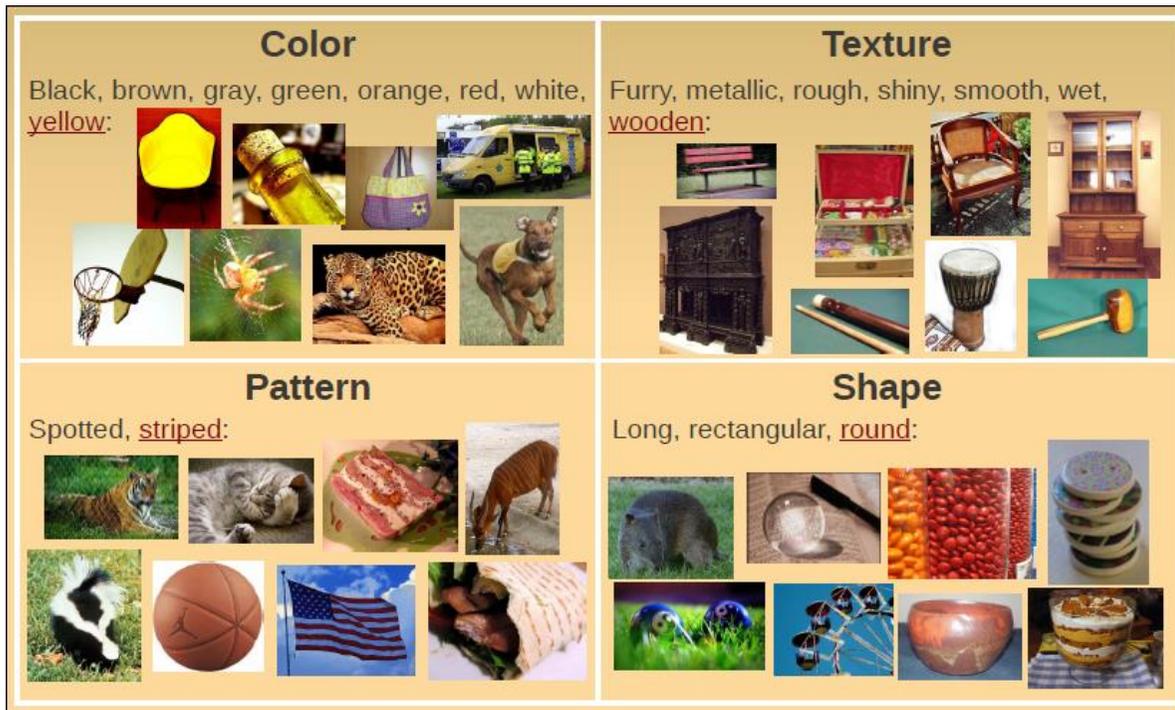
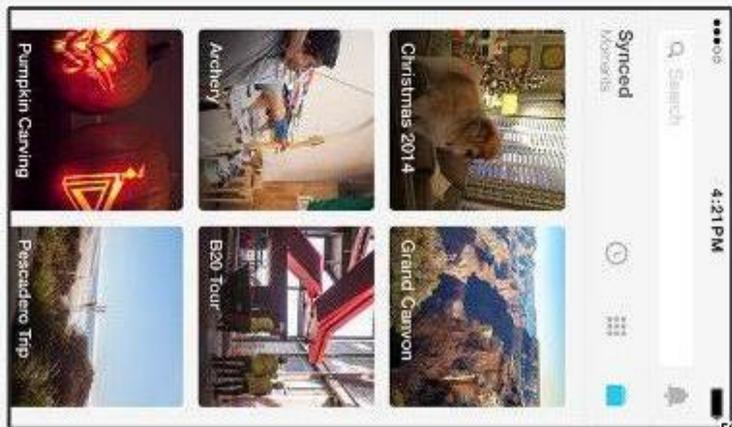


Au programme

- **Alternance entre séances de cours et de TP**
 - CM : 4 x 2h (JYR) + 2 x 2h (VPDA) → **Fil rouge = Image Tagging**
 - Travaux pratiques : 4 x 2h (JYR) → TP classique + seances libres = « Image tagging Contest » + rapport
 - Avec Python et OpenCV (jupyter Notebook ou autre)
 - **A télécharger et installer**
- **Evaluation**
 - CR des Travaux pratiques
- **Contenu**
 - Introduction : Organisation et objectifs (pédagogiques)
 - Types d'images, acquisition, caractérisation
 - Techniques de traitement à base de convolutions
 - Segmentation d'images et deep learning
 - De l'analyse d'images à la reconnaissance de formes (deep learning)
 - Un exemple applicatif : GED (**VPDA, Itesoft – Nime !!!**)



Fil rouge + TP : Image tagging contest...



Adaptations

Color : yellow, green, red, blue, brown, gray, white, black

Shape : spotted, striped

Texture : inside, outside, nature, city, artificial, people, face

Quality : Fuzzy, sharp, dark, light

Complexity: isolated, multiple

Quelques ouvrages

- Diane Lingrand. Introduction au Traitement d'Images, Vuibert, Paris, 2004.
- Alain Trémeau, Christine FernandezMaloigne & Pierre Bonton. Image Numérique Couleur, Editions Dunod, 2004.
- Rafael C. Gonzalez & Richard E. Woods. Digital Image Processing, 2nd edition, PrenticeHall, 2002.
- Wilhelm Burger & Mark J. Burge. Digital Image Processing: An Algorithmic Introduction using Java, Springer, 2007.
- Eric Favier. Cours sur L'analyse et le traitement des images ; Les principes de la vision assistée par ordinateur. ENISE (France).
<http://www.enise.fr/perso/favier/vision/indexvision.htm>
- **Nouvelles techniques de computer vision : Deep Learning → Voir aussi Parcours IA**



Si besoin...

- Si vous avez des difficultés :
 - Internet est vaste : soyez autonomes (et critique) ...
 - Références à des ouvrages / travaux tout au long du cours
- N'hésitez pas à demander, venir discuter :
 - ramel@univ-tours.fr - Bureau 101
- Quelques conseils
 - Ne surtout pas commencer à travailler à la dernière minute
 - Travaillez en plusieurs fois - Se laisser du temps pour réfléchir aux problèmes
 - Du temps est disponible durant les cours et TP - Profitez en !
 - Evitez de trop surfer le web, réfléchissez avant ...

Introduction

Analyse d'images - Computer vision

- **Qu'est ce que c'est ?**
 - connaissances et technologies qui permettent de concevoir des machines qui peuvent « voir »
- **Premier niveau : vision**
 - acquiert une image grâce à une chaîne intégrant des éléments optiques et capteurs
- **Deuxième niveau : traitement de l'image**
 - modifie le contenu de l'image afin de mettre en évidence des éléments d'intérêt (objets, contours)
- **Troisième niveau : analyse, reconnaissance, interprétation**
 - utilise des techniques d'intelligence artificielle pour identifier des formes connues dans l'image

Extension de la perception humaine ?

- **Capteurs meilleurs que l'oeil humain**

- voient ce que nous ne voyons pas directement, du fait de la limitation de notre système visuel
- Autres propriétés optiques : problème d'échelle, de résolution, de point de vue (ex : images satellitaires)
- Autres gammes de longueurs d'onde : caméras multispectrales, infrarouge proche, thermiques (infrarouge lointain)
- Autres cadences : caméras à haute cadence, plusieurs milliers d'images par seconde

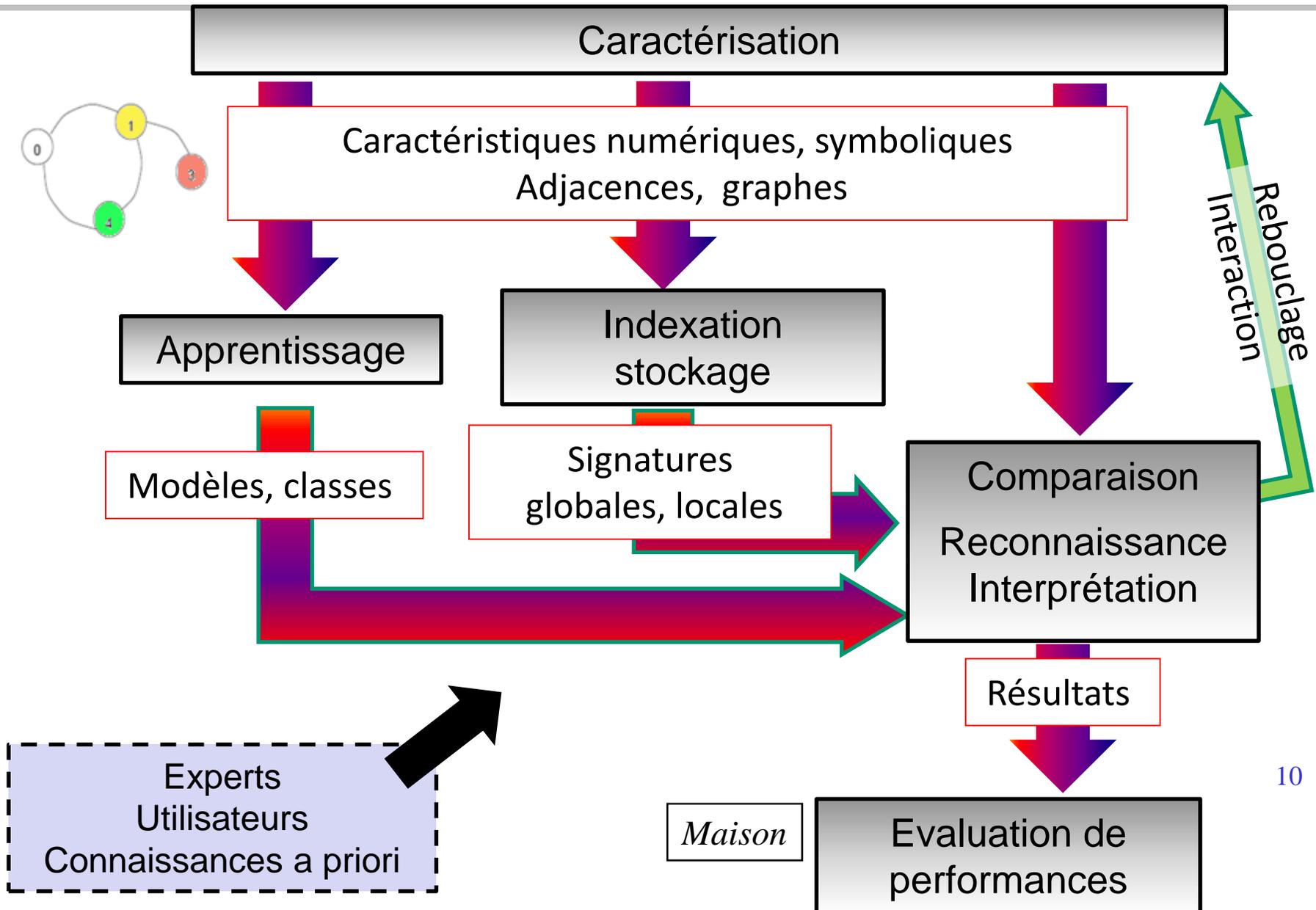


- **Moins « intelligent » que l'humain**

- Le résultat du traitement reste souvent basique
- Puissance de calcul : les systèmes de VA ont une capacité de calcul infiniment plus limitée que celle du cerveau humain
- Connaissance et reconnaissance : nous exploitons nos connaissances pour interpréter le contenu de l'image → on reconnaît parfois des objets alors qu'ils ne sont pas visibles



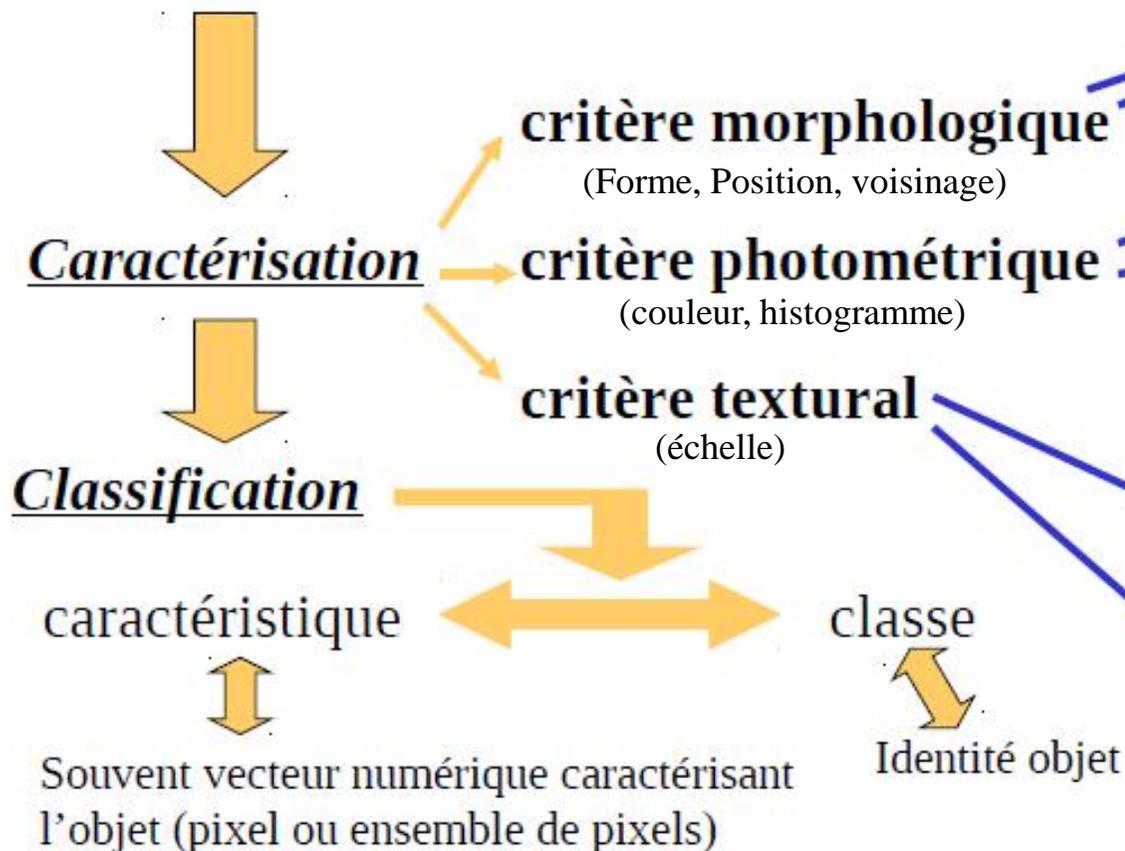
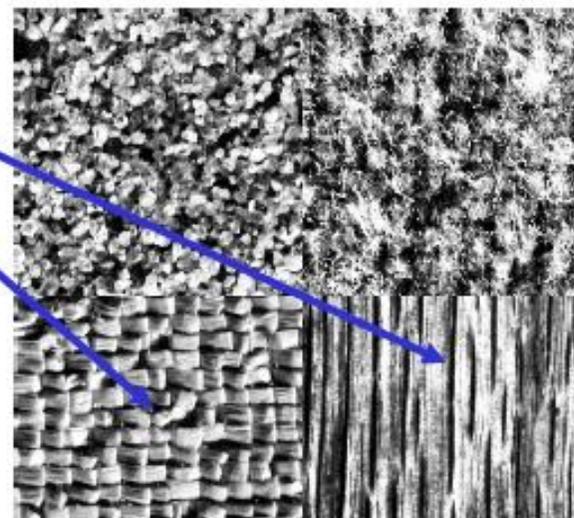
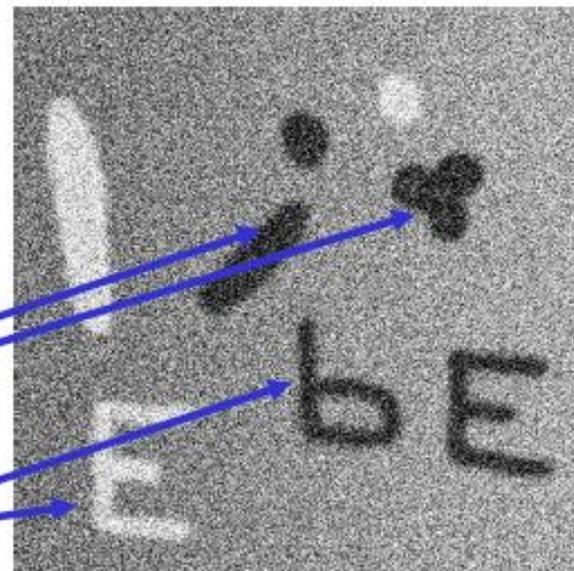
RECONNAISSANCE DES FORMES





Représentation, Caractérisation

Données initiales : prétraitement



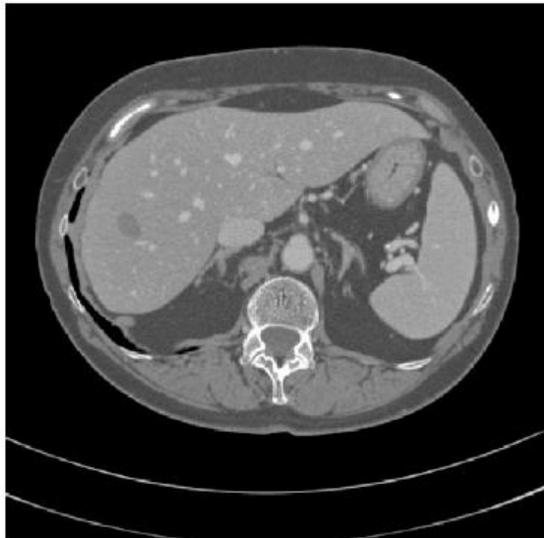
Segmentation vs Reconnaissance

- **Segmentation d'images** = Détection d'objets d'intérêt : localiser (et extraire) des zones caractéristiques = décomposition d'une image en régions homogènes (selon un critère donné).
- **Reconnaissance d'objets d'intérêt** = identifier un objet = lui assigner une classe d'appartenance



Le dilemme : Segmenter pour reconnaître ou reconnaître pour mieux segmenter ???

