme

- L'histogramme d'une image est l'histogramme des valeurs d'intensité d'une image.
- L'histogramme est un graphique donnant le nombre de pixels trouvés dans l'image pour chaque valeur d'intensité (nombre de pixels en fonction du niveau de gris).

# Histogramme

- Si l'image est en niveaux de gris codés sur 8 bits, il y a 256 valeurs d'intensité possibles. L'histogramme contient alors 256 valeurs  $h_k = n_k$   $k=0, \dots, 255$  avec  $n_k$  nombre de pixels de l'image de niveau de gris  $k$  (on trouve parfois la définition  $h_k = n_k / (N * M)$  ; On parle d'histogramme normalisé).
- Si l'image est en couleur, il y a trois histogrammes, un par composante R, G et B.

# Histogramme



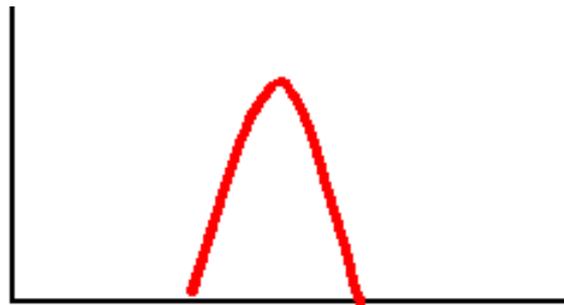
# Histogramme



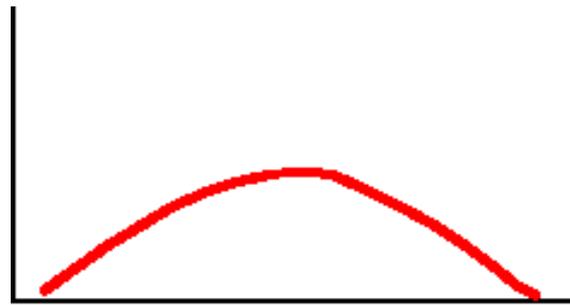
niveaux de gris  
Faible luminance



niveaux de gris  
Forte luminance



niveaux de gris  
Faible contraste



niveaux de gris  
Fort contraste

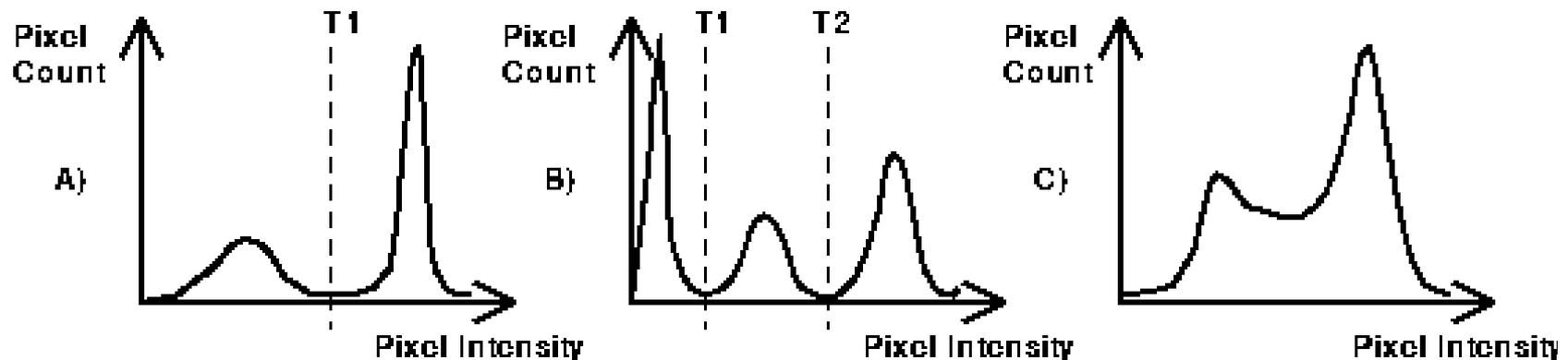
# Seuillage

- Le seuillage permet de séparer les régions d'intérêt de l'image des autres régions (par exemple le fond) en se basant sur les différentes intensités ou couleurs des régions.
- A partir de l'image initiale, on obtient une image noir et blanc.

