

ANALYSE D'IMAGES ET DE VIDÉOS

Année académique: 2023-2024

BONJOUR

MAHRAZ Mohamed Adnane

Professeur Habilité à la faculté des sciences Dhar El Mahraz

Membre permanant au Laboratoire de recherche LISAC

Pour me contacter:

Mon bureau: Département Informatique I207

Téléphone: 0676596104

Email: adnane_1@yahoo.fr

PROGRAMME DU COURS (PARTIE 1/2)

Introduction

Qu'est ce qu'une image ?

Traitements de base d'une image

Opérations sur les images

Filtrage

Détection des contours

Segmentation

Extraction de caractéristiques

Reconnaissance de formes

VISION ARTIFICIELLE (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 un capteur

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 : reconnaissance

- Utilise des techniques d'intelligence artificielle pour identifier des formes connues dans l'image

EXTENSION DE LA PERCEPTION HUMAINE ? OUI



Capteurs meilleurs que l'œil humain

- Voient ce que nous ne voyons pas directement, du fait de la limitation de notre système visuel

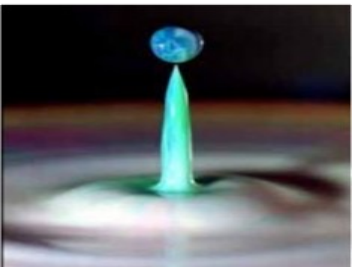


Autres propriétés optiques

- 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

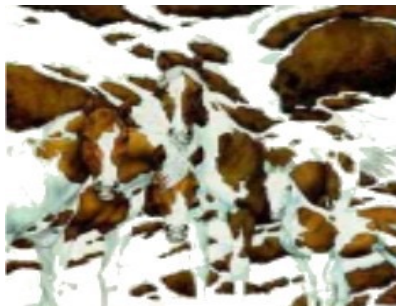
- Caméras à haute cadence, plusieurs milliers d'images par seconde.

MIEUX QUE LA PERCEPTION HUMAINE ? NON



Moins « intelligent » que l'humain

- Même si les images initiales sont parfois très riches, le résultat de leur traitement par un système de VA reste souvent basique



Puissance de calcul

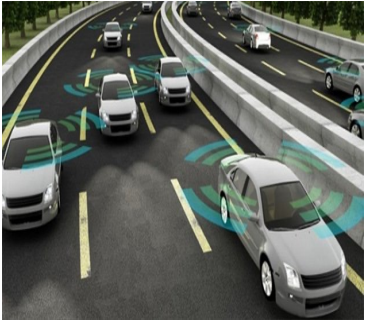
- Les systèmes de VA ont une capacité de calcul infiniment plus limitée que celle du cerveau humain, ou de celui des animaux évolués

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



EXEMPLE D'APPLICATION DE LA VISION ARTIFICIELLE : VÉHICULE AUTONOME



Objectifs

- Aider le conducteur à prendre des décisions pour plus de sécurité

Eviter les obstacles

- Véhicules équipés par un système de contrôle capable d'éviter les obstacles



Détection et reconnaissance

- Véhicules équipés par des caméras capables de détecter et de reconnaître les panneaux routiers et le compare avec une base de données interne. Une fois le panneau reconnu, le conducteur est averti de la situation grâce à un visuel sur le GPS ou l'instrumentation.



EXEMPLE D'APPLICATION DE LA VISION ARTIFICIELLE : IMAGERIE MÉDICALE



Objectifs

- Aider le médecin lors du diagnostic, le chirurgien lors de la réalisation d'un geste opératoire

Amélioration des images

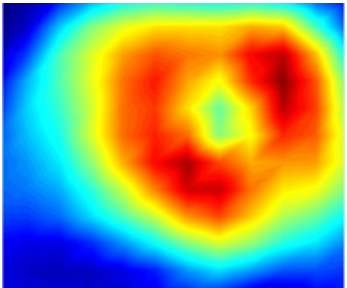
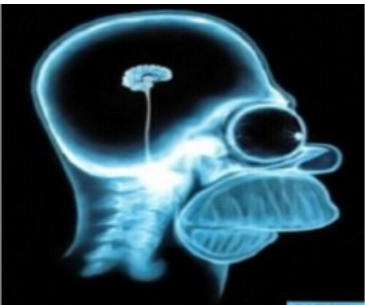
- Rehaussement du contraste, élimination du bruit, mise en évidence des détails

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



VISION INDUSTRIELLE : CONTRÔLE DE LA QUALITÉ, DE L'ASPECT, ...



Objectifs

Eviter 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

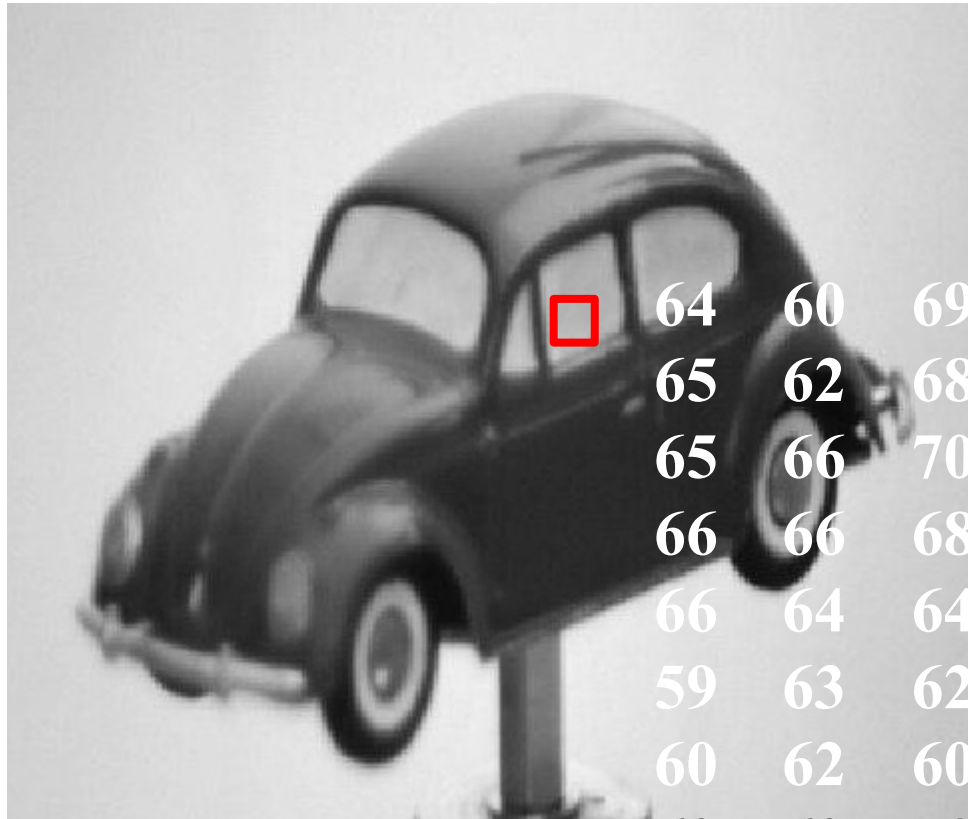
- A partir des données précédentes, le système détermine la qualité d'un produit



TRAITEMENT D'IMAGES

Introduction à l'image

QU'EST-CE QU'UNE IMAGE ?



Niveaux de gris - 8 bits:

0 - noir

255 - blanc

64	60	69	100	149	151	176	182	179
65	62	68	97	145	148	175	183	181
65	66	70	95	142	146	176	185	184
66	66	68	90	135	140	172	184	184
66	64	64	84	129	134	168	181	182
59	63	62	88	130	128	166	185	180
60	62	60	85	127	125	163	183	178
62	62	58	81	122	120	160	181	176
63	64	58	78	118	117	159	180	176

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 une matrice de nombres représentant un signal
- Plusieurs outils permettent de manipuler ce signal

D'un point de vue humain :

- Une image contient plusieurs informations sémantiques
- Il faut interpréter le contenu au-delà de la valeur des nombres

IMAGES NATURELLES ET ARTIFICIELLES

Image naturelle – Plusieurs moyens d'acquisition

- caméra, microscope, tomographie, infra-rouge, satellite, ...

Image artificielle – Plusieurs outils de représentation

- synthèse d'images, réalité virtuelle, visualisation scientifique, ...



Image naturelle



Image artificielle

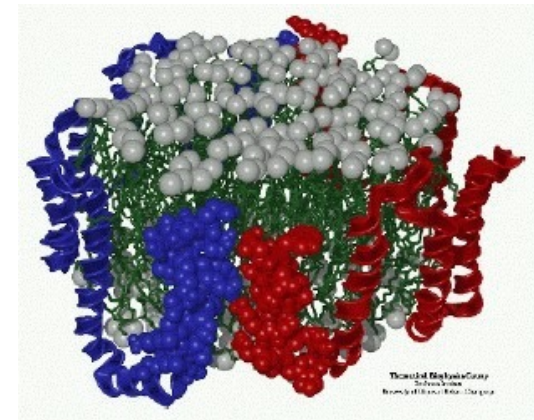


Image artificielle

TROIS PRINCIPAUX TYPES D'IMAGES

... et plus encore (image 3D, image réelle, ...)