Voire reconnaître sans segmenter ???



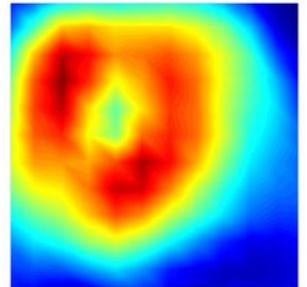
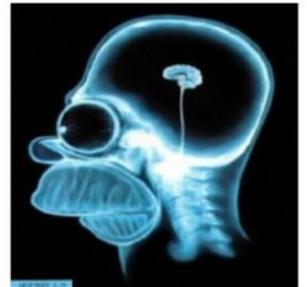
Exemple d'application : Vision industrielle...

- **Objectifs**
 - éviter le contrôle visuel par un opérateur (tâche répétitive peu valorisante)
- **Contrôle dimensionnel**
 - le système de vision détermine la dimension, la forme, la position de l'objet qu'il observe
- **Contrôle d'aspect**
 - le système détermine la couleur, la texture des objets observés
- **Contrôle de la qualité**
 - à partir des données précédentes, le système détermine la qualité d'un produit

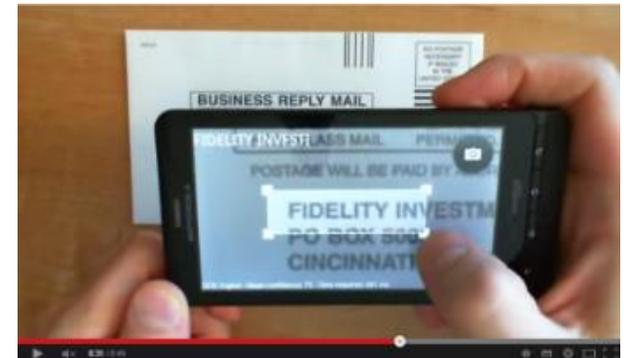
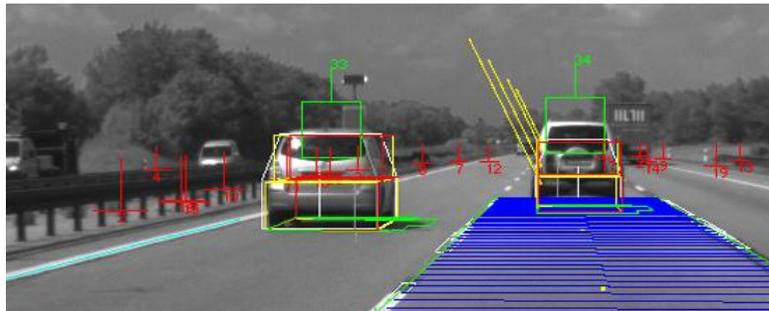
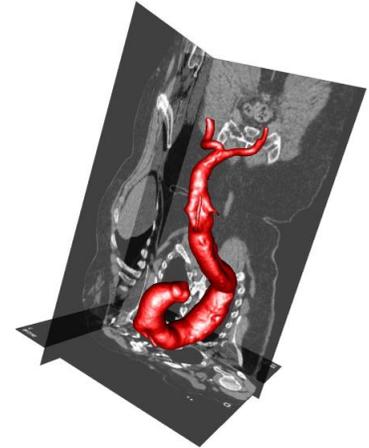
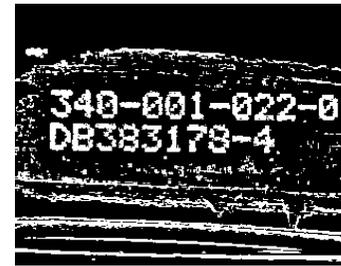
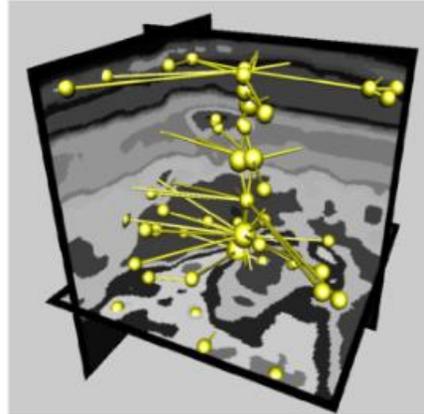
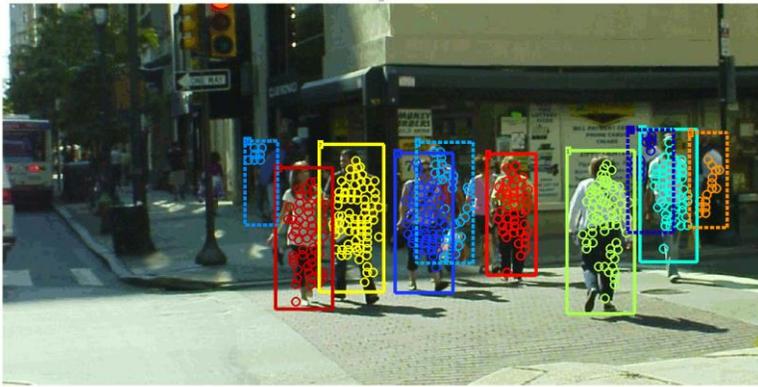
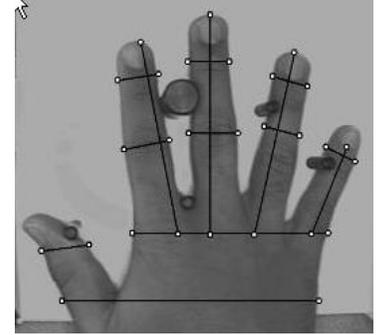
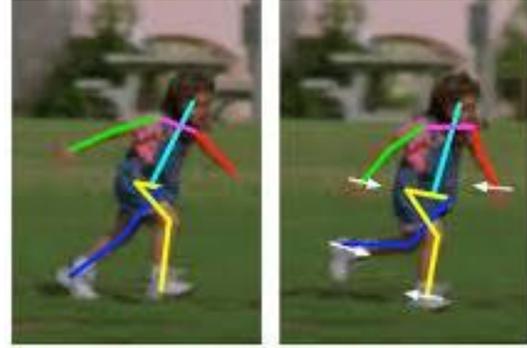
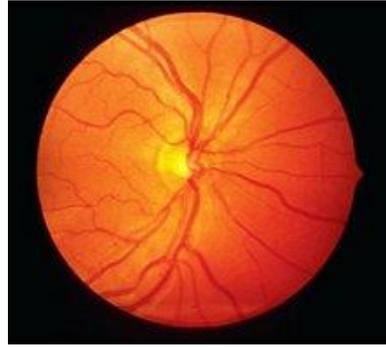
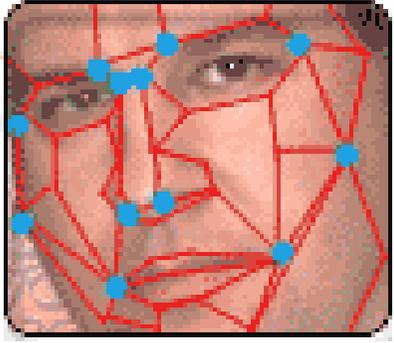


Exemple d'application : Imagerie médicale

- **Objectifs**
 - aider le médecin lors du diagnostic, le chirurgien lors de la réalisation d'un geste opératoire
- **Amélioration des images**
 - rehaussement du contraste, élimination du bruit, mise en évidence des détails
- **Détection et localisation**
 - positionnement des organes, détection des tumeurs, mesure de dimensions et de volumes
- **Imagerie interventionnelle**
 - assistance en ligne au praticien : opérations réalisées sur les images en temps réel



Computer vision is everywhere ...

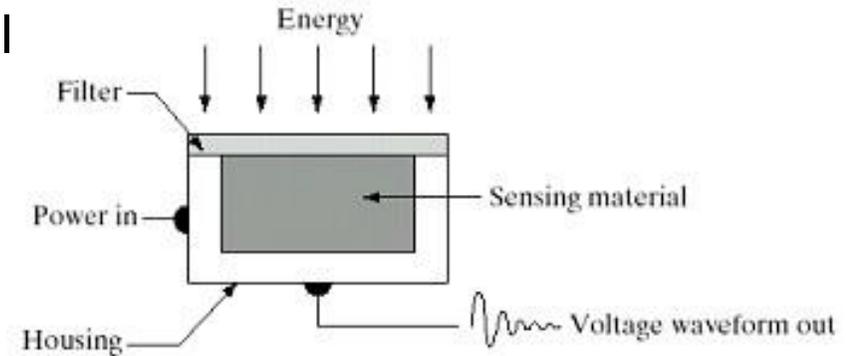


Acquisition, amélioration et mesure de qualité d'images et vidéos

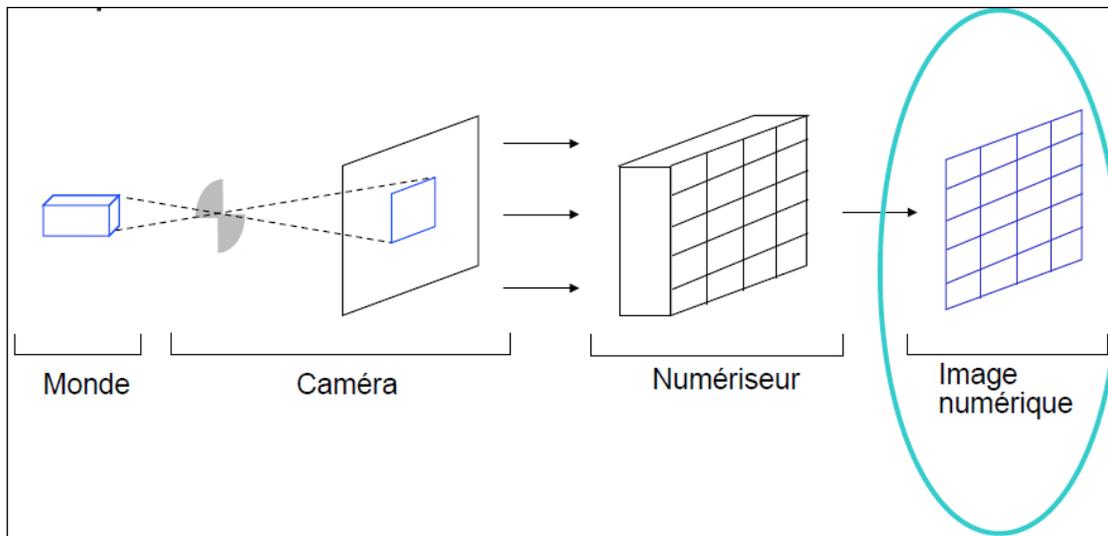
Chapitre 1

Acquisition d'images

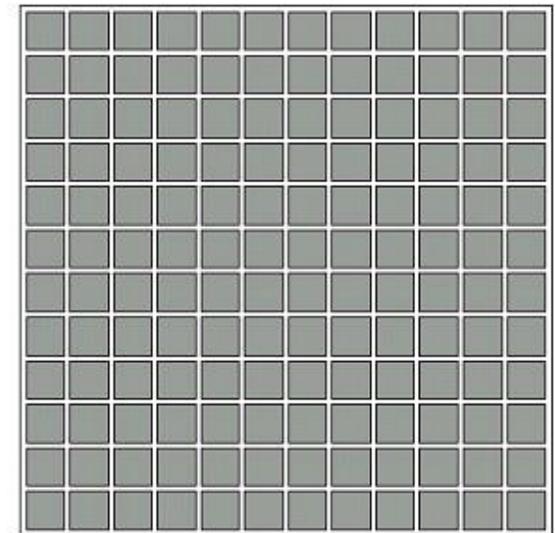
- Capteur : principe général (ex: photodiode)
 - L'énergie incidente est convertie en signal électrique
 - Sortie est proportionnelle à la lumière
 - Filtre pour augmenter la sélectivité



- Schéma de principe



Matrice 2D



Acquisition d'images

- Caméras numériques – Capteur CCD
 - Matrice CCD (Charged Coupled Devices)
 - Système d'acquisition numérique 2D le plus utilisé
 - La réponse est proportionnelle à l'intégrale de l'énergie lumineuse qui atteint chaque élément
 - Pour la couleur, on utilise trois capteurs par pixel réagissant à des longueurs d'ondes différentes (rouge, vert et bleu)
- Images numérique
 - Les valeurs de $f(x,y)$ sont la réponse du capteur au phénomène observé
 - Les valeurs de $f(x,y)$ sont des valeurs de « voltage » continu
 - Les valeurs de $f(x,y)$ doivent être converties vers le domaine numérique
 - *Conversion Analogique/Numérique (A/N)*
 - Deux procédés sont impliqués pour numériser une image

Numérisation = Échantillonnage + Quantification

Acquisition d'images

Echantillonnage et quantification

- L'**échantillonnage** est limité par la capacité du capteur, donc le nombre de pixels disponible (ou autre limite imposée)
- La **quantification** est limitée par la quantité de tons (de gris) définie dans l'intervalle