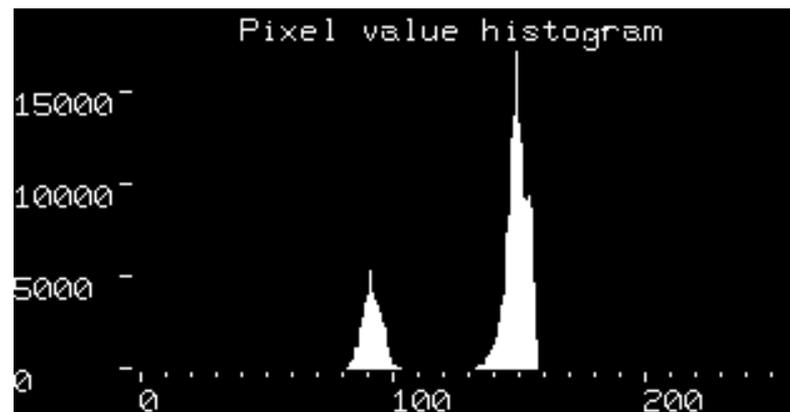
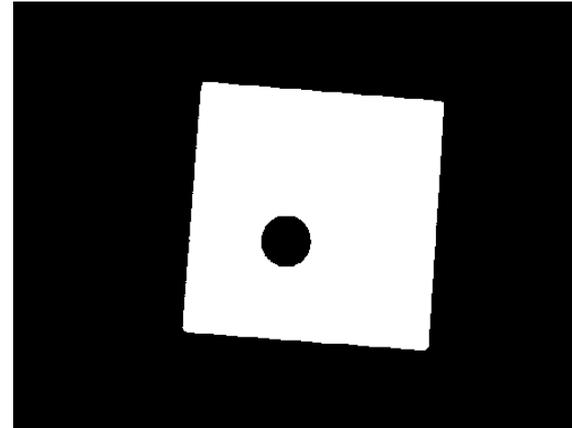
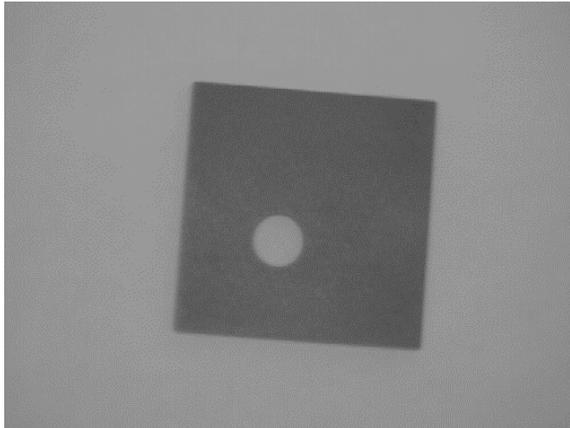
$$I_2(x, y) = \begin{cases} 0 & \text{si } I_1(x, y) < s \\ 255 & \text{si } I_1(x, y) \geq s \end{cases}$$

# Seuillage

- S'il est possible de séparer des parties de l'image à l'aide de l'intensités des pixels, l'histogramme peut laisser apparaître un pic. Le ou les seuils de l'image sont alors choisis pour isoler ce pic.



# Seuillage



# Seuillage

- Seuillage adaptatif : le seuil n'est pas uniforme pour toute l'image
- Découpage de l'image en sous-images et étude de chaque histogramme de chacune des sous-images pour déterminer un seuil propre à chacune.
- Autre solution : s'intéresser au voisinage de chacun des pixels et prendre comme seuil pour ce pixel la moyenne ou la médiane des niveaux de gris du voisinage.

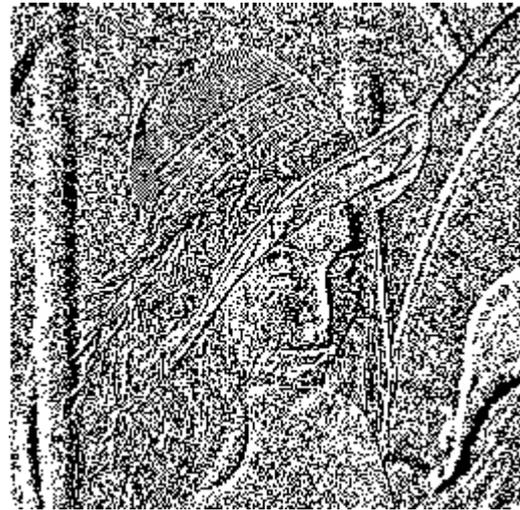
# Seuillage

Méthode efficace lorsque l'histogramme présente des regroupements par type de teinte



# Seuillage

Méthode peu efficace lorsque l'histogramme est uniformément réparti



# Contraste

- Plusieurs définitions du contraste  $C$  (image en niveaux de gris) :

$$C = \sqrt{\frac{1}{N * M} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} (I(x, y) - B)^2}$$

avec  $B$  moyenne de la luminance

$$B = \frac{1}{N * M} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} I(x, y)$$

# Contraste

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

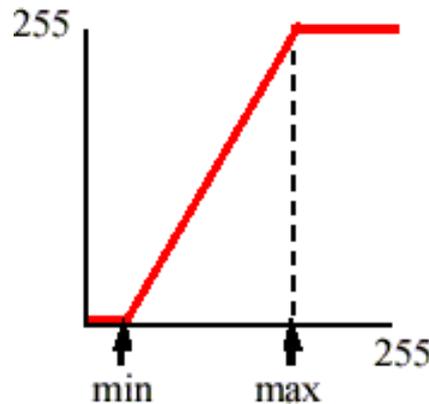
- Par la suite, on suppose traiter des images en 256 niveaux de gris. Quelles sont les méthodes d'amélioration du contraste ?