Images en niveaux de gris
 $I(x,y) \in [0..255]$

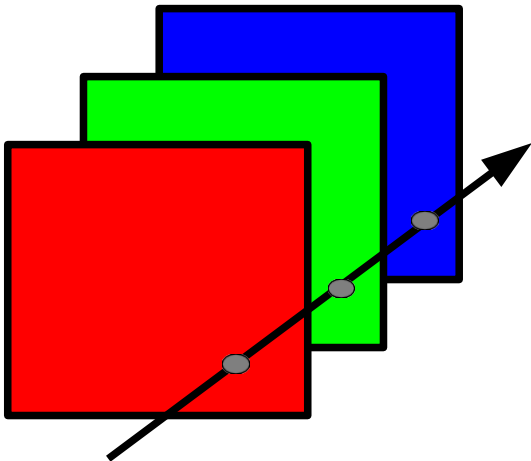
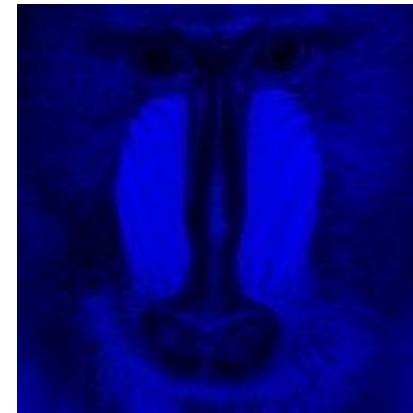


Images binaires
 $I(x,y) \in \{0, 1\}$



Images couleurs
 $I_R(x,y) \ I_G(x,y) \ I_B(x,y)$

IMAGE COULEUR DANS L'ESPACE RVB



*Mais il existe aussi
d'autres codages de la
couleur que RVB...*

ACQUISITION DES IMAGES

ACQUISITION D'UNE IMAGE

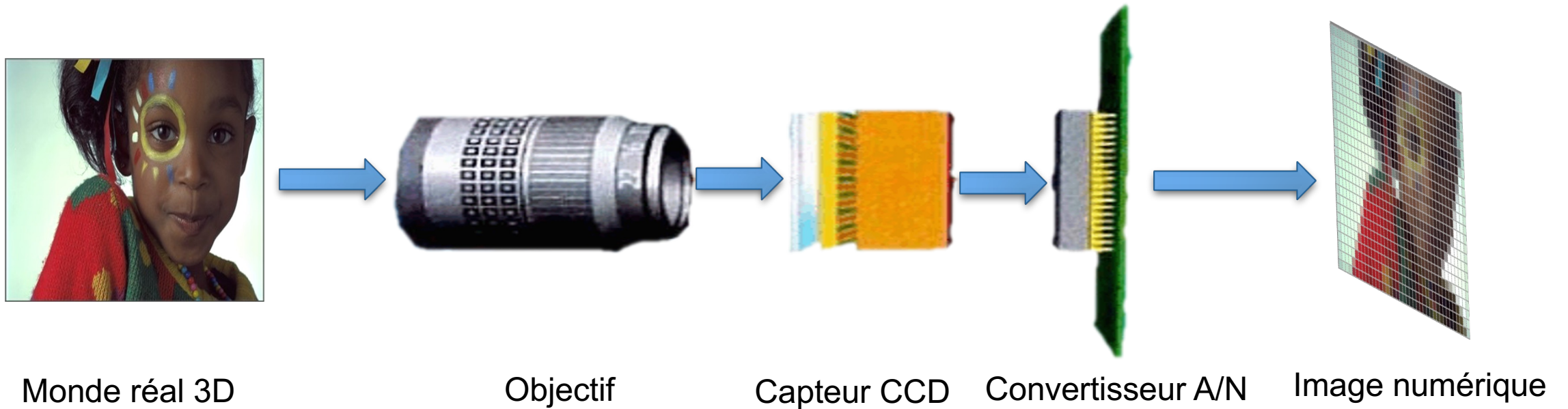


IMAGE NUMÉRIQUE

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

ECHANTILLONNAGE ET QUANTIFICATION

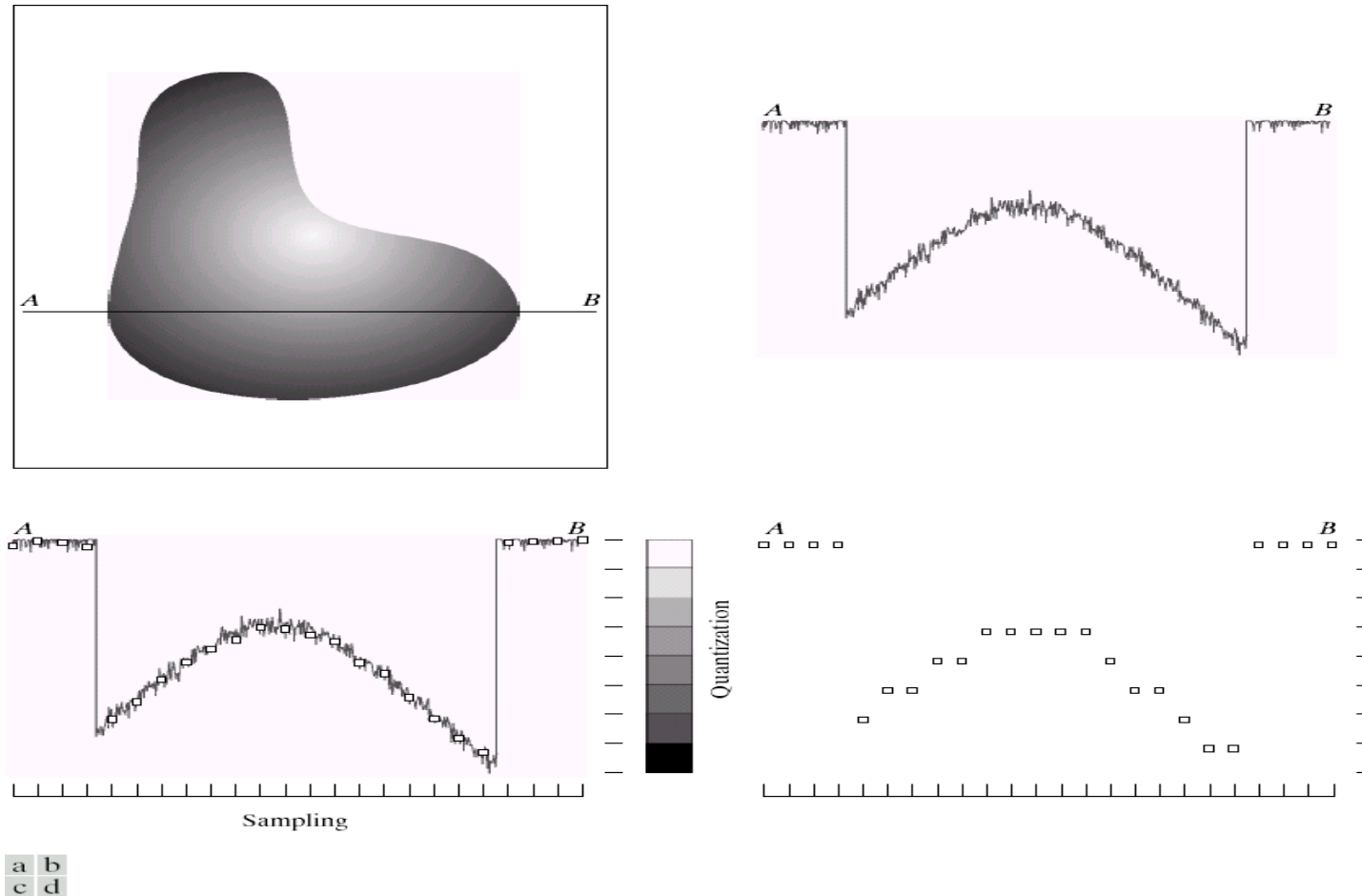
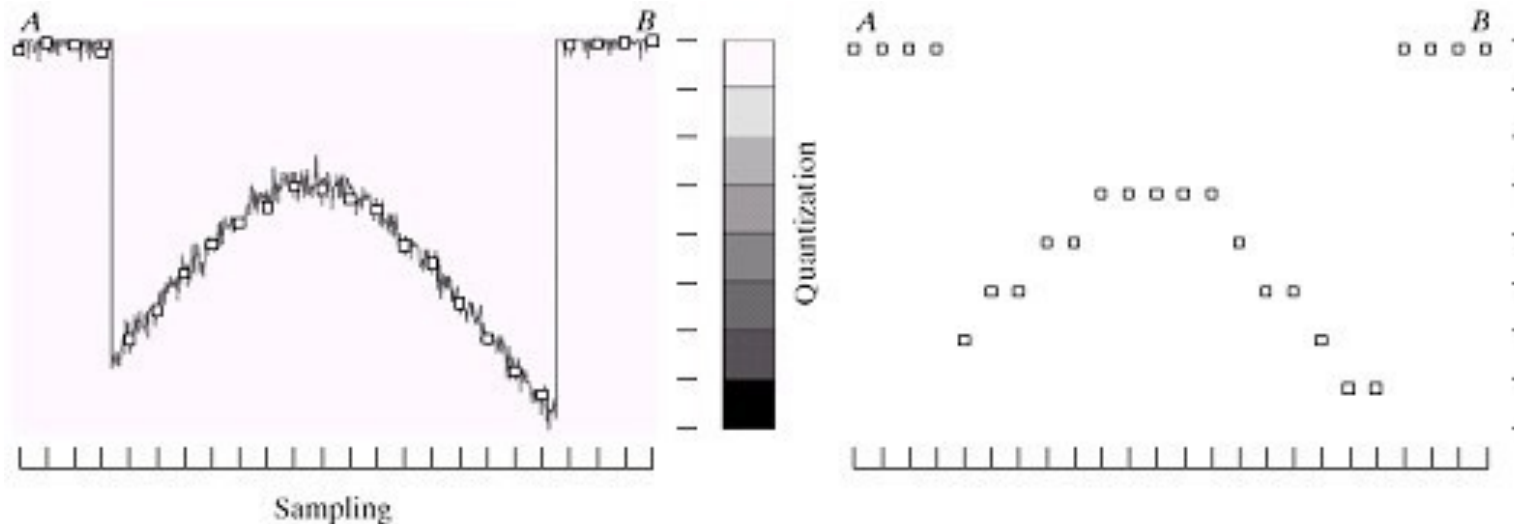


FIGURE 2.16 Generating a digital image. (a) Continuous image. (b) A scan line from *A* to *B* in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.