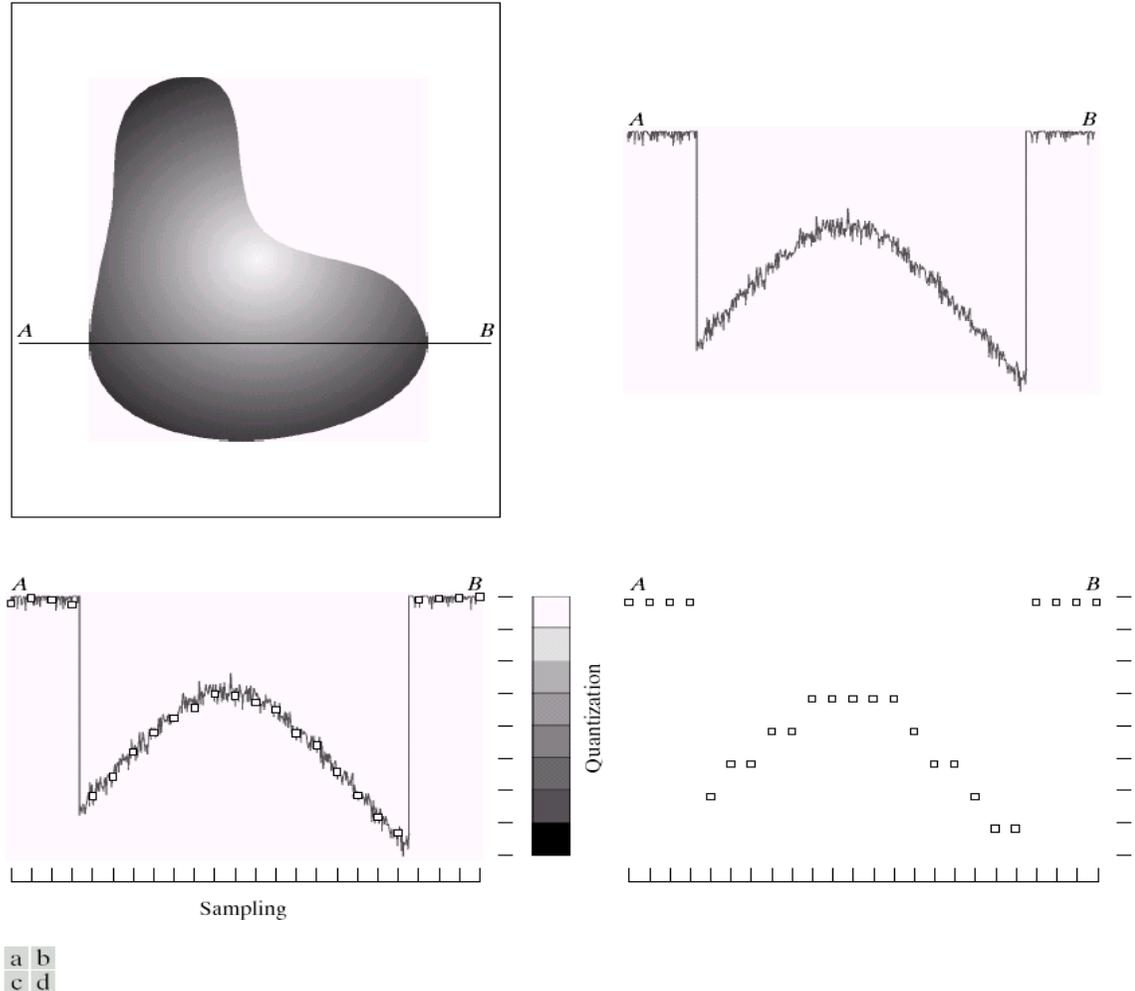
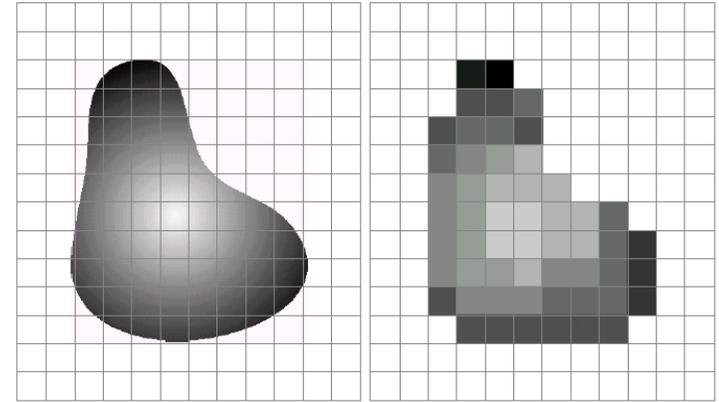


FIGURE 2.16 Generating a digital image. (a) Continuous image. (b) A scan line from *A* to *B* in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.

Acquisition d'images

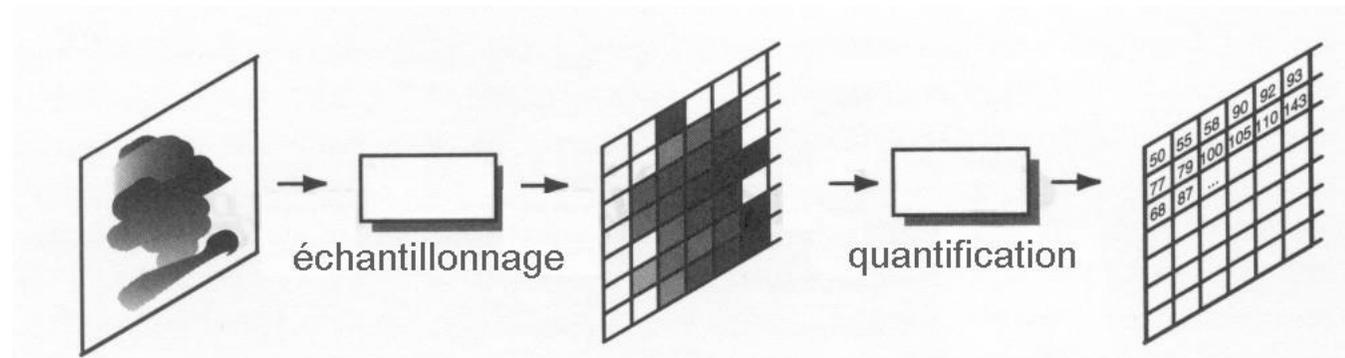
Echantillonnage et quantification

- **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 $M \times N$ 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) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.



Acquisition d'images

Echantillonnage et quantification

- Résolution spatiale



256x256



128x128



64x64



32x32

- Résolution tonale



6 bits



4 bits



3 bits



2 bits



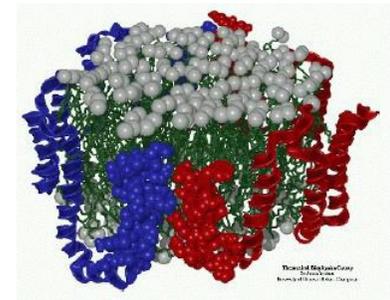
1 bit

Format de fichiers

- Nombreux formats de fichiers pour sauvegarder les images
 - TIF, GIF, JPEG, PNG, PPM, PGM, BMP, ...
 - Chaque format a ses particularités
 - Entête contenant les informations de l'image
 - Pixels de l'image codés de différentes façons
- **A retenir**
 - Formats sans compression
 - Format avec compression
 - Compression sans perte : Fichier compressé sans modification de la valeur des pixels
 - Exemple : PNG
 - Compression avec perte : Valeurs des pixels modifiées pour prendre encore moins de place après la compression
 - Exemple : JPEG
 - **JPEG** est très bien pour visualiser les images (vision humaine) mais très mauvais pour le traitement d'images, car on risque de perdre beaucoup d'informations
 - ➔ Eviter autant que possible de sauvegarder les résultats en JPEG

Qu'est ce qu'une image ?

- Une image est avant tout un **signal 2D** (x,y)
 - Souvent, cette image représente une **réalité 3D** (x,y,z)
- D'un point de vue **mathématique** :
 - Une image est (un ensemble de) matrice(s) de nombres représentant un signal
 - Plusieurs outils permettent de manipuler ce signal
- **Image naturelle – Image artificielle**
 - Plusieurs moyens d'acquisition : caméra, microscope, tomographie, infra-rouge, satellite, ...
 - Plusieurs outils de représentation, synthèse d'images, réalité virtuelle, visualisation scientifique, ...
 - De plus en plus souvent **multi-modales, multi-spectrales, vidéos, 3D+T, ...**
- D'un point de vue **humain** :
 - Une image contient plusieurs informations sémantiques
 - Il faut interpréter le contenu au-delà de la valeur des nombres

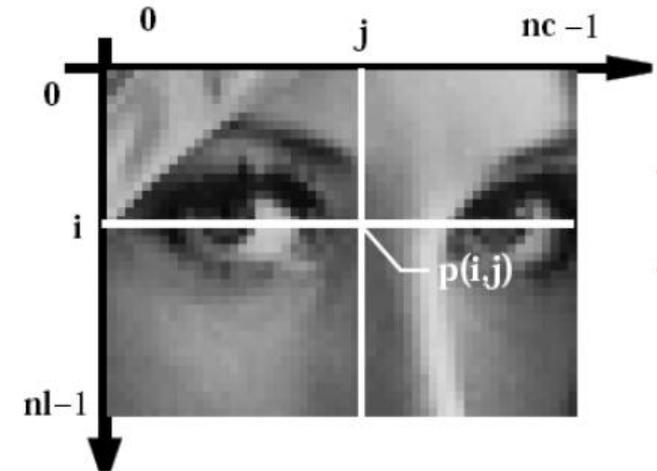
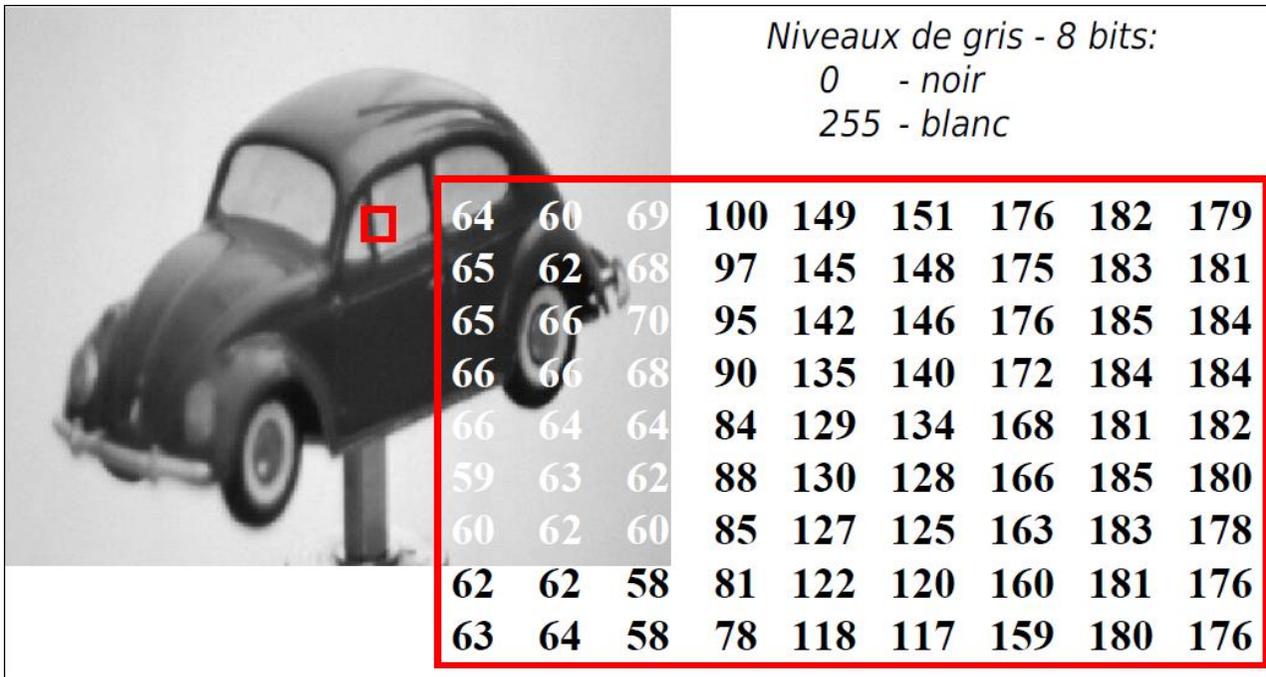


Qu'est ce qu'une image ?

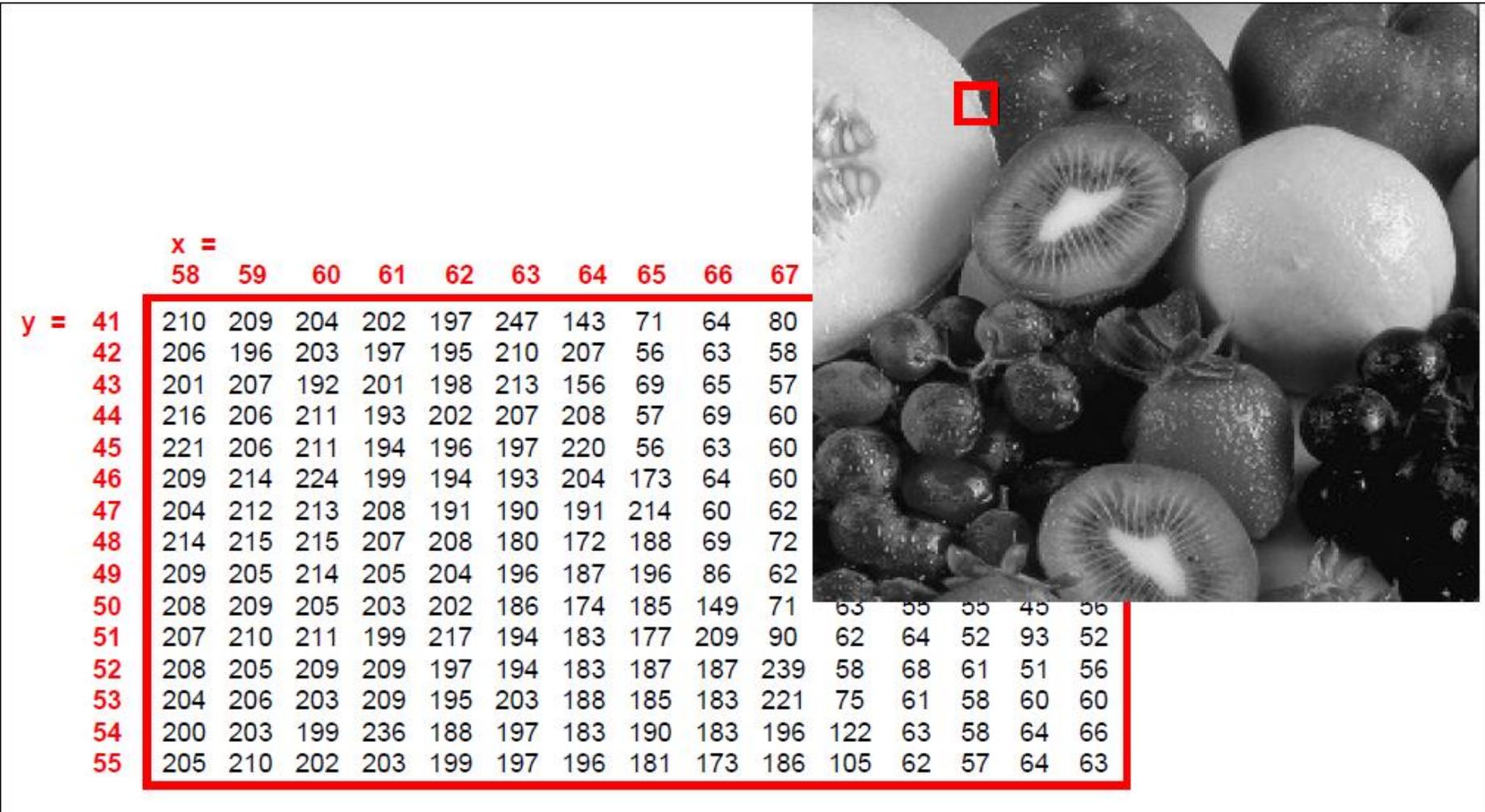
Une image numérique est un tableau de pixels.

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 ?



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

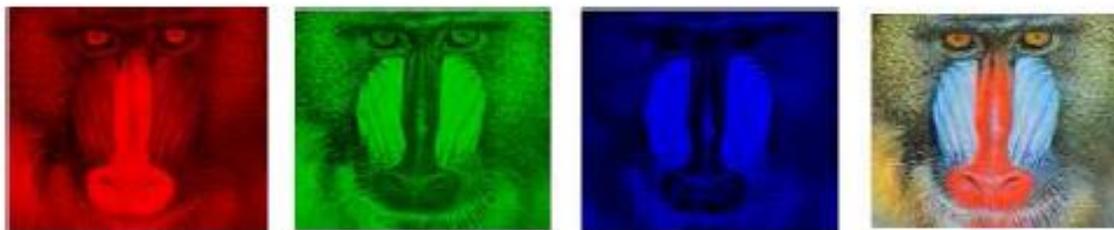
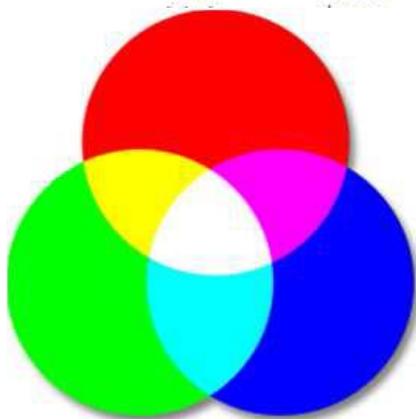
binaire : $I(i, j) = 0$ noir ou $I(i, j) = 1$ blanc → **Images binaires**

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

En couleur

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

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

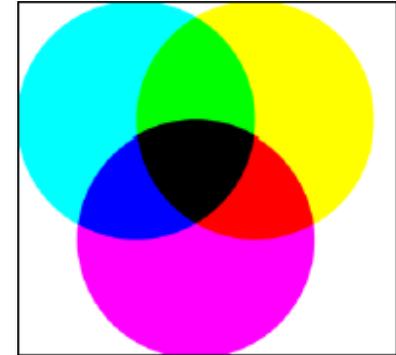


Modèle additif

Qu'est ce qu'une image ?