# Transf. Linéaire

## Egalisation d'histogramme

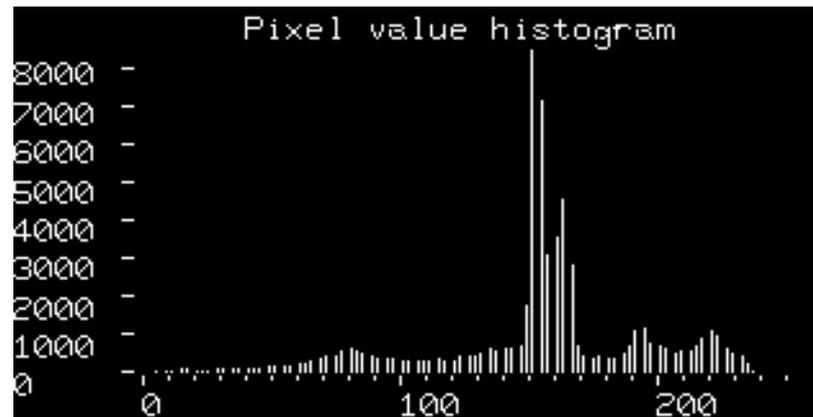
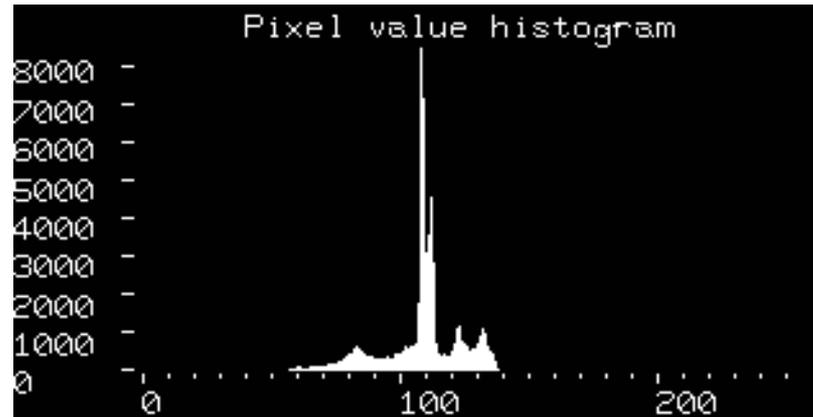
- Principe : étaler les valeurs de niveau de gris pour profiter de toute la largeur de l'histogramme

$$I_2(x, y) = \frac{255}{\max[I_1(x, y)] - \min[I_1(x, y)]} (I_1(x, y) - \min[I_1(x, y)])$$



# Transf. Linéaire

## Egalisation d'histogramme



# Transf. Linéaire

## Egalisation d'histogramme

- Problème de cette transformation : la présence de points parasites avec une très forte ou une très faible valeur modifie  $\max[I_1(x,y)]$  ou  $\min[I_1(x,y)]$  et peut conduire à de mauvais résultats.

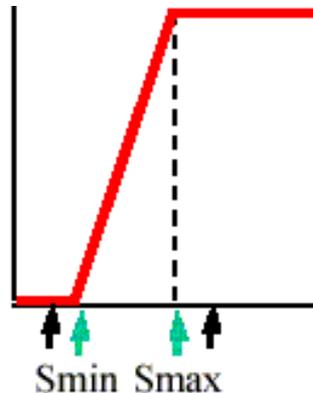
# Egalisation à saturation

- Solution : choisir deux seuils manuellement

$$I_2(x, y) = \frac{255}{s_{\max} - s_{\min}} (I_1(x, y) - s_{\min})$$

si  $I_1(x, y) < s_{\min}$  alors  $I_2(x, y) = 0$

si  $I_1(x, y) > s_{\max}$  alors  $I_2(x, y) = 255$



# Egalisation à saturation

- On choisit  $s_{min}$  et  $s_{max}$  tels que

$$\min[I(x, y)] < s_{min} < s_{max} < \max[I(x, y)]$$

- Une technique de choix consiste à prendre  $s_{min}$  tel que 5% des pixels de l'histogramme aient des valeurs inférieures à  $s_{min}$  et à prendre  $s_{max}$  tel que 5% des pixels aient des valeurs supérieures à  $s_{max}$ .
- L'extension aux images couleurs : mêmes paramètres sur les 3 canaux RGB.