ECHANTILLONNAGE ET QUANTIFICATION

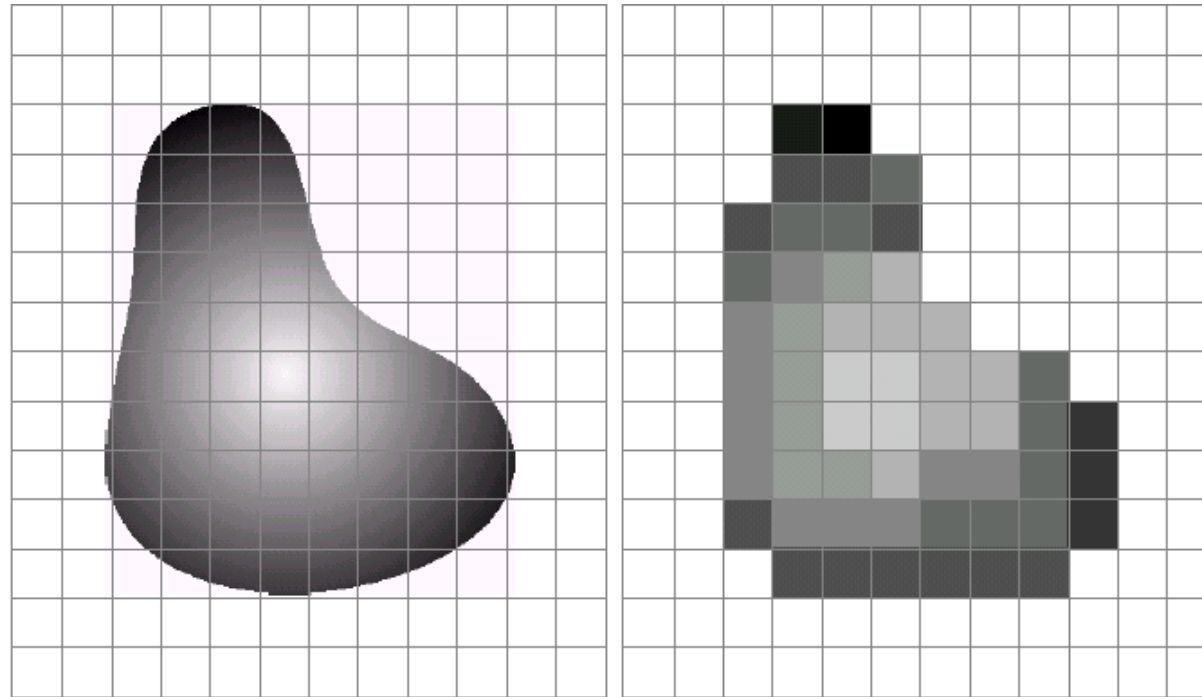
L'échantillonnage est limité par la capacité du capteur, donc le nombre de pixels disponible (ou autre limite imposée)

La quantification est limitée par la quantité de tons (de gris) définie dans l'intervalle



ECHANTILLONNAGE ET QUANTIFICATION

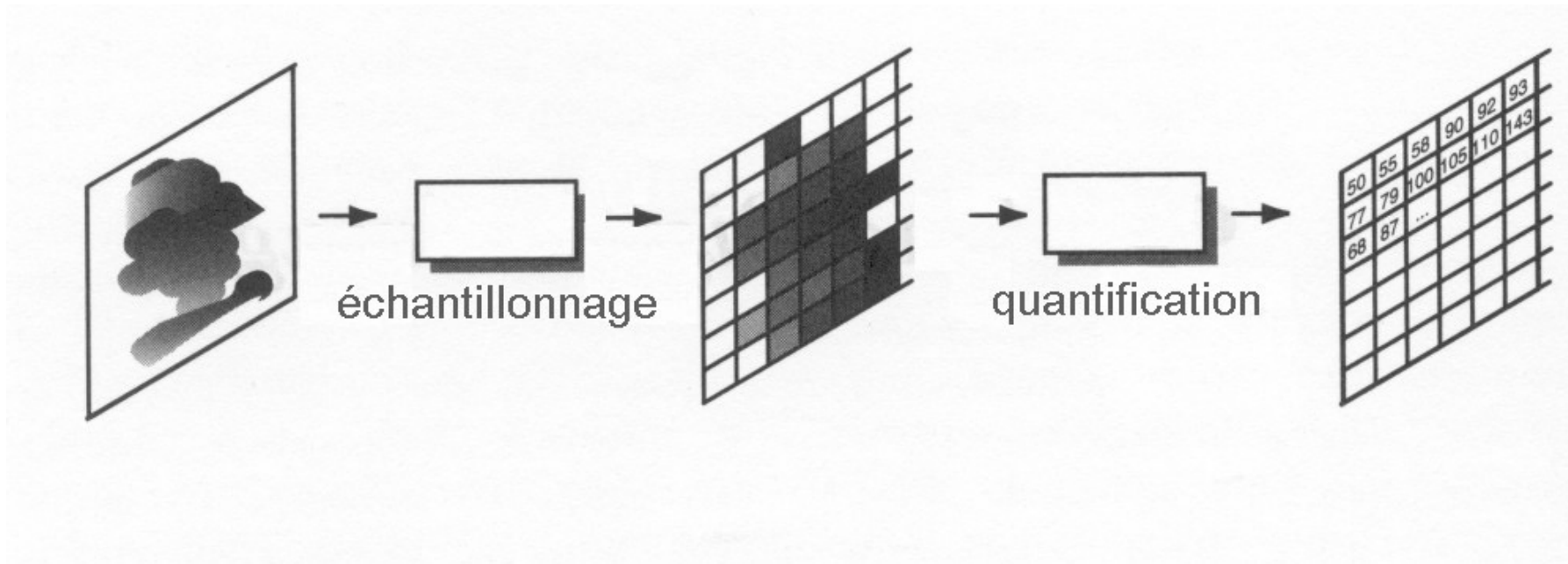
Avec un capteur à matrice :



a b

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

ECHANTILLONNAGE ET QUANTIFICATION



REPRÉSENTATION DES IMAGES

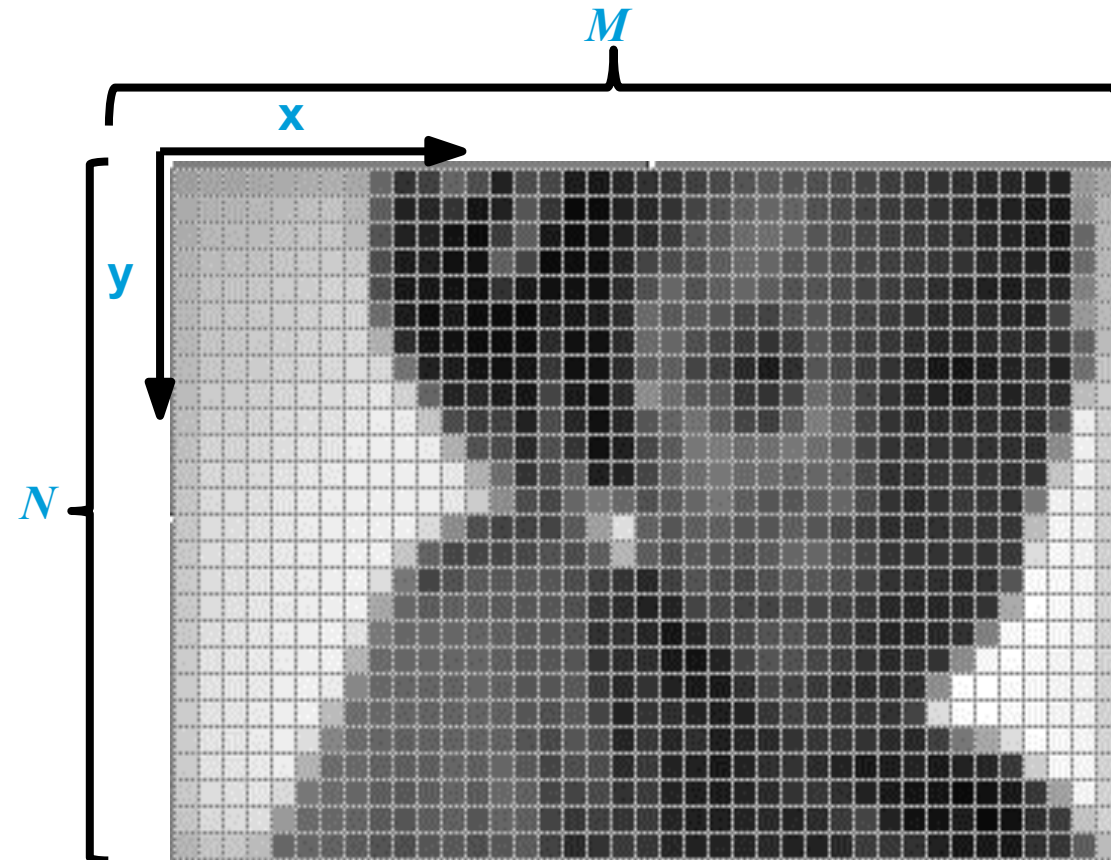
Une Image numérique monochrome est une:

- Matrice de dimension **$M \times N$**
- Chaque élément à une valeur entière dans l'intervalle **$[L_{\min} / L_{\max}]$**
- Le nombre de « bits » requis pour représenter les niveaux de gris dans l'intervalle « L » est « K », avec :

$$L = 2^K$$

- Le nombre de bits pour entreposer une image est donc :

$$b = M \times N \times K$$



RÉSOLUTIONS DES IMAGES

Résolution spatiale

- Le plus petit détail discernable

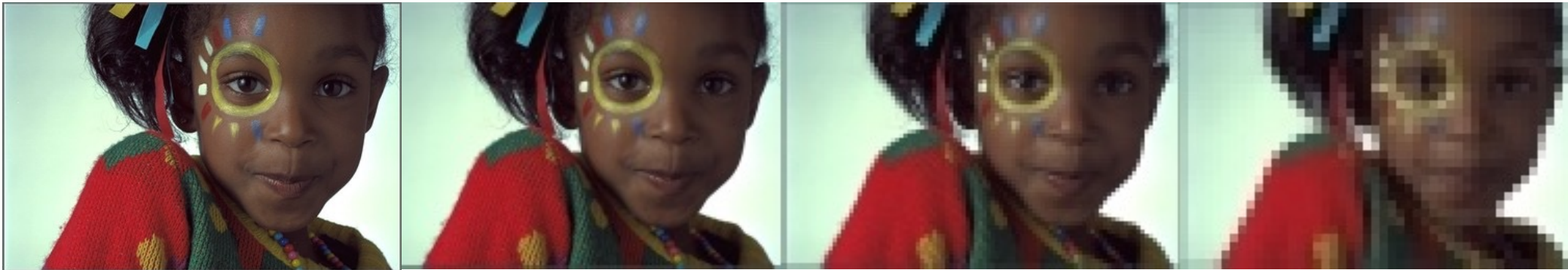
Résolution tonale (de tons de gris)

- Le plus petit changement discernable

Une image a donc une résolution spatiale de $M \times N$ pixel et une résolution de tons de gris de K bits ou de L niveaux ou tons

RÉSOLUTIONS SPATIALE ET TONALE

Échantillonnage : Résolution spatiale



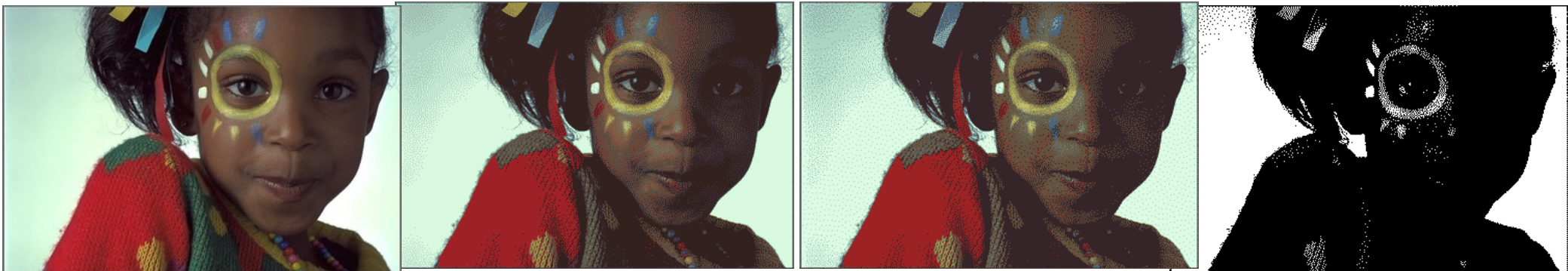
256 x 384

128 x 192

64 x 96

32 x 48

Quantification : résolution tonale



256 x 256 x 8 bits

256 x 256 x 7 bits

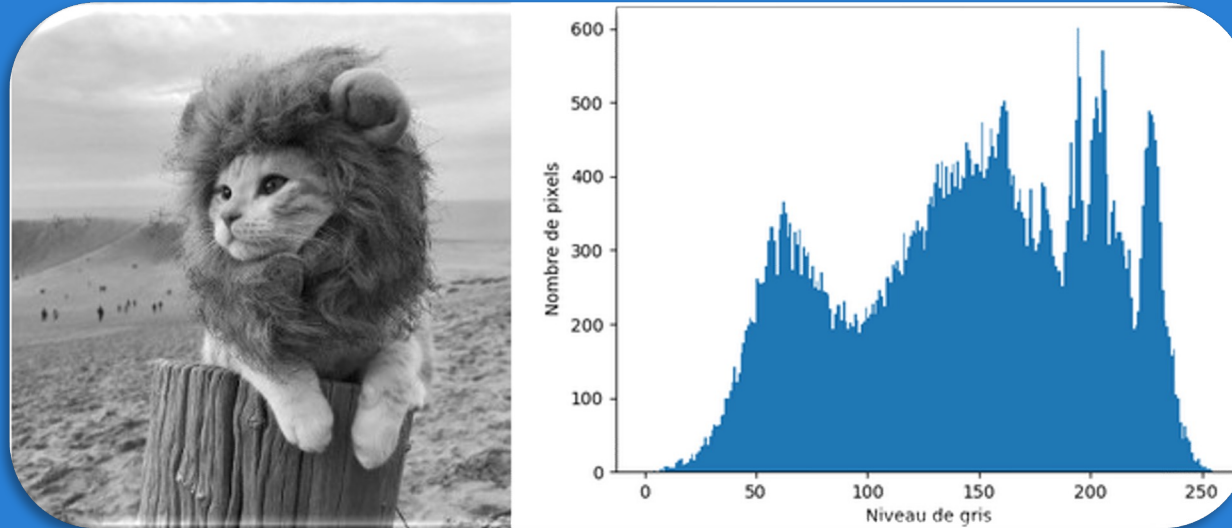
256 x 256 x 6 bits

256 x 256 x 1 bit²⁸



CHAPITRE 1

Traitement d'image



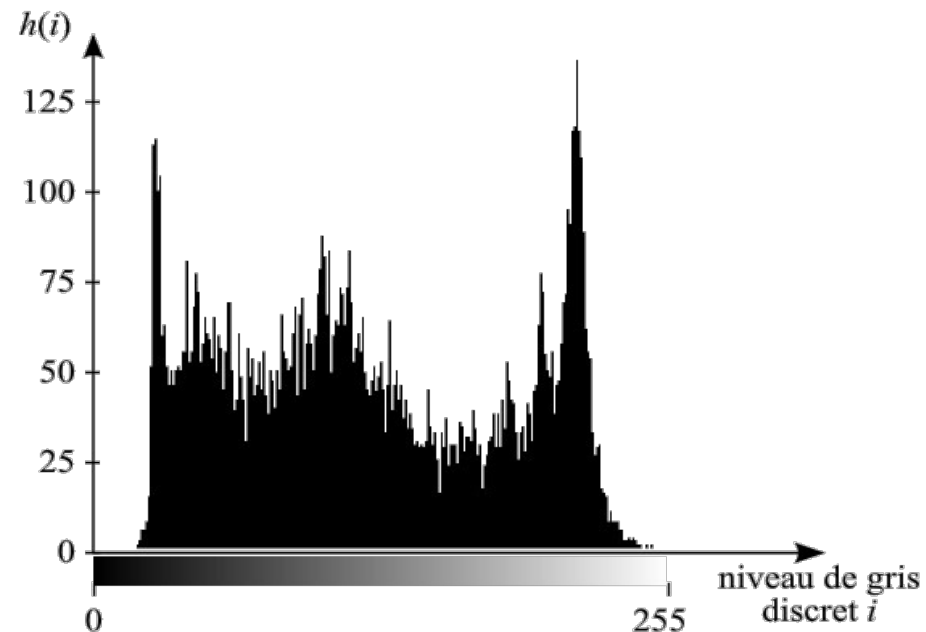
MANIPULATIONS D'HISTOGRAMMES

Chapitre 1: Traitement d'image

QU'EST CE QU'UN HISTOGRAMME?

L'histogramme d'une image numérique est une courbe statistique représentant la répartition de ses pixels selon leur intensité.