- Le **codage CMY** basé aussi sur 3 couleurs

- Cyan
- Magenta
- Jaune

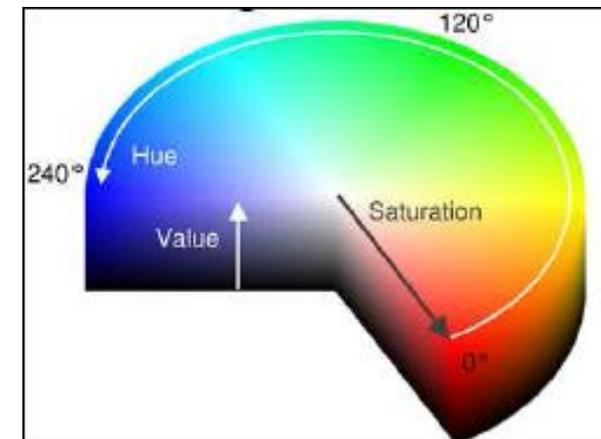


Alors que le codage RGB part du Noir et ajoute les couleurs RGB, le codage CMY part du blanc et les soustrait

C'est une synthèse soustractive : la couleur perçue est blanche si les composantes n'ont aucune intensité, et noire si leurs intensités sont maximales

- Le **codage HSV** basé sur la perception humaine

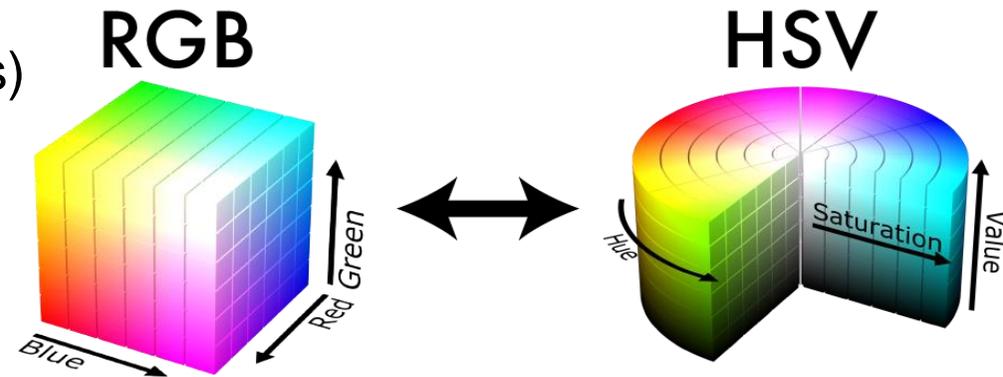
- Teinte (Hue): valeur sur le cercle des couleurs
- Saturation : distance par rapport au niveau de gris de même couleur
- Valeur : brillance de la couleur



Qu'est ce qu'une image ?

Il existe de nombreuses façons de représenter la couleur (espaces couleurs)

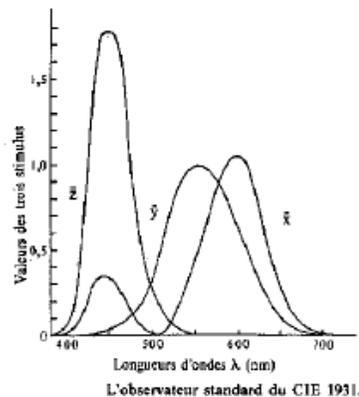
- RVB
- HSL
- YUV
- ...



Formules de transformation :

Diagramme de chromaticité du CIE (1935)

- couleurs primaires :
 - couleurs théoriques X, Y et Z
 - Y luminance
 - mélange donne blanc
- élimine les valeurs négatives
- modèle CIEXYZ



$$\begin{aligned} X &= 0.489989 & R + 0.310008 & G + 0.2 & B \\ Y &= 0.26533 & R + 0.81249 & G + 0.01 & B \\ Z &= 0.0 & R + 0.01 & G + 0.99 & B \end{aligned}$$

$$\begin{aligned} R &= 2.3647 & X - 0.89658 & Y - 0.468083 & Z \\ G &= -0.515155 & X + 1.426409 & Y - 0.088746 & Z \\ B &= -0.005203 & X - 0.014407 & Y + 1.0092 & Z \end{aligned}$$

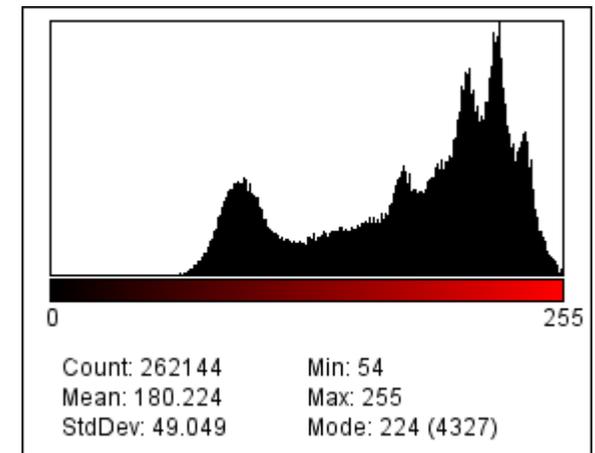
Le TP en Fil rouge....



Comment détecter des couleurs dans des images ?

Difficultés ?

- Sélection de pixels → énumérations → histogramme simplifié
- Quantification → Clustering

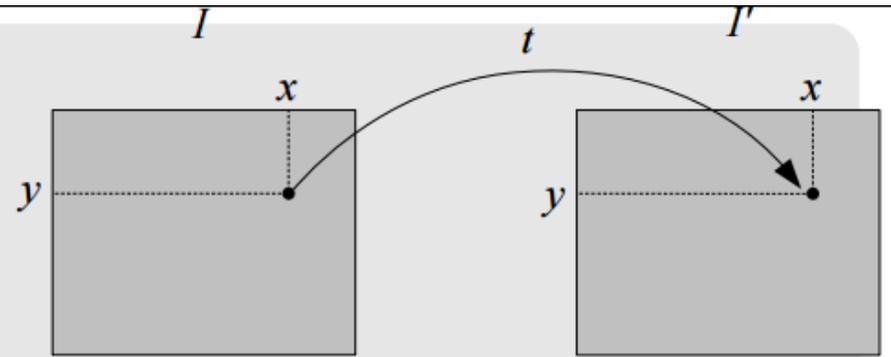


Type de traitements d'images

- Transformations ponctuelles**

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

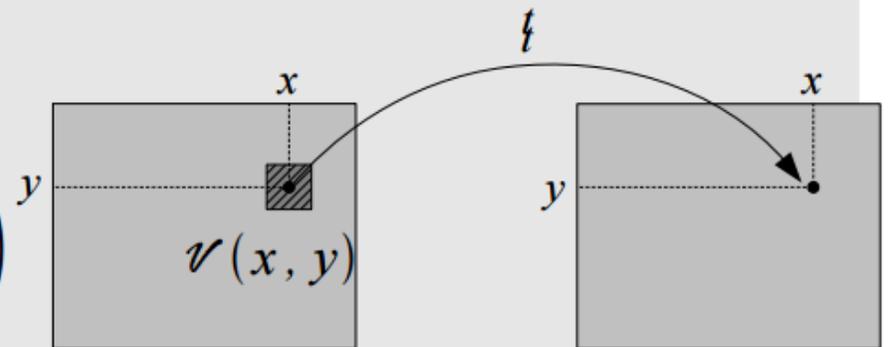
Ex. seuillage, ajustement luminosité/contraste
opérations algébriques, manip. d'histogramme



- Transformations locales**

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

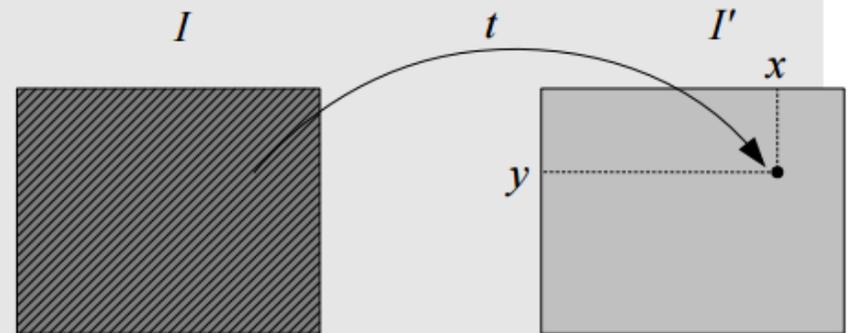
Ex. filtrage



- Transformations globales**

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

Ex. transformation dans l'espace de Fourier



Changement d'échelle (interpolation)

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

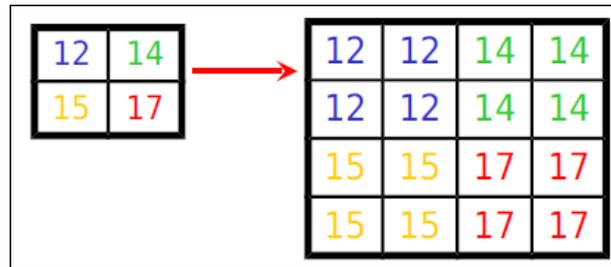
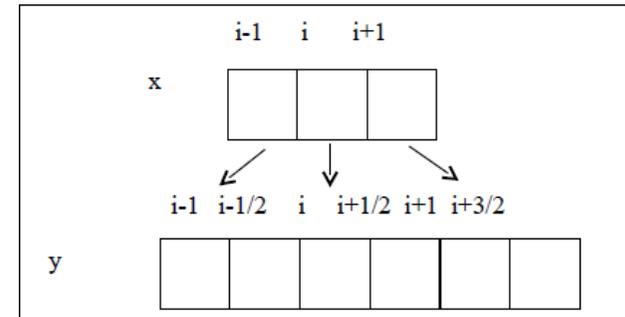


FIGURE 2.19 A 1024×1024 , 8-bit image subsampled down to size 32×32 pixels. The number of allowable gray levels was kept at 256.

Changement d'échelle (interpolation)

• 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*
- *extrapolation linéaire de* : $2 \cdot x_N - x_{N-1}$



• Interpolation bicubique

- Même raisonnement que précédemment
- $y_{i+1/2} = (-x_{i-1} + 9x_i + 9x_{i+1} - x_{i+2}) / 16$
- Problèmes pour $y_{1/2}$, ainsi que pour les 2 derniers points →

$$\begin{aligned} x_{-1} &= 3x_0 - 3x_1 + x_2 \\ x_N &= 3x_{N-1} - 3x_{N-2} + x_{N-3} \\ x_{N+1} &= 2x_N - x_{N-1} \end{aligned}$$

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

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

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

Changement d'échelle (interpolation)

- *Plus proches voisins*



- *Bilinéaire (4 voisins)*



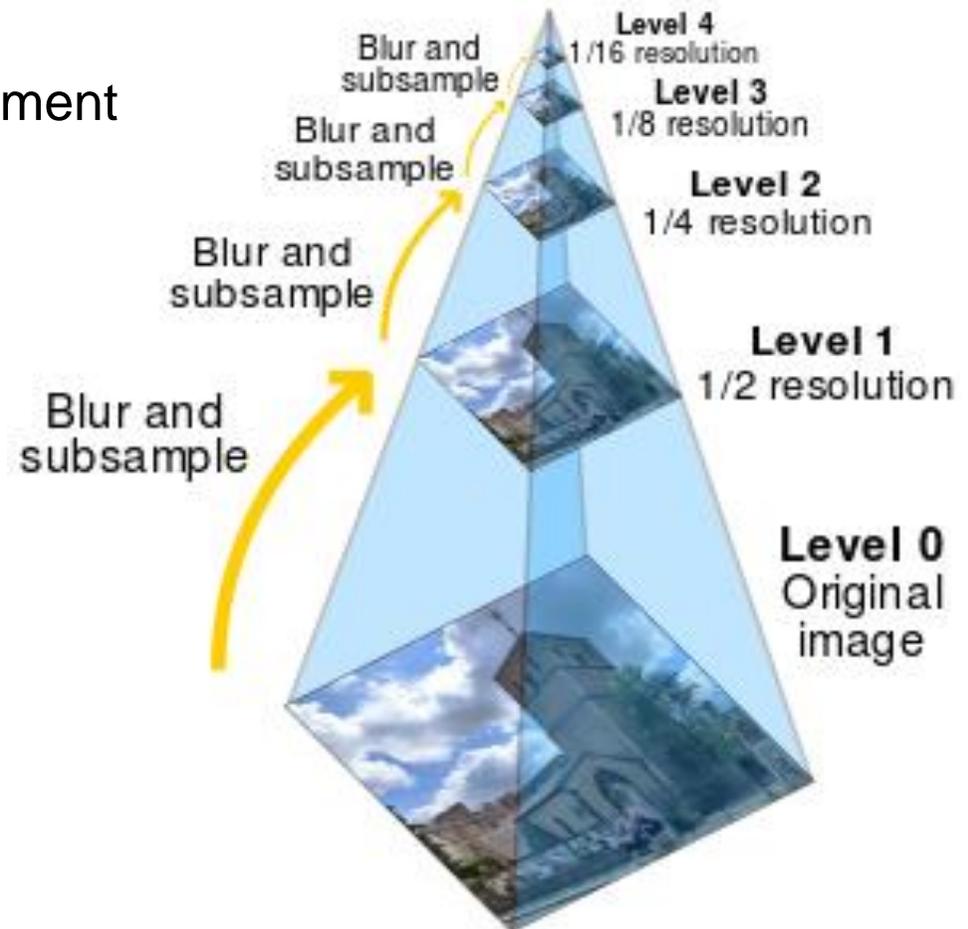
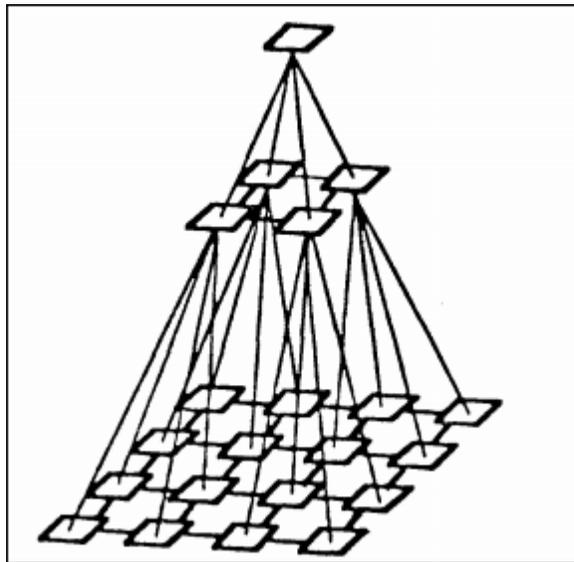
- *Bicubique (16 voisins)*



Le TP en Fil rouge....