# Transf. non linéaires

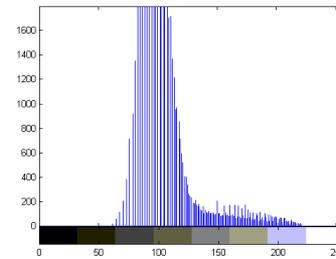
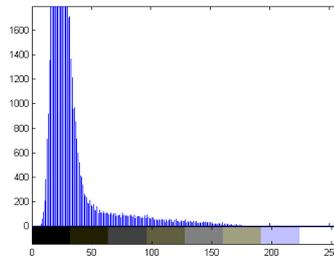
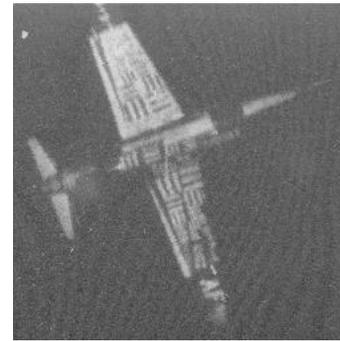
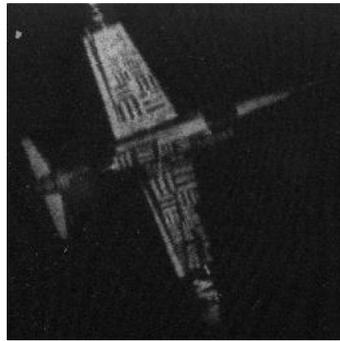
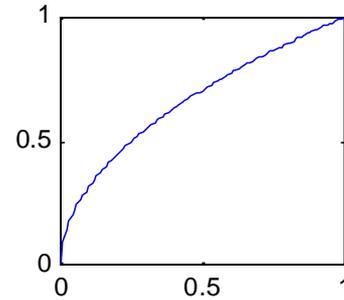
$$I_2(x, y) = f(I_1(x, y))$$

$$f(x) = x^\gamma$$

- $\gamma < 1$  pour améliorer le contraste dans les régions sombres
- $\gamma > 1$  pour améliorer le contraste dans les régions claires

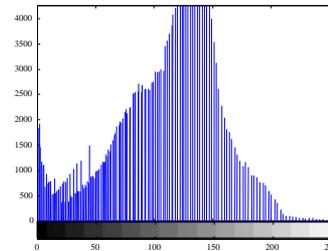
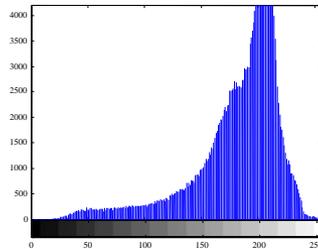
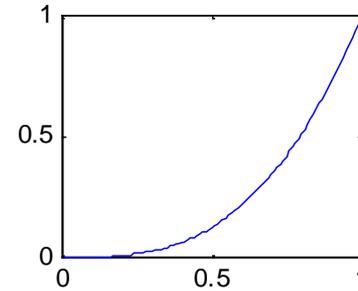
# Transf. non linéaires

Racine carrée  $\gamma = 0,5$



# Transf. non linéaires

Loi carrée  $\gamma = 2$



# Transf. non linéaires

Autres lois usuelles

$$f(x) = k \log(x)$$

$$f(x) = k \log(1 + x)$$

# Transf. non linéaires

## Egalisation automat.

On peut démontrer que pour avoir un histogramme aussi plat "que possible" la fonction correspondante est

$$f(x) = \max\left(0, \text{round}\left[\frac{G * CH(x)}{N * M}\right] - 1\right)$$

où  $G$  est le nombre de niveaux de gris possibles dans l'image et  $CH(x) = \sum_{k=0}^x n_k$

round : arrondi à l'entier le plus proche

# Transf. non linéaires

## Egalisation automat.



# Transf. non linéaires

## Egalisation automat.

Exemple  
G=16  
MxN=262112

$I_1(x,y)$	$N_j$	$CH(j)$	$I_2(x,y)$
0	43	43	
1	5874	5917	
2	13885	19802	
3	11976	31778	
4	8900	40678	
5	7267	47945	
6	9283	57228	
7	13375	70603	
8	28540	99143	
9	66853	165996	
10	56863	222859	
11	21820	244679	
12	8138	252817	
13	4471	257288	
14	4337	261625	
15	487	262112	