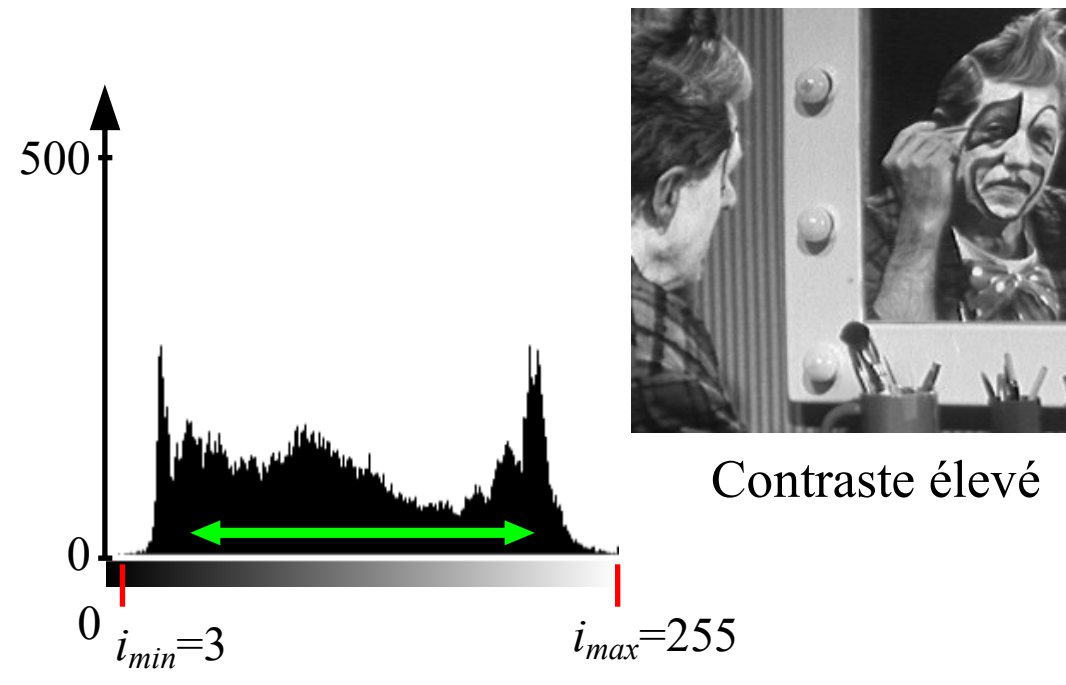
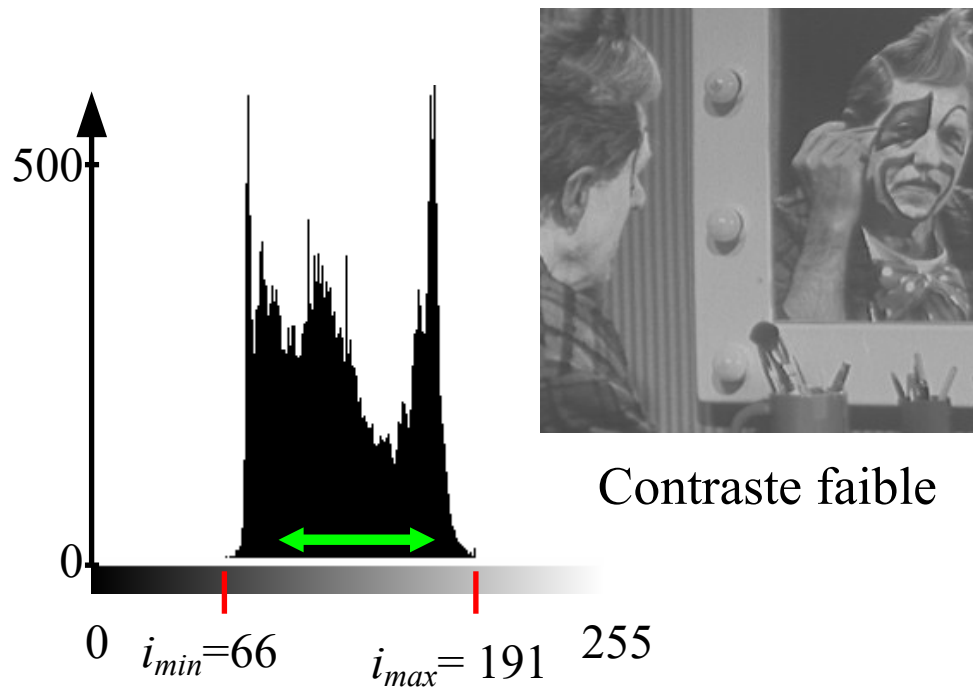
Pour une image en niveau de gris, il indique en abscisse le niveau de gris (entier entre 0 et 255) et en ordonnée, le nombre de pixels ayant cette valeur $h(i)$.



COMMENT INTERPRÉTER UN HISTOGRAMME?

dynamique de l'image : nombre de niveaux $[i_{\min}, i_{\max}]$ réellement présents.

contraste de l'image : peut être estimé par l'écart-type σ . Il est d'autant plus élevé que l'histogramme est « étalé » horizontalement.



HISTOGRAMMES NORMALISÉ ET CUMULÉ

Lorsque l'**histogramme est normalisé**, il indique en ordonnée la probabilité p_i de trouver un pixel de niveau de gris i dans l'image :

$$\forall i \in \{0, \dots, 255\}, p_i = \frac{\text{nombre de pixels d'intensité } i}{\text{nombre total de pixels}}$$

Ou
$$h_n(i) = \frac{h(i)}{N \times M}$$

L'intensité d'un pixel est alors vue comme une variable aléatoire discrète.

L'histogramme cumulé est donné par :

$$h_c(k) = \sum_{i \leq k} h(i)$$

HISTOGRAMME CUMULÉ NORMALISÉ

Un histogramme cumulé normalisé calcule le pourcentage de pixels ayant une valeur inférieure à un niveau de gris donné :

$$\forall k \in \{0, \dots, 255\}, h_c(k) = \sum_{i \leq k} h_n(i)$$

L'histogramme normalisé peut être interprété comme une densité de probabilité, et l'histogramme cumulé normalisé comme la fonction de répartition.

GÉNÉRER UN HISTOGRAMME AVEC PYTHON

La génération d'histogrammes se fait aisément en Python avec la fonction **hist** de **matplotlib.pyplot**, la manipulation de l'image est générée par le module **Pillow**, voir [support](#):

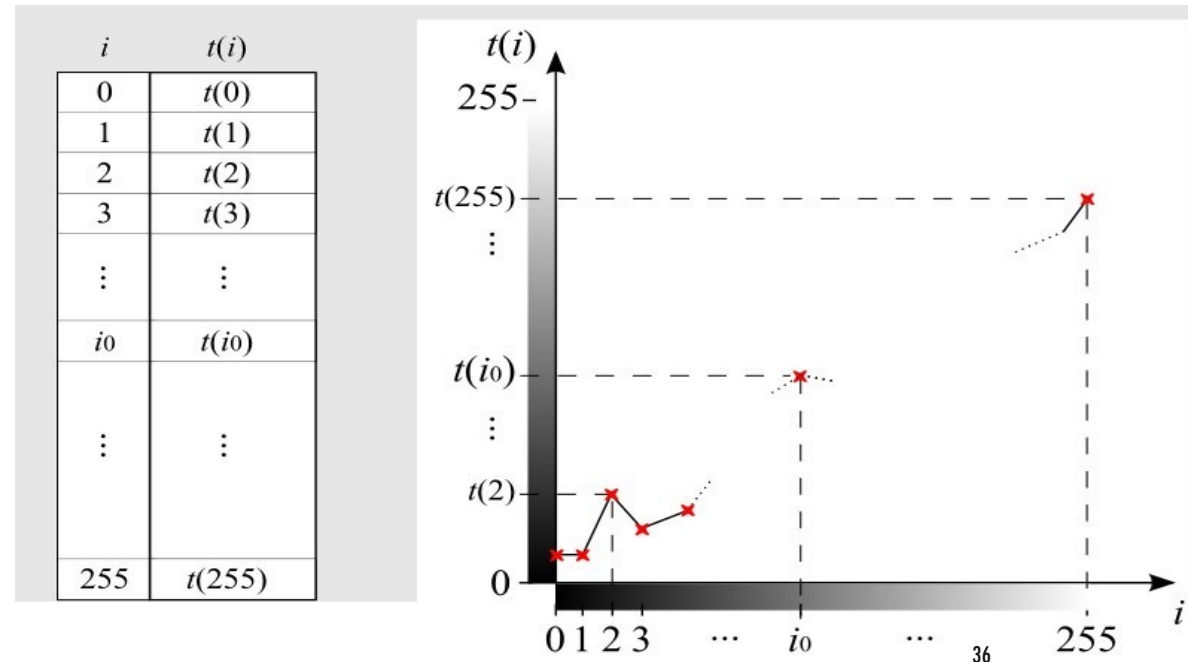
```
import matplotlib.pyplot as plt
import numpy as np
from PIL import Image
# Charger l'image comme matrice de pixels
img = np.array(Image.open('simba.png'))
# Générer et afficher l'histogramme
# Pour le normaliser : argument density=True dans plt.hist
# Pour avoir l'histogramme cumulé : argument cumulative=True
n, bins, patches = plt.hist(img.flatten(), bins=range(256))
plt.show()
```


TRANSFORMATION D'IMAGES

Une transformation t change un niveau de gris i dans l'image initiale I en un niveau de gris i' pour obtenir l'image résultat I' .

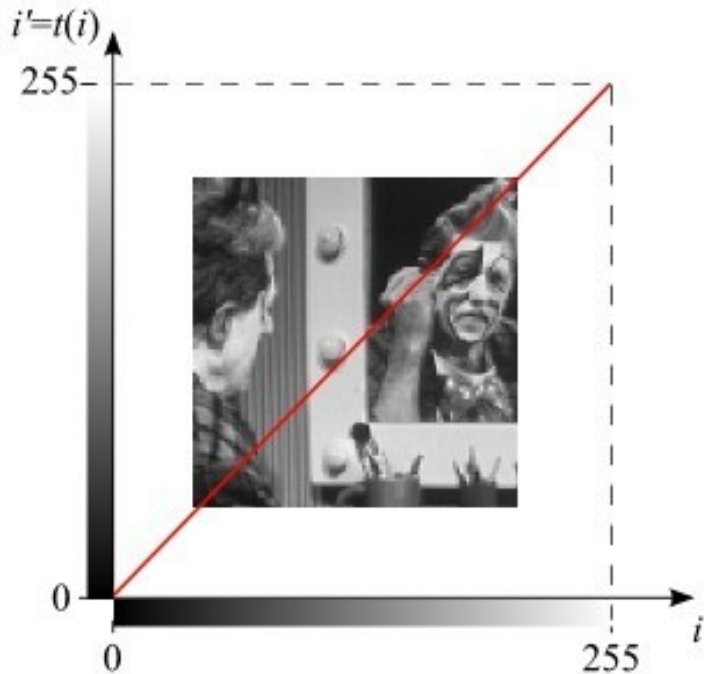
Elle est donc définie par une table de correspondance ou LUT (Look-Up Table) qui définit, pour chaque niveau de gris i , le nouveau niveau $i' = t(i)$.