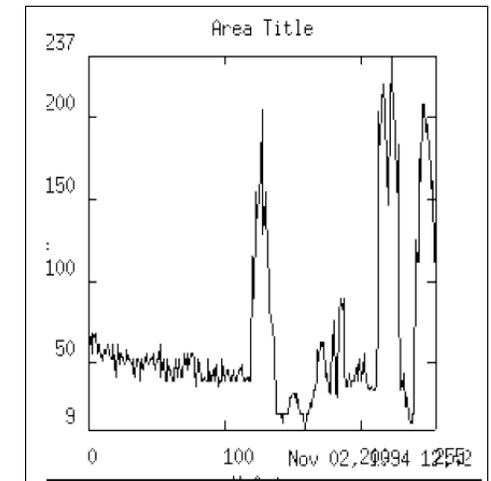
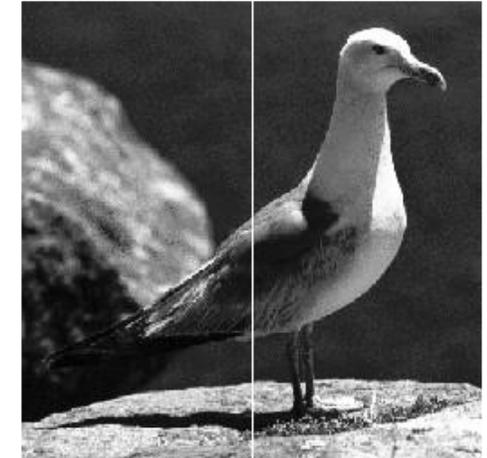
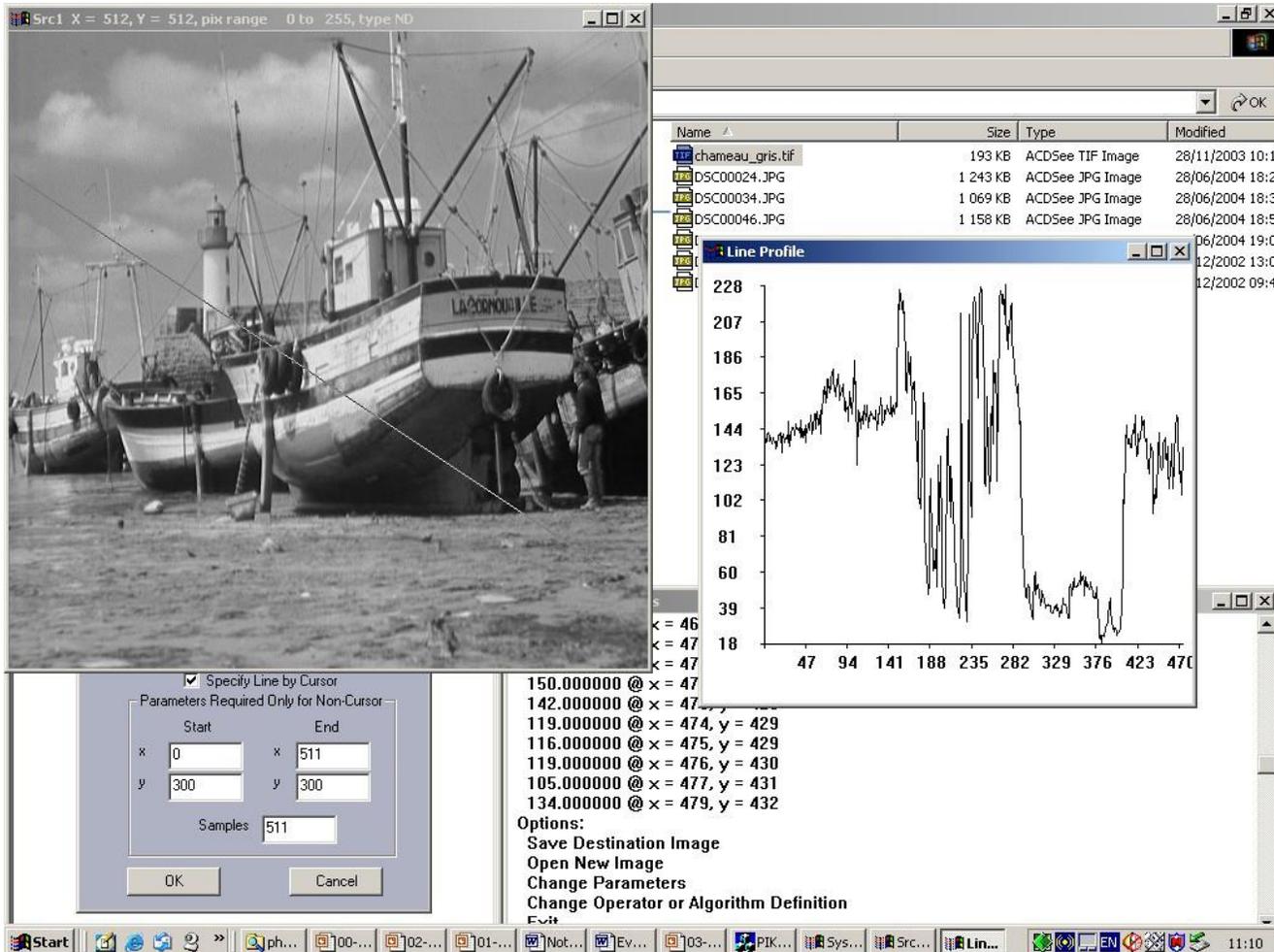
Choix de l'échelle d'analyse ? Analyse pyramidale ?

- Sous-échantillonnage → analyse / traitement



Amélioration de qualité (Niveaux de gris)

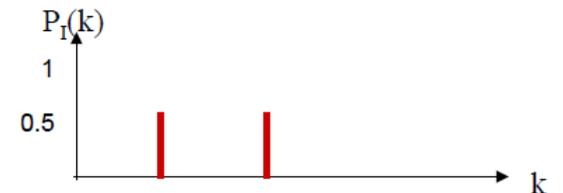
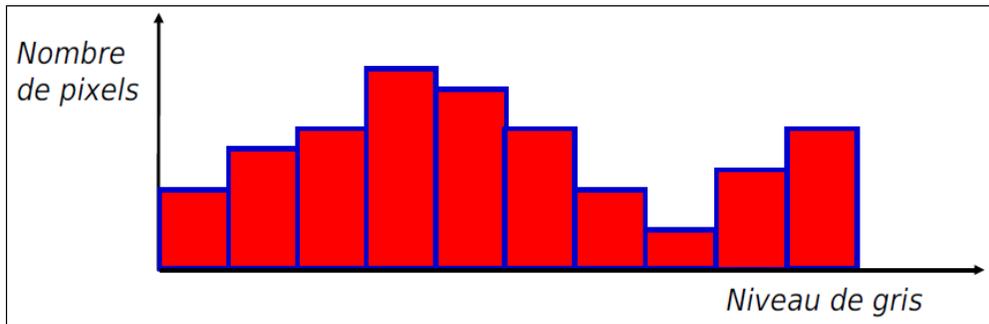
- Un profil d'intensité d'une ligne dans une image est représenté par des signaux 1D.



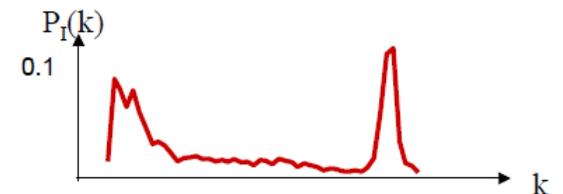
Amélioration de qualité (Niveaux de gris)

Histogramme des niveau 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]*



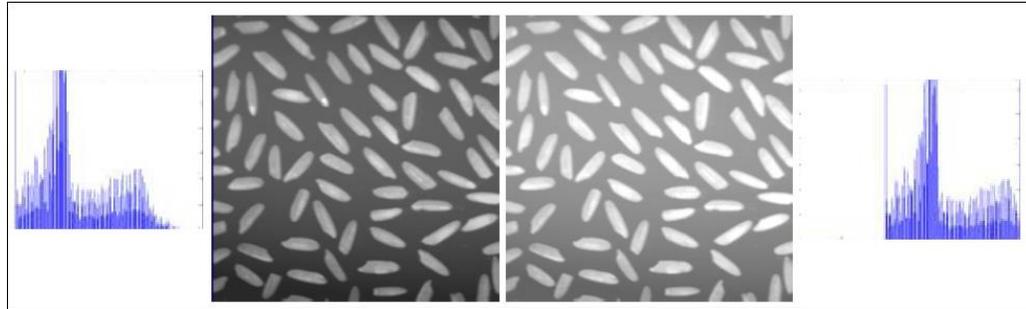
Histogramme normalisé →
 $P(k) = H(k) / Nb_pixels$



Amélioration de qualité (Niveaux de gris)

Luminance (ou brillance)

- La luminance est définie comme la moyenne de tous les pixels de l'image



Contraste

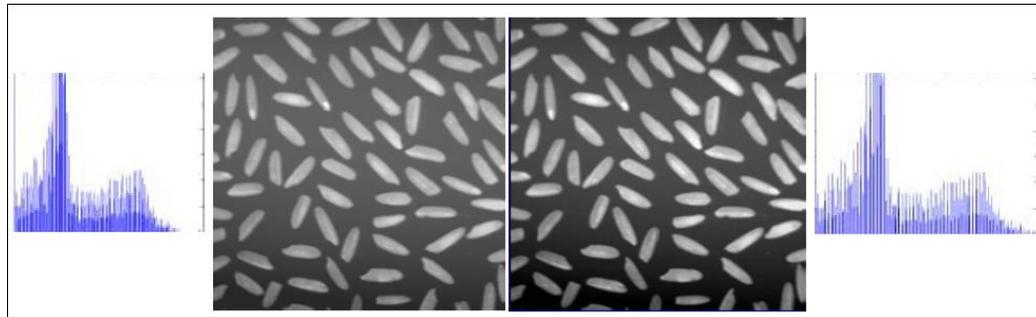
- Le contraste peut être défini de plusieurs façons :

Ecart-type des variations des NdG

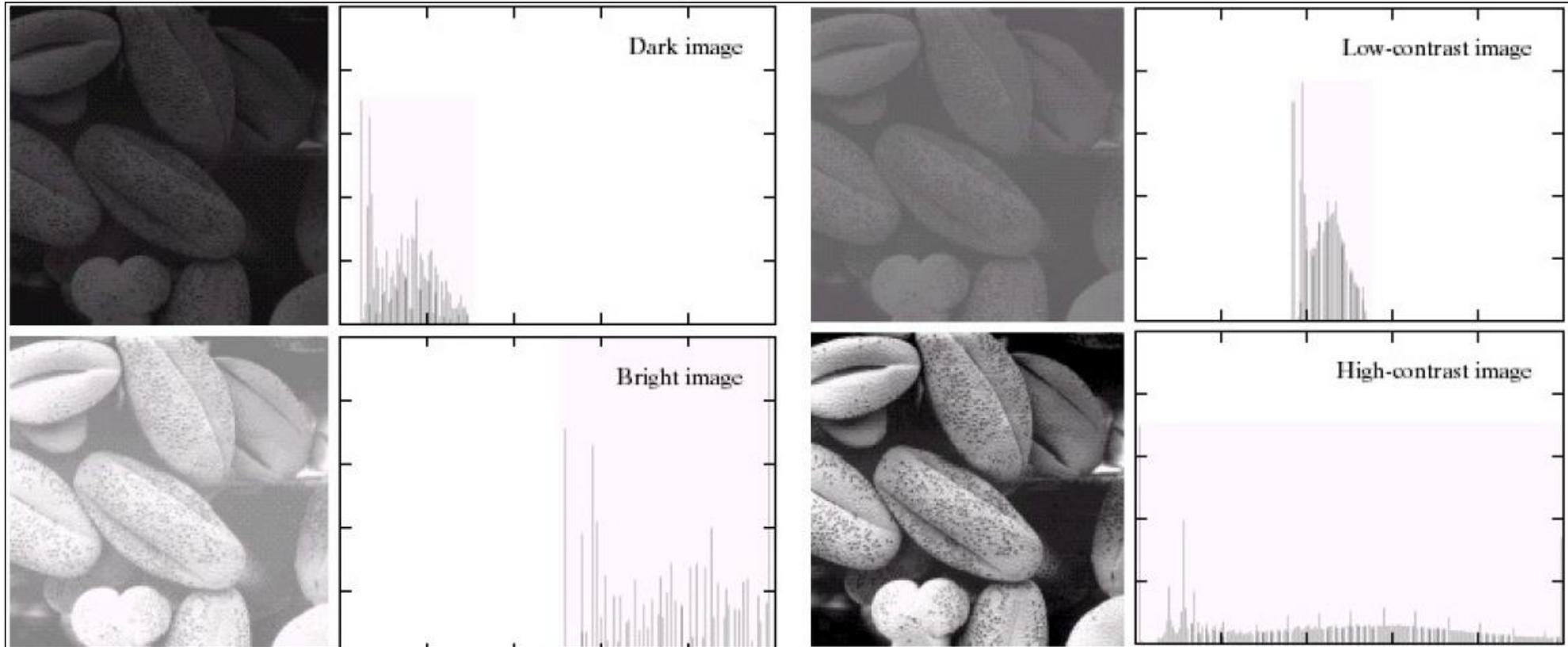
$$C = \sqrt{\frac{1}{M \times N} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} (f(x,y) - Moy)^2}$$

Variation entre NdG min et max

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



Amélioration de qualité (Niveaux de gris)



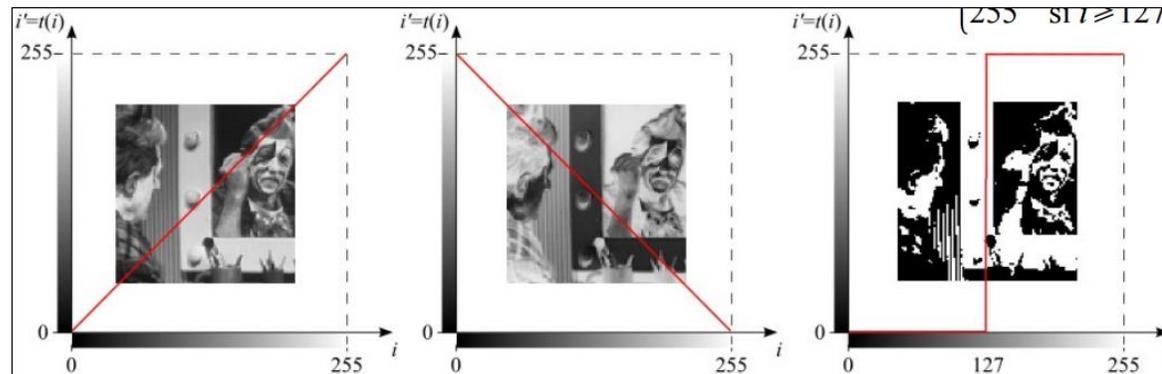
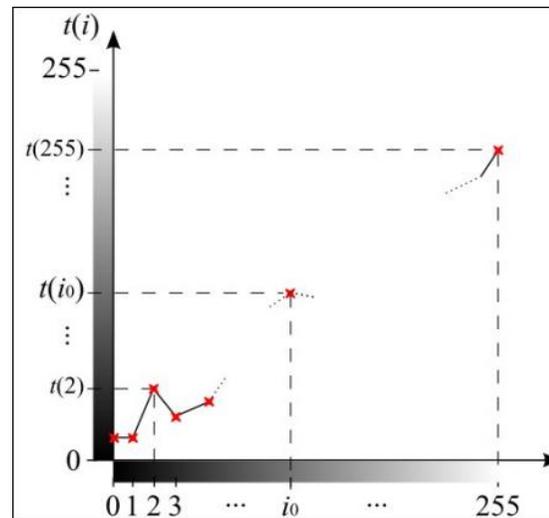
- **Amélioration du contraste**

- Transformation linéaire et Transformation linéaire par morceau
- Transformation non-linéaire
- Égalisation de l'histogramme

Amélioration par Transformation ponctuelle

- Une **transformation ponctuelle** t change un niveau de gris i dans l'image initiale I en un niveau de gris i' pour obtenir l'image résultat I' .
- Elle est donc définie par une table de correspondance ou LUT (Look-Up Table) qui définit, pour chaque niveau de gris i , le nouveau niveau $i'=t(i)$.