Représentations de la LUT

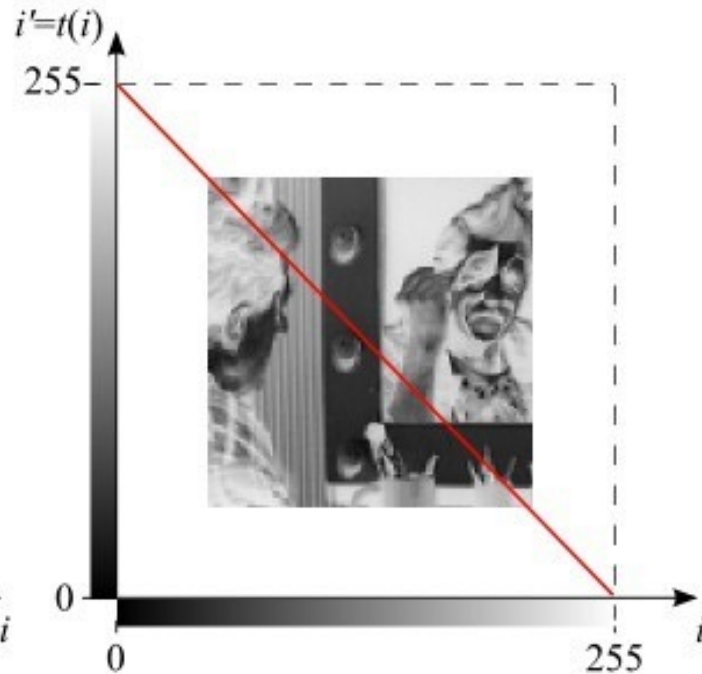


EXEMPLES DE TRANSFORMATIONS

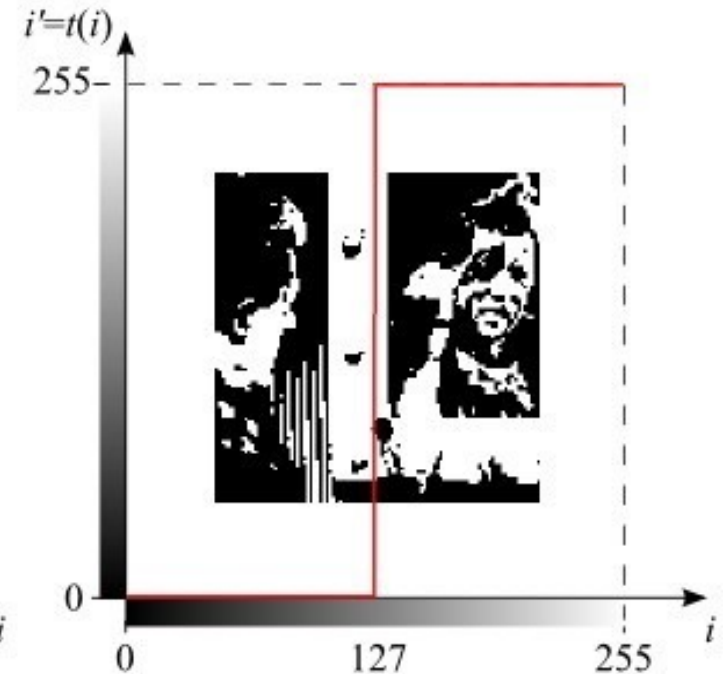
Identité $i'=i$



Négatif $i'=255-i$

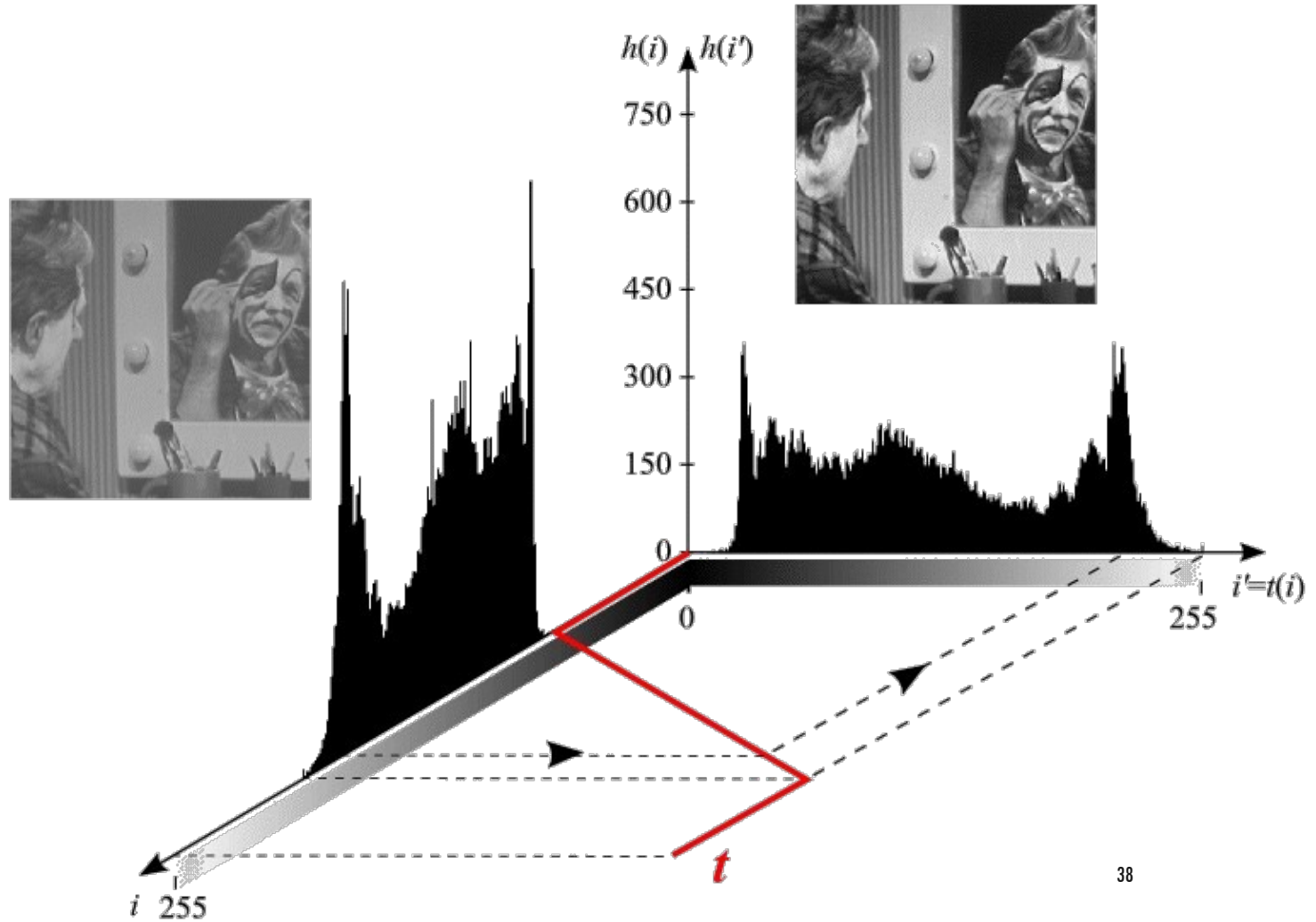


Seuillage $i' = \begin{cases} 0 & \text{si } i < 127 \\ 255 & \text{si } i \geq 127 \end{cases}$



ÉTIREMENT D'HISTOGRAMMES

Une première application consiste à corriger la luminosité, ou exposition, de l'image. L'image d'origine est trop claire, ou sur-exposée, se concentrent dans la partie droite de l'histogramme



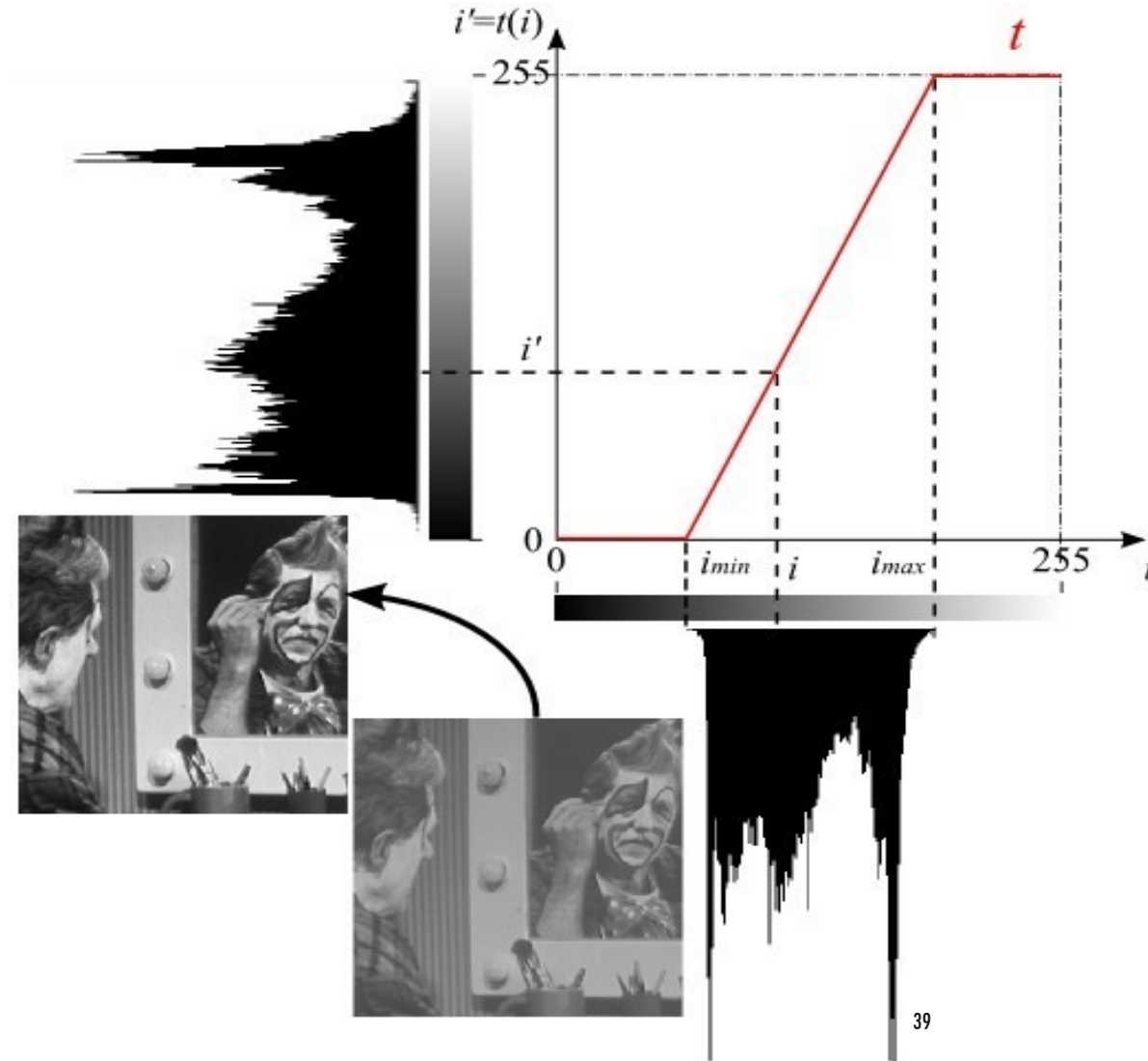
ÉTIREMENT D'HISTOGRAMMES

Pour corriger les défauts liés à l'exposition d'une image, il suffit simplement d'étirer son histogramme : l'objectif est d'étendre les valeurs des niveaux de gris de l'image mal exposée à tout l'intervalle disponible.

La valeur de chaque pixel est remplacée par le résultat de la formule ci-dessous.

$$I'(x, y) = \frac{255 \times (I(x, y) - I_{min})}{I_{max} - I_{min}}$$

où $I(x, y)$ et $I'(x, y)$ désignent les intensités du pixel dans l'image mal exposée et la nouvelle image.

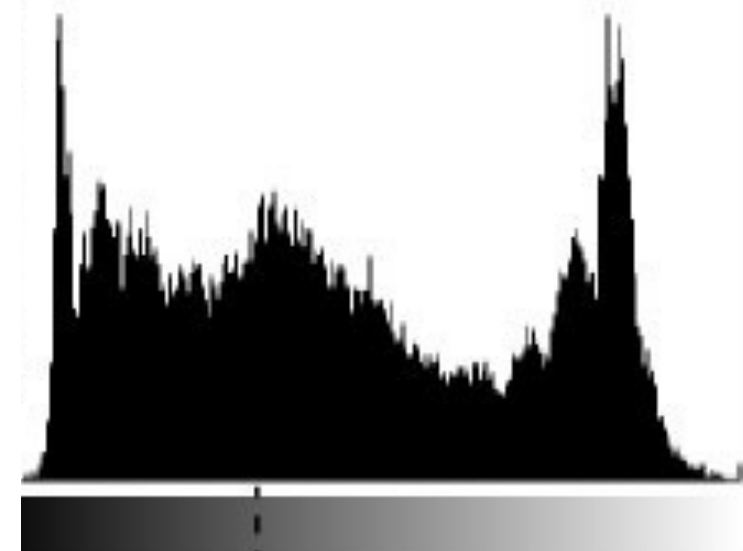


ÉTIREMENT D'HISTOGRAMMES EN PYTHON

L'étirement d'histogramme se fait avec la fonction `autocontrast` du module `ImageOps` de Pillow.

Les pixels se répartissent bien dans tout l'intervalle $[0, 255]$ et l'image présente une meilleure luminosité.

En guise de petit exercice, vous pouvez appliquer la transformation sur l'image trop sombre.



```
from PIL import Image, ImageOps
im1 = Image.open(r"cailloux.png").convert("L")
im2 = ImageOps.autocontrast(im1)
im1.show()
im2.show()
```

ÉGALISATION D'HISTOGRAMMES

La deuxième application courante concerne l'amélioration du contraste de l'image.

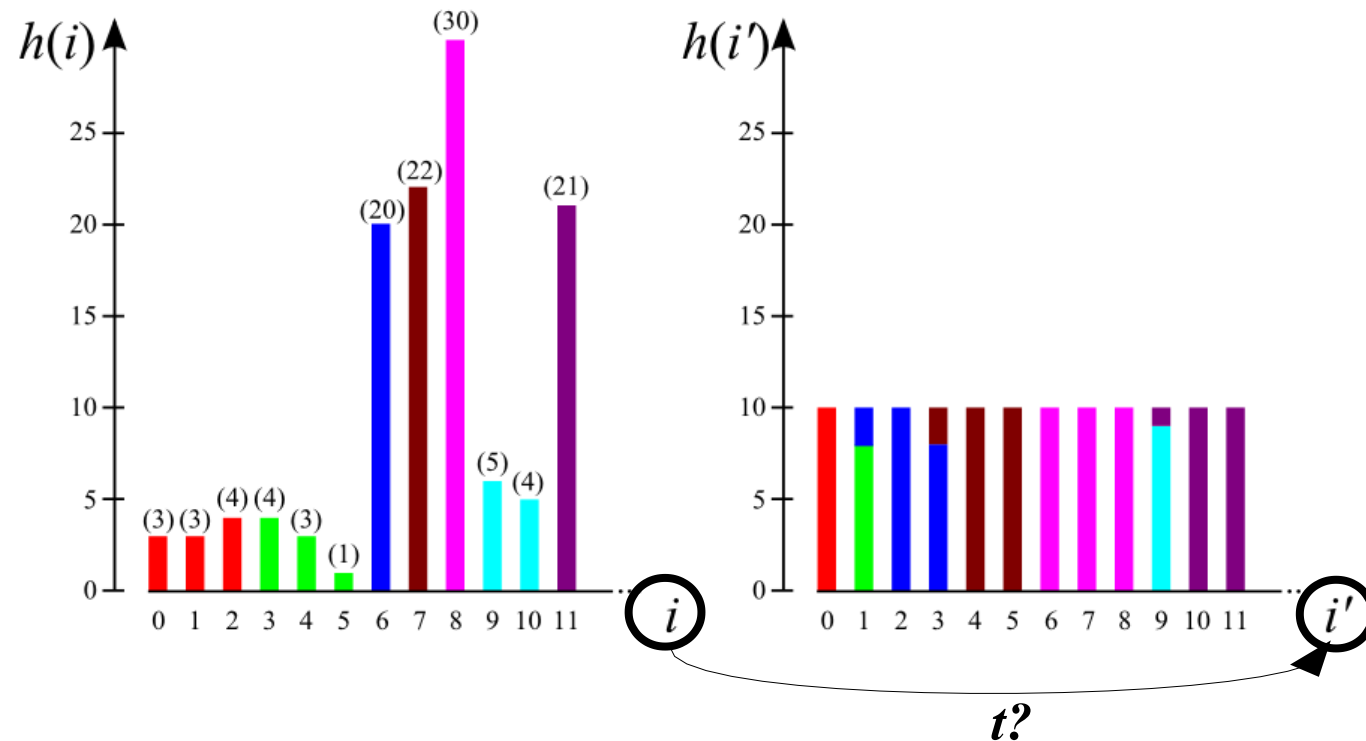
Le contraste caractérise la répartition de lumière dans une image : plus une image est contrastée, plus la différence de luminosité entre ses zones claires et sombres est importante.

En général, une image peu contrastée est terne, tandis qu'une image trop contrastée est visuellement "agressive".

Dans les deux cas, l'image manque de clarté car certains de ses détails seront peu, voire pas du tout, visibles

ÉGALISATION D'HISTOGRAMMES

On cherche une fonction de transformation qui permet d'harmoniser la distribution des niveaux de gris de l'image, de sorte que chaque niveau de l'histogramme contienne idéalement le même nombre de pixels. Concrètement, on essaye d'aplatir au maximum l'histogramme original.