i	$t(i)$
0	$t(0)$
1	$t(1)$
2	$t(2)$
3	$t(3)$
⋮	⋮
i_0	$t(i_0)$
⋮	⋮
255	$t(255)$



Transformation linéaire

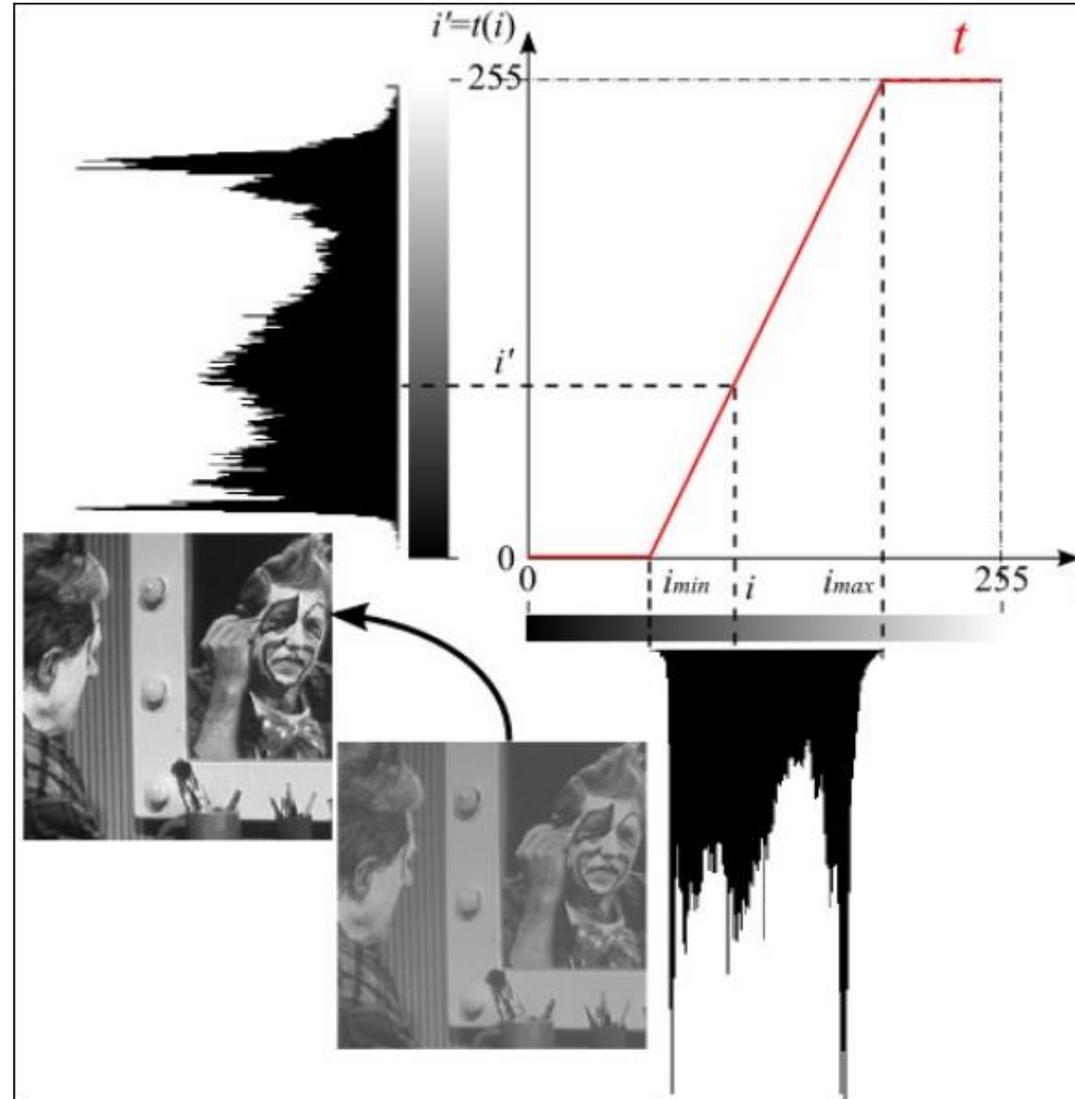
Expansion de la dynamique

- 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

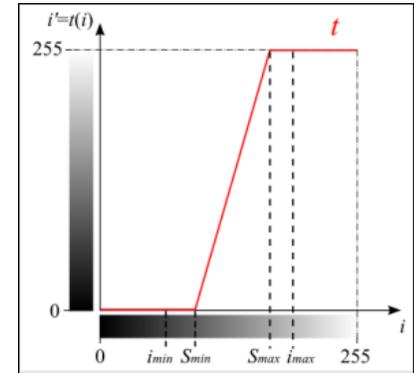
$$\frac{i - i_{min}}{i_{max} - i_{min}} \in [0, 1]$$



Transformation linéaire et non linéaire

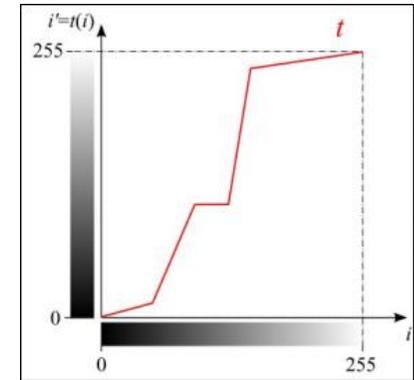
Avec saturation

- Définie par 2 seuils : $i_{min} < S_{min} < S_{max} < i_{max}$
- Rehaussement du contraste des niveaux $S_{min} < i < S_{max}$
- Saturation
 - à 0 des niveaux $i_{min} < i < S_{min}$
 - à 255 des niveaux $S_{max} < i < i_{max}$



Transformation linéaire par morceaux

- Effet : selon la pente locale (gain), expansion ou compression de la dynamique
 - rehaussement du contraste si gain > 1
 - diminution du contraste si gain < 1



Transformation non-linéaire gamma

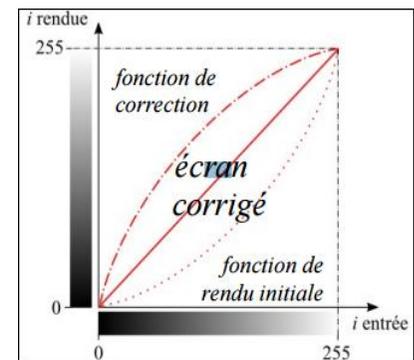
Correction des défauts liés aux dispositifs d'acquisition et écran

Correction γ de ces défauts :

$$i' = 255 \left(\frac{i}{255} \right)^{1/\gamma}$$

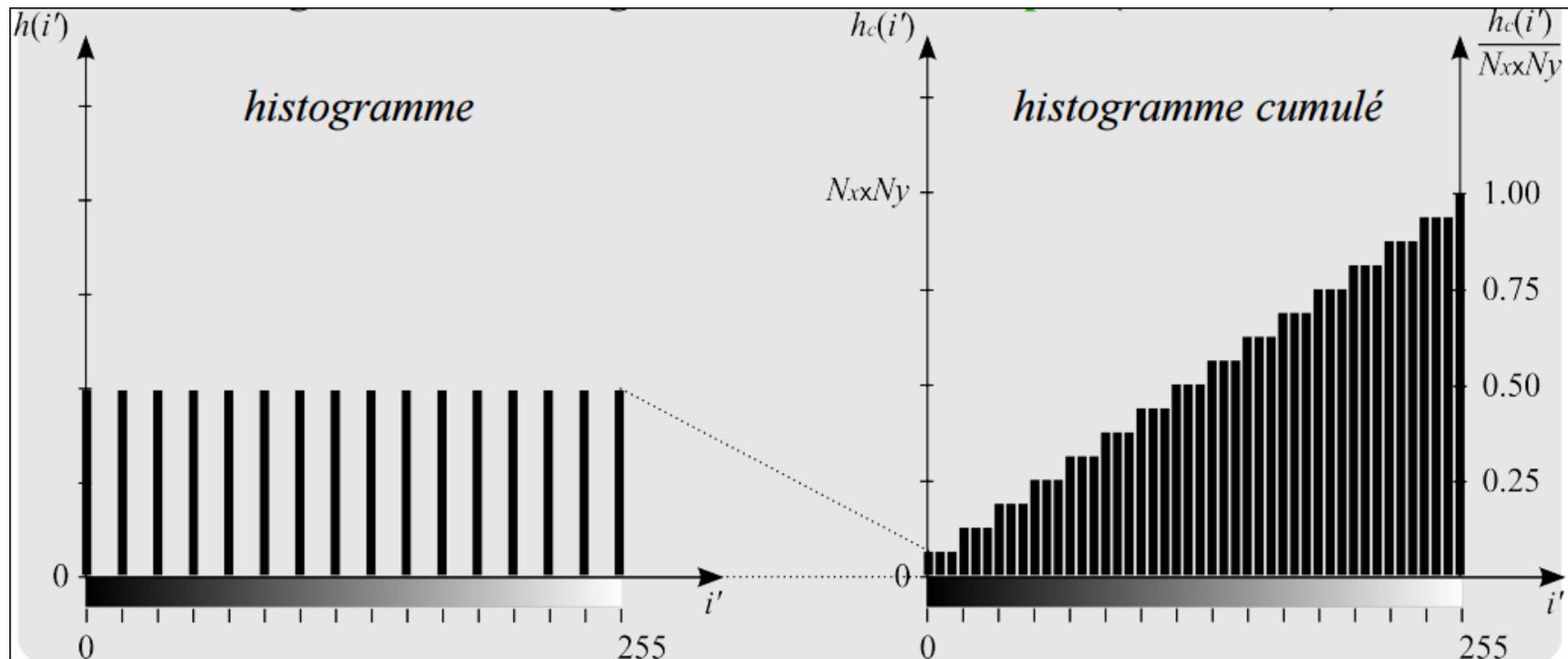
$1/\gamma < 1$ éclaircit principalement les parties foncées

$1/\gamma > 1$ assombrit principalement les parties claires



Egalisation d'histogramme

- On cherche à obtenir une image où les niveaux de gris sont répartis de manière la plus égalitaire possible (contraste maximal).
 - L'histogramme de l'image résultat I' est donc plat (idéalement)
 - Il est plus simple de travailler avec l'histogramme cumulé et normalisé



Egalisation d'histogramme

• Histogramme normalisé

- La probabilité d'apparition d'un niveau de gris i dans l'image est

$$h_n(i) = \frac{h(i)}{N_x \times N_y} \quad , \quad h_n(i) \in [0, 1]$$

- La fonction correspondante h_n est l'**histogramme normalisé**.

• Histogramme cumulé

- L'**histogramme cumulé** dénombre les occurrences cumulées de chacun des niveaux :

$$h_c(i) = \sum_{k=0}^i h(k) \quad , \quad h_c(i) \in [0, N_x \times N_y]$$

- Il est défini de façon récursive par
$$\begin{cases} h_c(0) & = & h(0) \\ h_c(i) & = & h_c(i-1) + h(i) \end{cases}$$

- La probabilité qu'un pixel ait un niveau de gris inférieur ou égal à i est

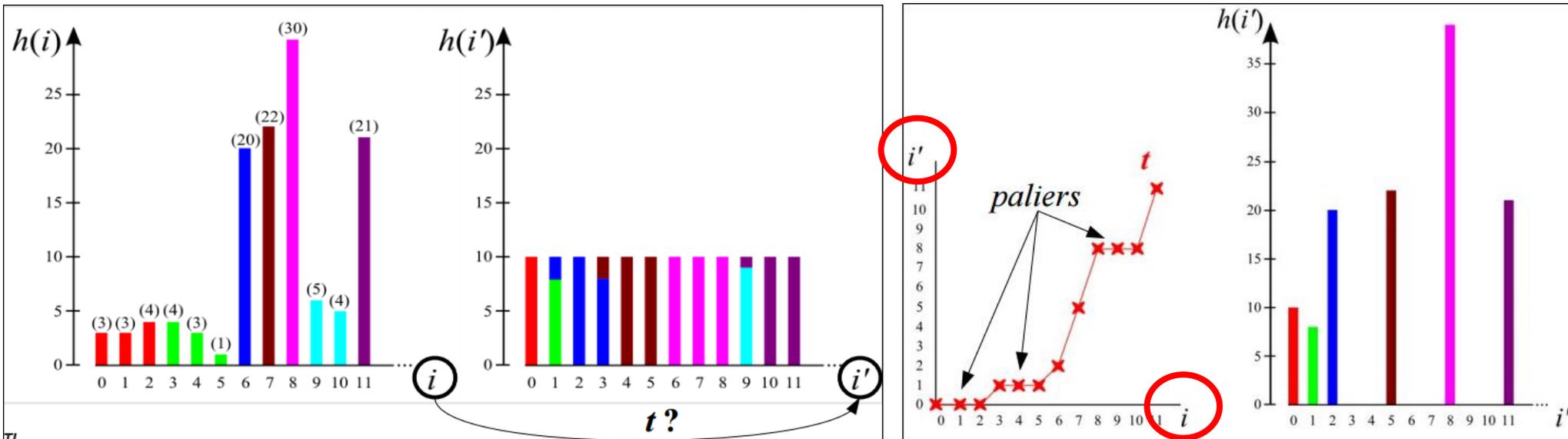
$$\frac{h_c(i)}{N_x \times N_y} \quad , \quad \text{avec} \quad \frac{h_c(255)}{N_x \times N_y} = 1$$

Egalisation d'histogramme

- On cherche une fonction de transformation $t : i \rightarrow i'$
 - croissante (i.e. préservant l'ordre des niveaux de gris)
 - qui génère (autant que possible) un histogramme $h(i')$ « plat »,
 - c'est-à-dire une distribution uniforme des niveaux de gris
- La transformation $t : i \rightarrow i'$ cherchée est : $E(\bullet)$ désigne la fonction partie entière.

$$i' = E\left(\frac{256}{N_x \times N_y} h_c(i) - 1\right), \text{ avec } h_c(i) = \sum_{k=0}^i h(k)$$

- Certains niveaux de gris disparaissent par regroupement



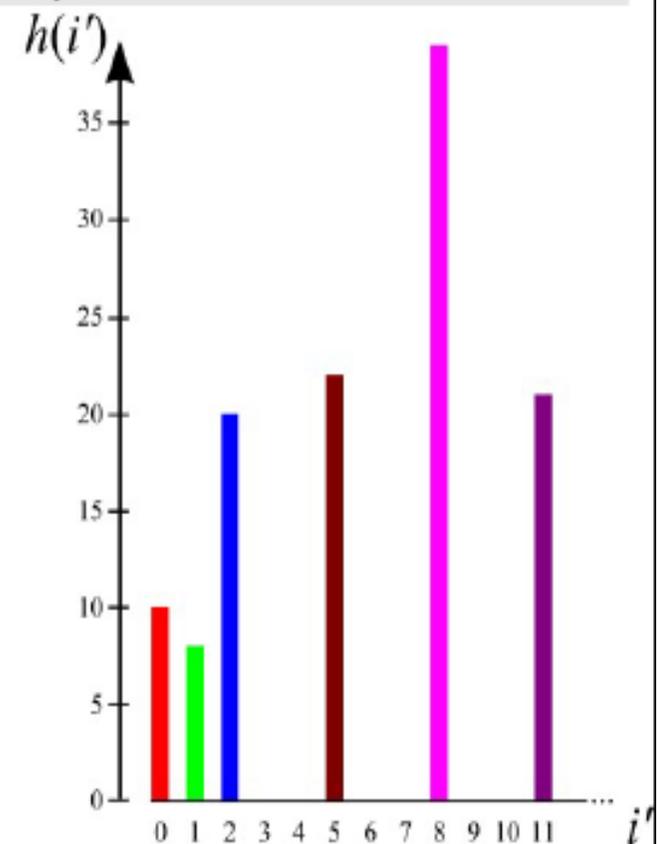
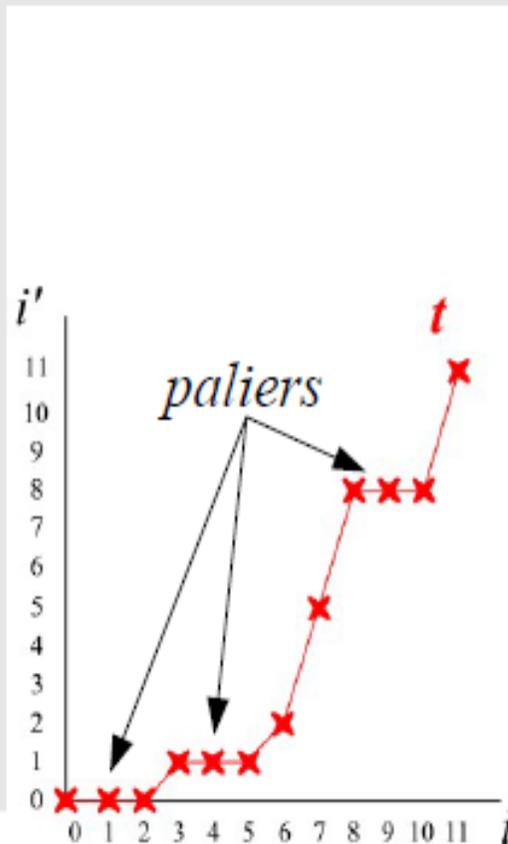
Egalisation d'histogramme

$$i' = E \left(\frac{256}{N_x \times N_y} h_c(i) - 1 \right), \text{ avec } h_c(i) = \sum_{k=0}^i h(k)$$