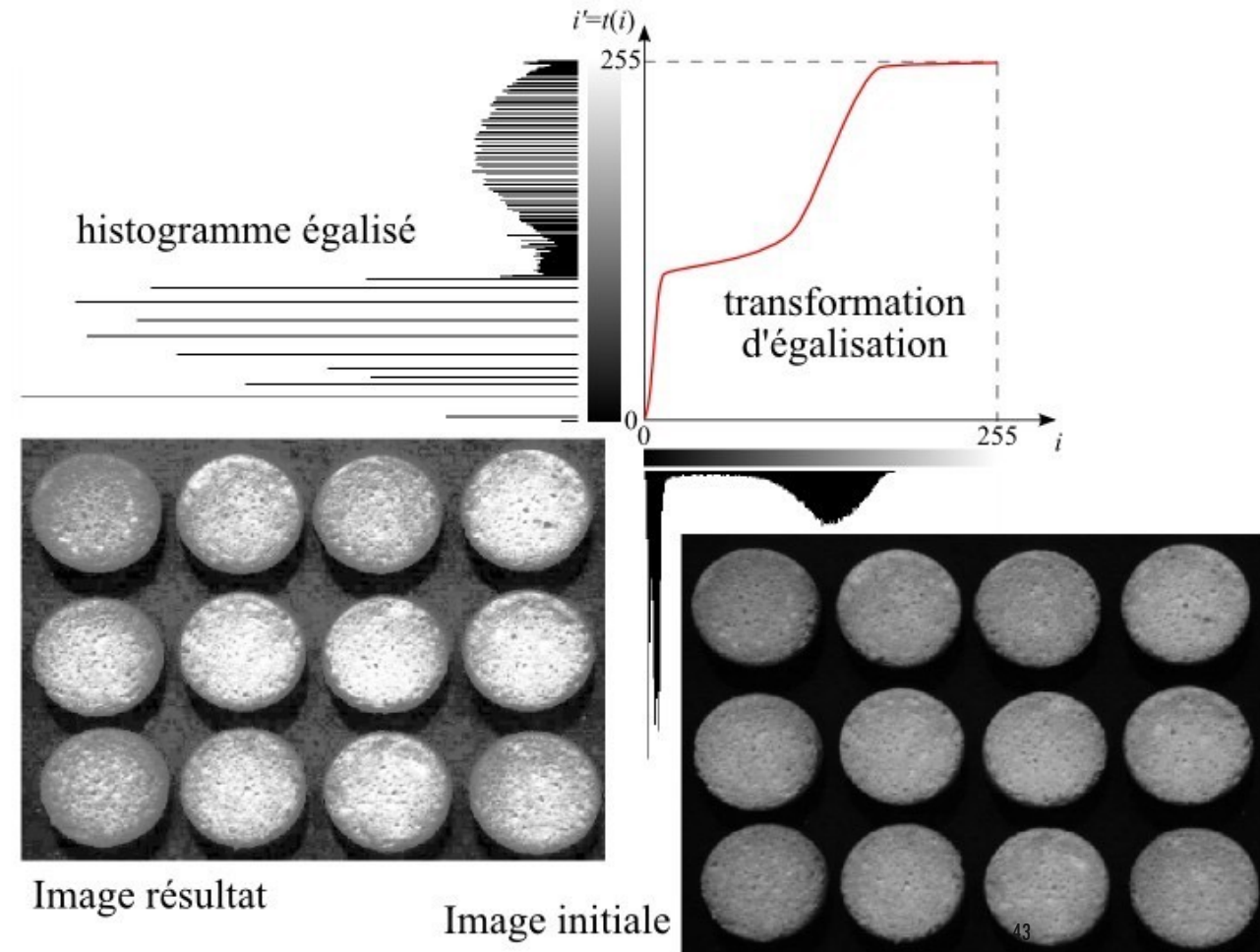
ÉGALISATION D'HISTOGRAMMES

La transformation $t : i \rightarrow i'$ cherchée est:

$$i' = \text{round}\left(255 \times \frac{h_c(i)}{N \times M}\right)$$

Avec,

$$h_c(k) = \sum_{i \leq k} h(i)$$



ÉGALISATION D'HISTOGRAMMES



Image
initiale

Histogrammes
initial et
initial cumulé

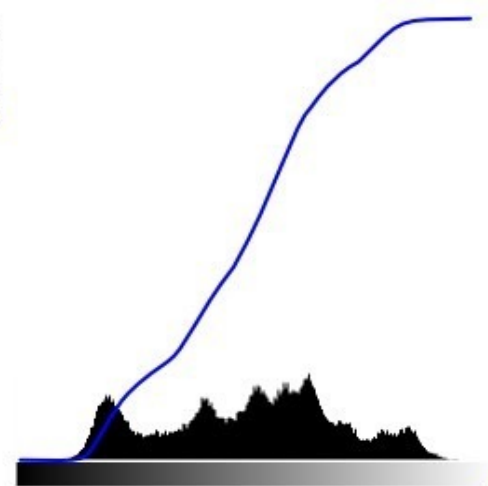
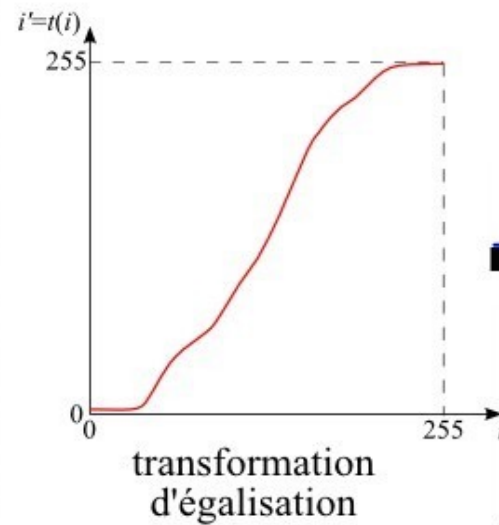
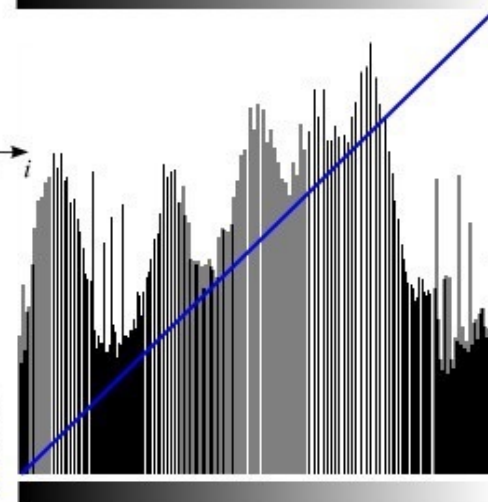


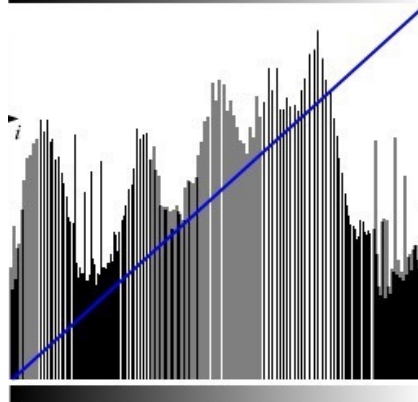
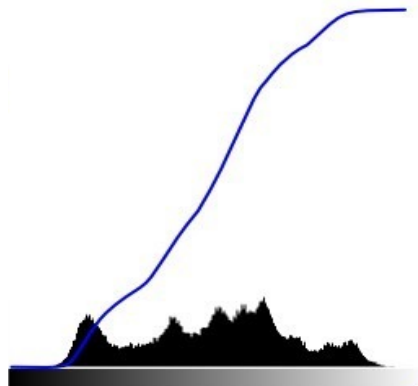
Image
résultat

Histogrammes
égalisé et
égalisé cumulé



ÉGALISATION D'HISTOGRAMMES AVEC PYTHON

L'égalisation d'histogramme correspond à la fonction `PIL.ImageOps.equalize`



```
from PIL import Image  
img = Image.open('cailloux.png')  
img2 = ImageOps.equalize(img)  
img2.show()
```

L'image est plus contrastée et son histogramme confirme que la distribution de ses niveaux de gris est plus uniforme.

TRANSFORMATIONS GÉOMÉTRIQUES

Chapitre 1: Traitement d'image

TRANSFORMATIONS GÉOMÉTRIQUES

Une autre catégorie de traitement ponctuel regroupe les transformations géométriques, qui modifient la position des pixels dans l'image. Une rotation, une translation ou un changement d'échelle (zoom) en sont des exemples typiques.

Une transformation géométrique est caractérisée par une matrice et un vecteur réels, respectivement notés T et V . Les nouvelles coordonnées (x', y') du pixel de position initiale (x, y) sont déterminées selon l'équation ci-dessous :

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = T \begin{pmatrix} x \\ y \end{pmatrix} + V$$

TRANSFORMATIONS GÉOMÉTRIQUES

Plus précisément, les rotations, translations et zooms s'expriment de la manière suivante :

- Translation de vecteur $(t_1 \ t_2)^T$: $\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}$
- Zoom de coefficients φ_1 et φ_2 : $\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \varphi_1 & 0 \\ 0 & \varphi_2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$
- Rotation d'angle θ : $\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$

TRANSFORMATIONS GÉOMÉTRIQUES

Le module ***PIL.Image*** contient les méthodes permettant d'appliquer les transformations géométriques.

Leur paramètre d'entrée ***resample*** donne la possibilité de choisir le mode d'interpolation, et ***expand*** est un paramètre Boolean qui c'est sa valeur est true cela permet d'agrandir l'image au cas où des pixels se trouvent en dehors des bornes après transformation.

TRANSFORMATIONS GÉOMÉTRIQUES

Comparons les effets des interpolations au plus proche voisin et bilinéaire avec une rotation (`PIL.Image.rotate`) :



Image après rotation avec une interpolation au plus proche voisin (à gauche) et une interpolation bilinéaire (à droite)

TRANSFORMATIONS GÉOMÉTRIQUES AVEC PYTHON

```
from PIL import Image  
im=Image.open("simba.png")  
# Rotate the image by 60 degrees counter clockwise  
theta = 60  
# Angle is in degrees counter clockwise  
im_rotated = im.rotate(angle=theta,resample=Image.BILINEAR, expand=1,  
center=None, translate=None, fillcolor=None)  
im_rotated.show()
```