→ Exemple simplifié avec 12 niveaux (non 256) et $N_x \times N_y = 120$ pixels

i	$h(i)$	$h_c(i)$	$i'=t(i)$
0	3	3	0
1	3	6	0
2	4	10	0
3	4	14	1
4	3	17	1
5	1	18	1
6	20	38	2
7	22	60	5
8	30	90	8
9	5	95	8
10	4	99	8
11	21	120	11

i'	$h(i')$
0	10
1	8
2	20
3	0
4	0
5	22
6	0
7	0
8	39
9	0
10	0
11	21



Egalisation d'histogramme

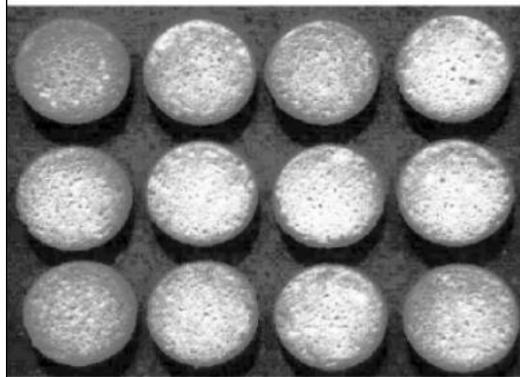
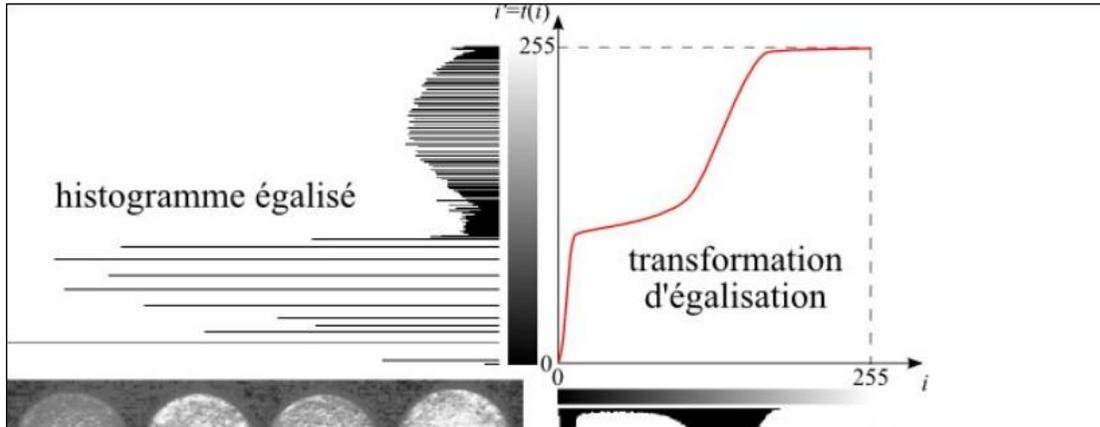


Image résultat

Image initiale

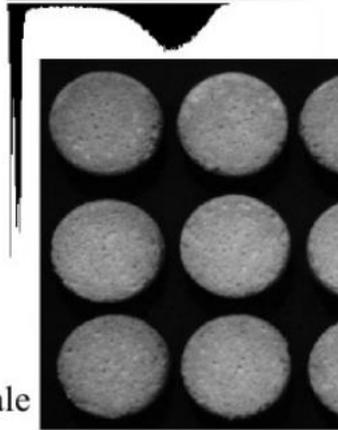


Image initiale

Histogrammes initial et initial cumulé

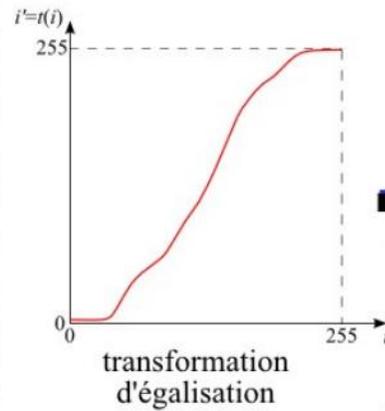
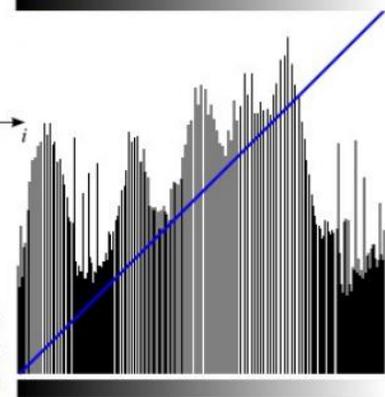


Image résultat

Histogrammes égalisé et égalisé cumulé



Egalisation d'histogramme

- Résumé de la méthode et remarques
 - L'égalisation de l'histogramme consiste à regrouper des ensembles de niveaux de gris de valeurs voisines, pour obtenir un nouveau niveau de gris en quantité approchant $N_x \times N_y / 256$;
 - Fournit de nouveaux niveaux uniformément répartis sur $[0, 255]$.
- Remarques sur l'amélioration par égalisation d'histogramme :
 - Peut améliorer une image quand la correction de dynamique est inefficace.
 - On obtient une image améliorée pour la visualisation (amélioration subjective).
 - Il est impossible d'obtenir une égalisation parfaite.
- Le regroupement de niveaux initiaux (paliers de t) implique :
 - une perte objective d'information ;
 - que cette transformation n'est pas réversible.

Fil rouge : Format d'images

Implémentation d'algorithmes de transformation ponctuelle

- Double boucles ?

```
Pour i=1 à nblig  
  Pour j=1 à nbcot  
    I'(i,j) = 255*(I(i,j)-minI)  
              /(maxI-minI);
```

Pas optimal



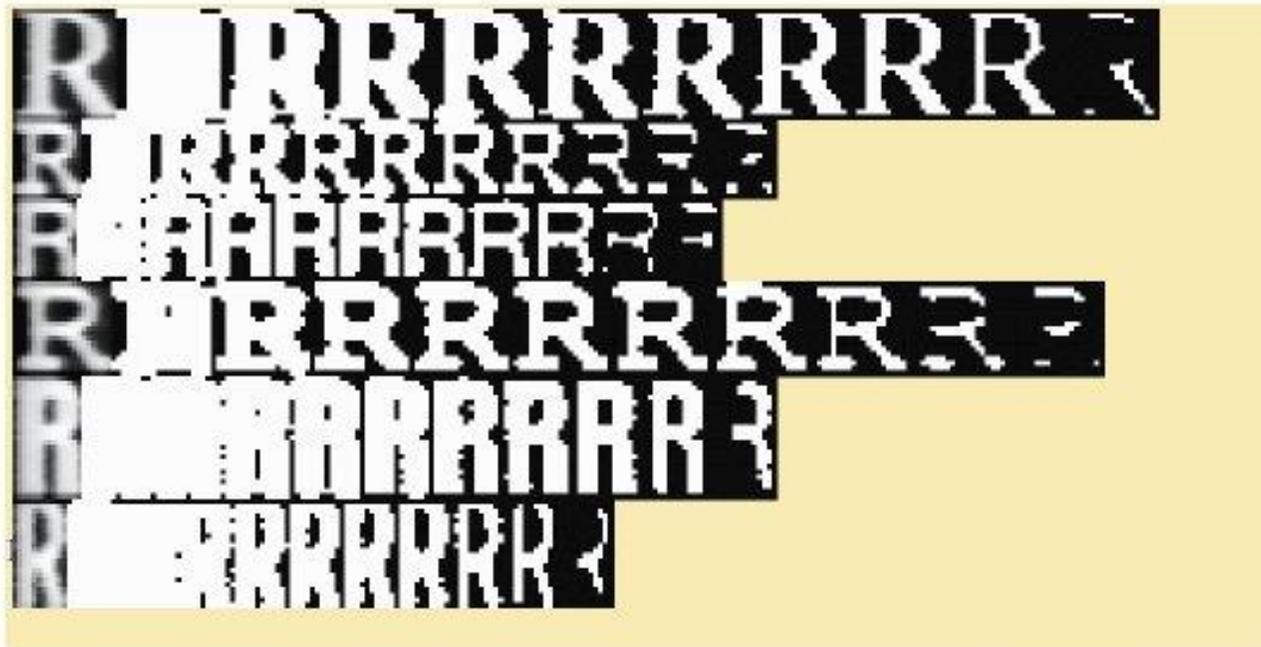
Fil rouge : Format d'images

Couleurs → Niveau de gris → Binaire

- **Attention : les apparences sont souvent trompeuses !**
- Certains traitements d'images ne sont applicables que sur certains types d'images (couleurs, niveau de gris, binaire)
- La plupart concerne les images en niveaux de gris
- Les images doivent alors être converties dans le bon format → bonne profondeur (**nombre de bits par pixel**)
- Couleurs → Niveau de gris = *Recommandation ITU-R BT 709*
$$\text{Luminance} = 0,2126 \times \text{Rouge} + 0,7152 \times \text{Vert} + 0,0722 \times \text{Bleu}.$$
- Niveau de gris → binaire = **algorithmes de binarisation**

Binarisation par seuillage

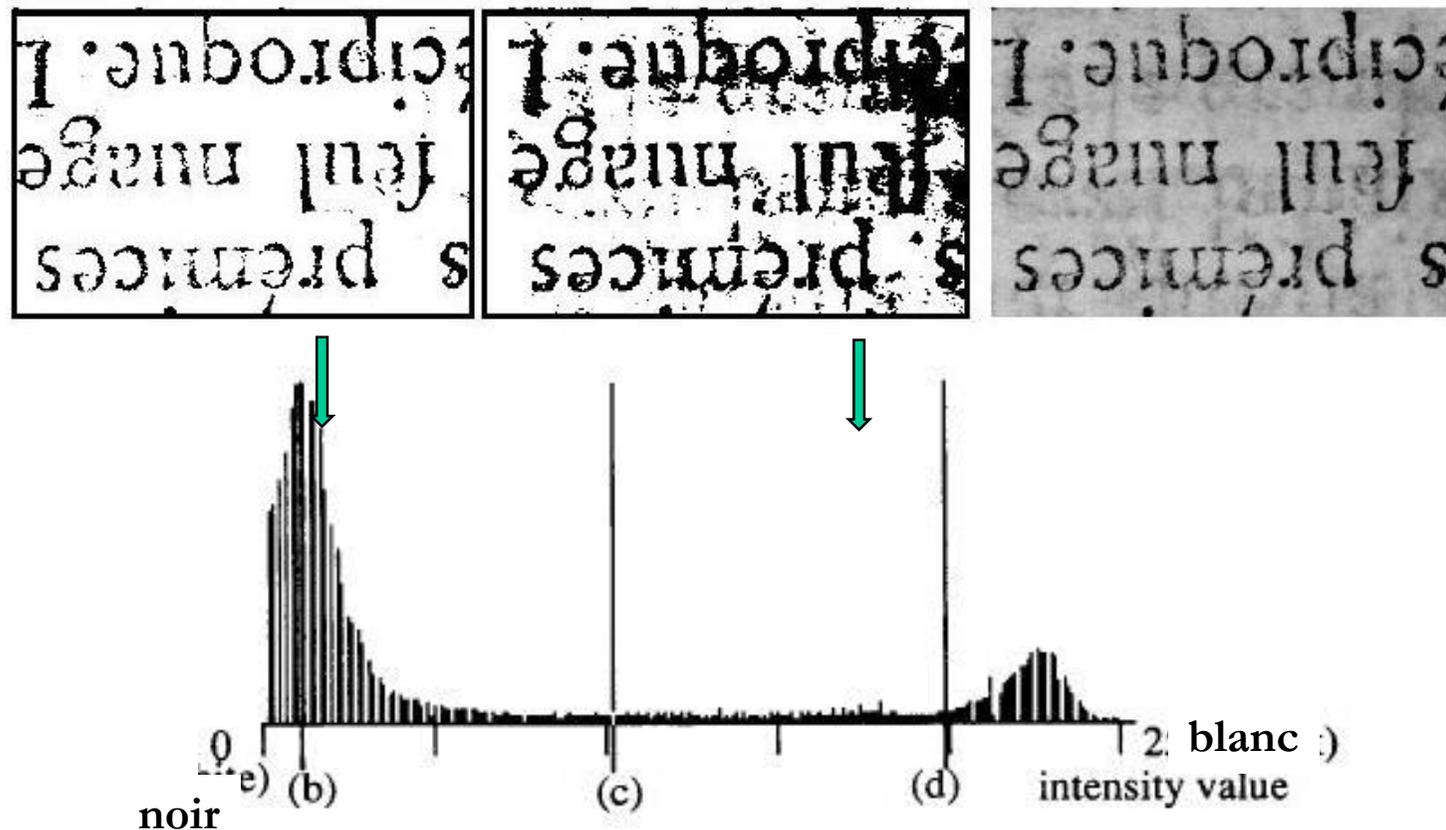
- **Seuillage manuel** → C'est la méthode la plus simple et la plus utilisée
- Il y a une relation entre les niveaux de gris d'un pixel et son appartenance ou non à une forme



Niveaux de gris 0 Noir Blanc 255

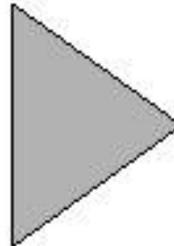
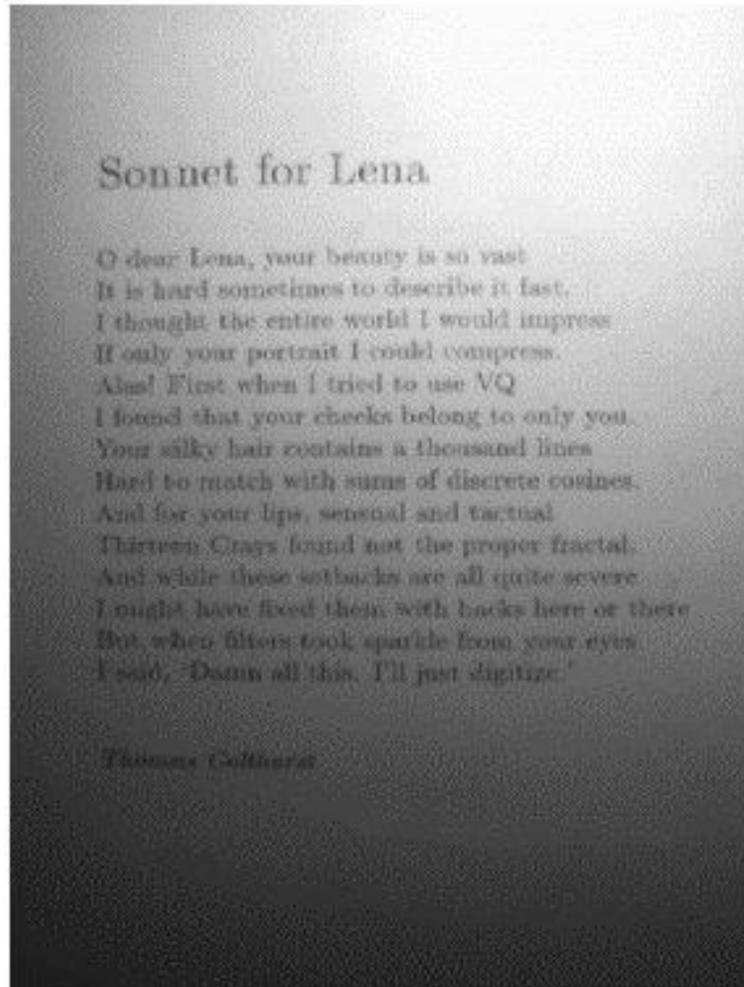
Binarisation par seuil global fixe

- Comment choisir le bon seuil → une multitude de méthodes
 - Différents objectifs : tramage, segmentation, ...
 - Otsu, ...
 - **Voir durant les TP ImageJ**



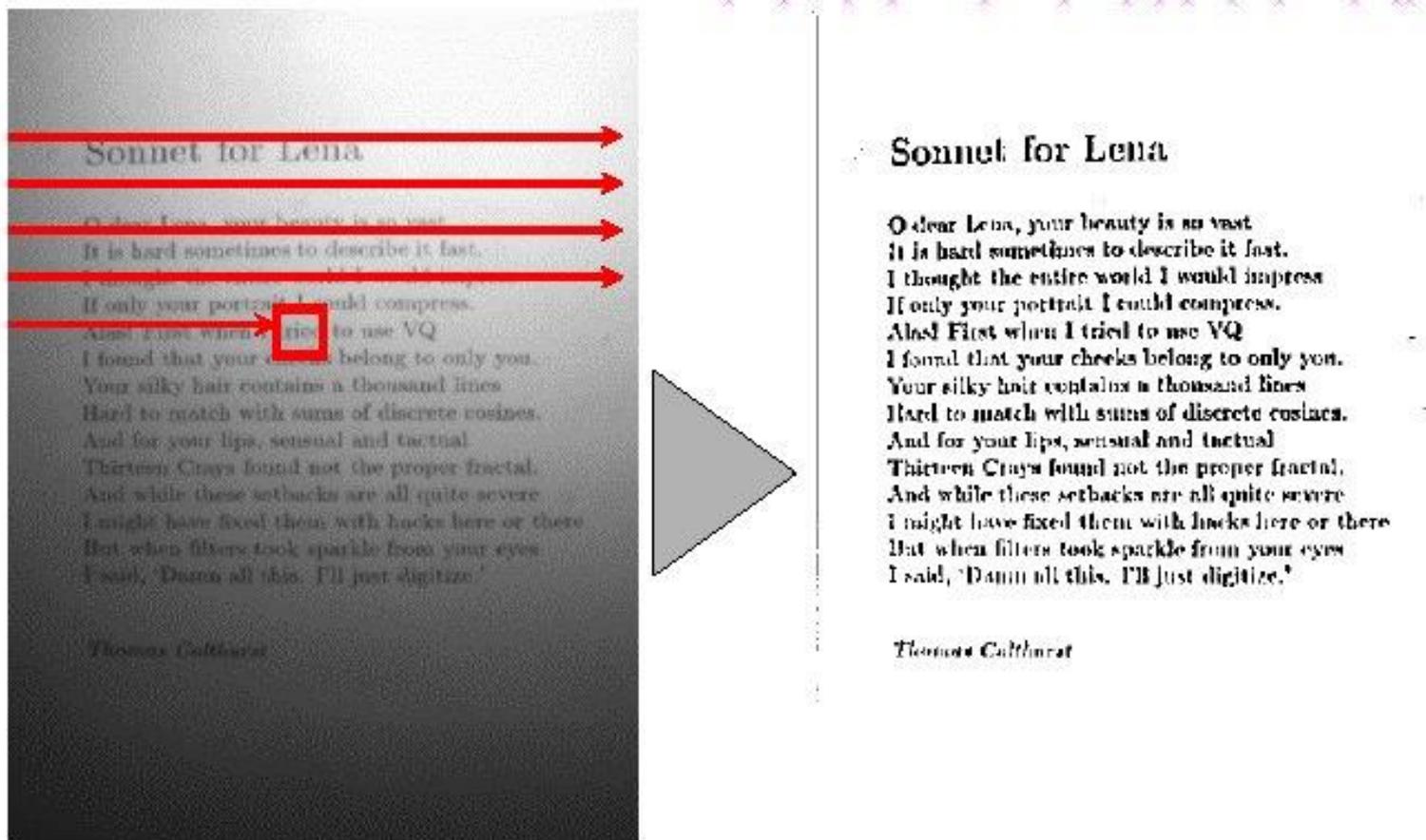
Inconvénient d'un seuil global

- Idées de solution ?



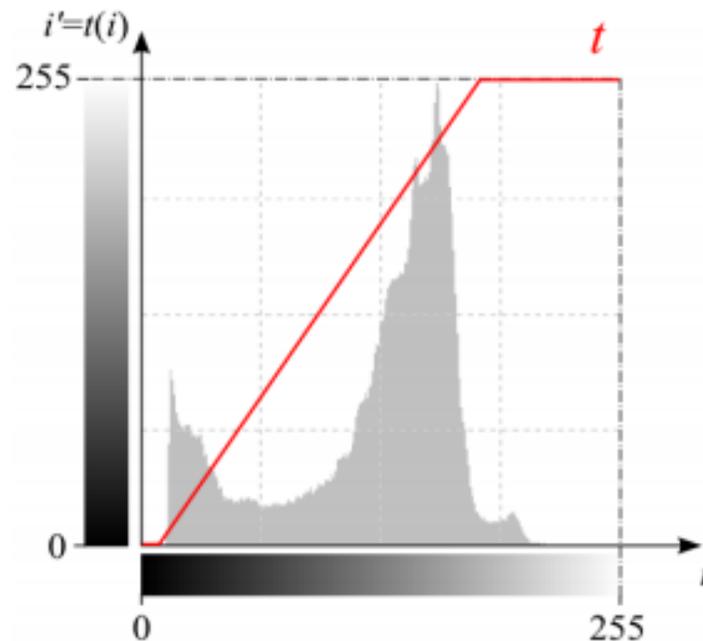
Seuillage adaptatif

- 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



Exercice

Expliquer quel sera l'effet de la transformation ponctuelle t suivante sur l'image dont l'histogramme est représenté en filigrane sous la courbe de t . Tracer (approximativement) l'histogramme de l'image résultat I' .



Transformation t et histogramme de l'image I .

Analyse d'histogrammes....

Luminosité ? Contraste ? Egalisation, seuillage, égalisation ou ?

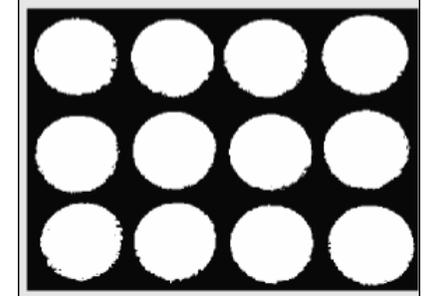
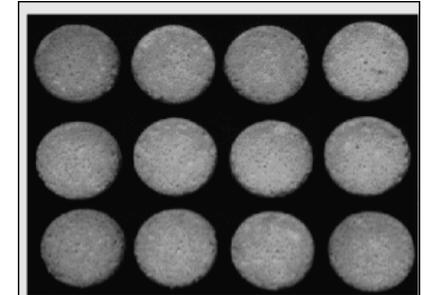
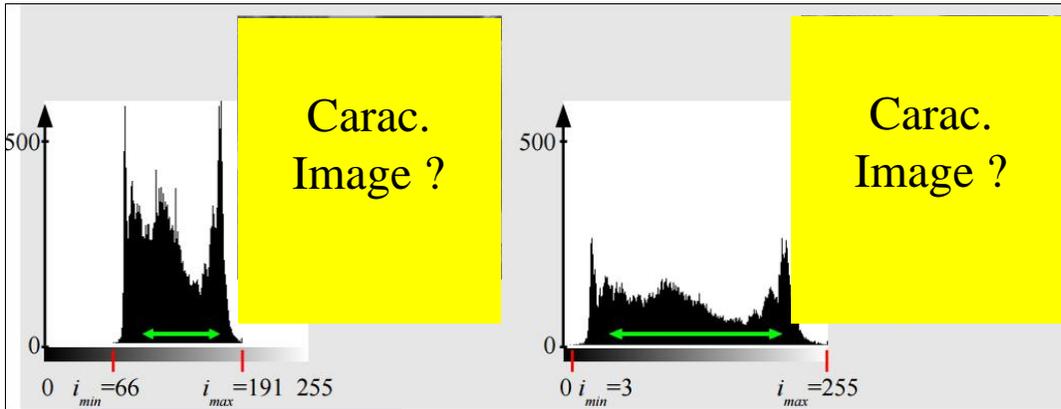
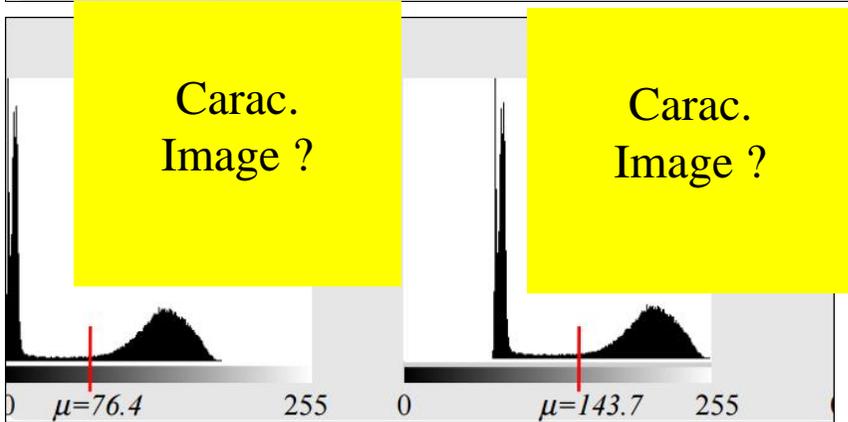
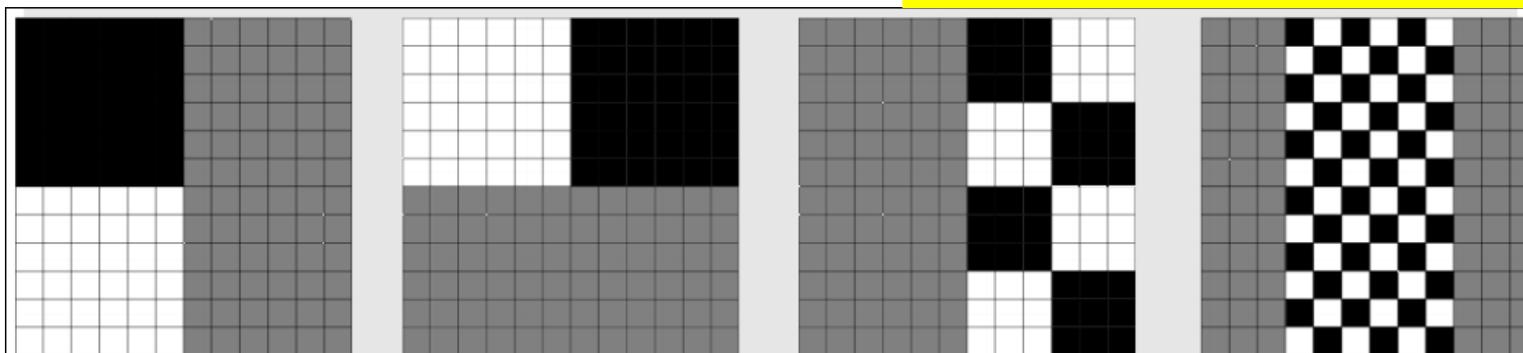


Image binarisée au seuil $i_0=63$



Quels Histogrammes ?



Quel
Traitement ?
Pourquoi ?

Fil rouge : Binarisation ?

- Binarisation et détection de complexité ?
- Seuillages multiples
- Fonctions Puits

Opérations logiques & arithmétiques

Principe :

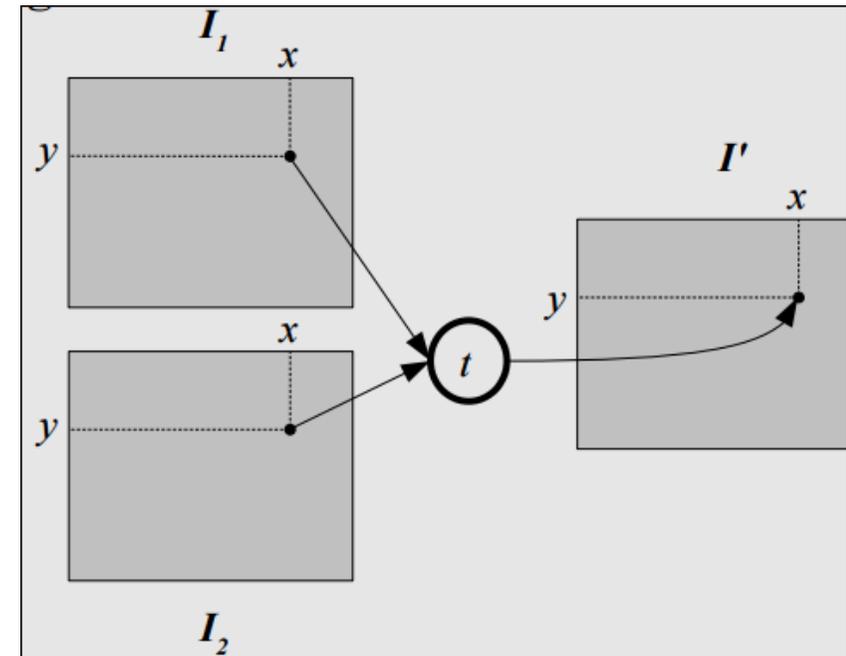
- Appliquer, pixel à pixel, les opérations logiques et arithmétiques classiques à deux (ou plusieurs) images
- Les images opérandes doivent être de même taille
- peuvent être des images constantes.

Exemples :

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

Problèmes :

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

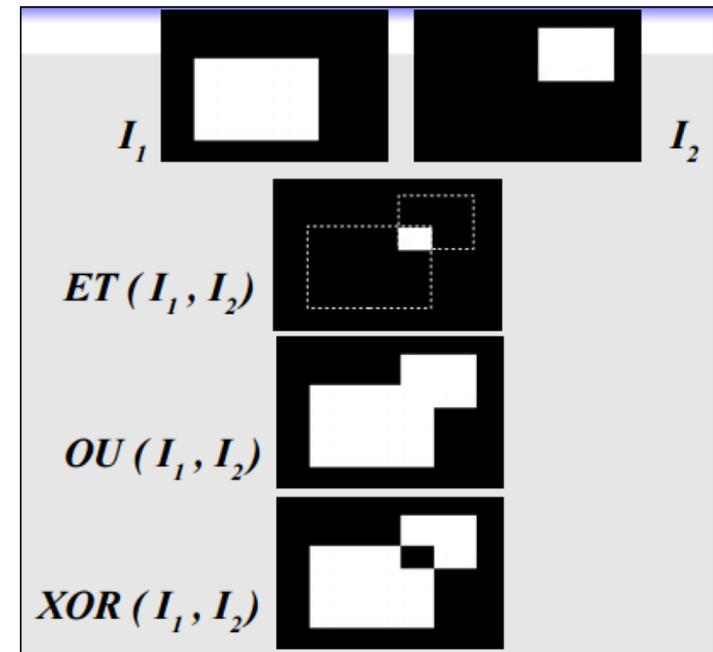
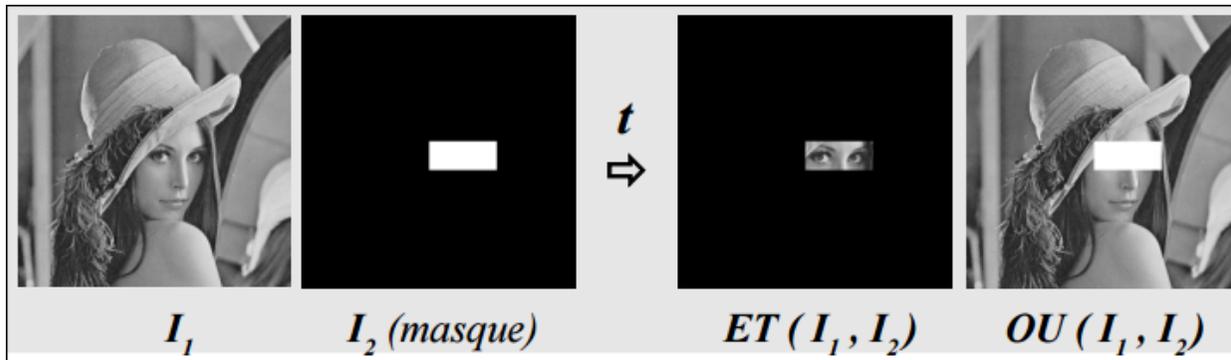


$$I_1(x, y), I_2(x, y) \xrightarrow{t} I'(x, y) = t(I_1(x, y), I_2(x, y))$$

Opérations logiques et arithmétiques

Opérations logiques

- ET, OU, XOR
- Bit à bit meme pour les images en niveaux de gris !
- Utile surtout en binaire ou pour du masquage



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 luminosité d'une image (par addition d'une constante ou d'une image avec elle-même)
 - Diminution du bruit dans une série d'images

Opérations arithmétiques

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

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 I_1 et de I_2
- 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

