UE Introduction au Traitement d'Images et à la Vision par Ordinateur

Isabelle Debled-Rennesson
Et Gilles Simon

Introduction au Traitement d'Images

Isabelle Debled-Rennesson
debled@loria.fr

Contenu

- 6 cours/TD/TP de 2h
- Points abordés :
  - Codage des images
  - Histogrammes
  - Transformations géométriques
  - Filtrages
  - Opérateurs morphologiques
  - Détection de contours
  - Segmentation

Vu en semestre 1 du M1

Introduction aux modèles de perception et de raisonnement

- Transformées
- Numérisation – quantification
- La couleur et les espaces de couleur

Modalités d'évaluation

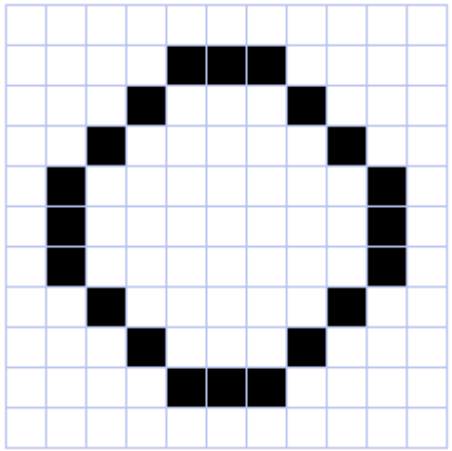
- Examen de 2 h
  - 1h sur traitement d'images
  - 1h sur vision
- 1 TP noté ou un travail à réaliser

Codage des Images

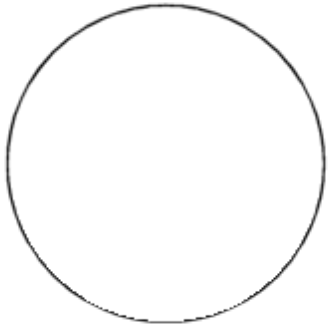
Qu'est-ce qu'une image numérique ?

Deux formes courantes :

- **Images bitmap**: tableau à deux dimensions contenant des valeurs



- **Images vectorielles**: description géométrique de l'image



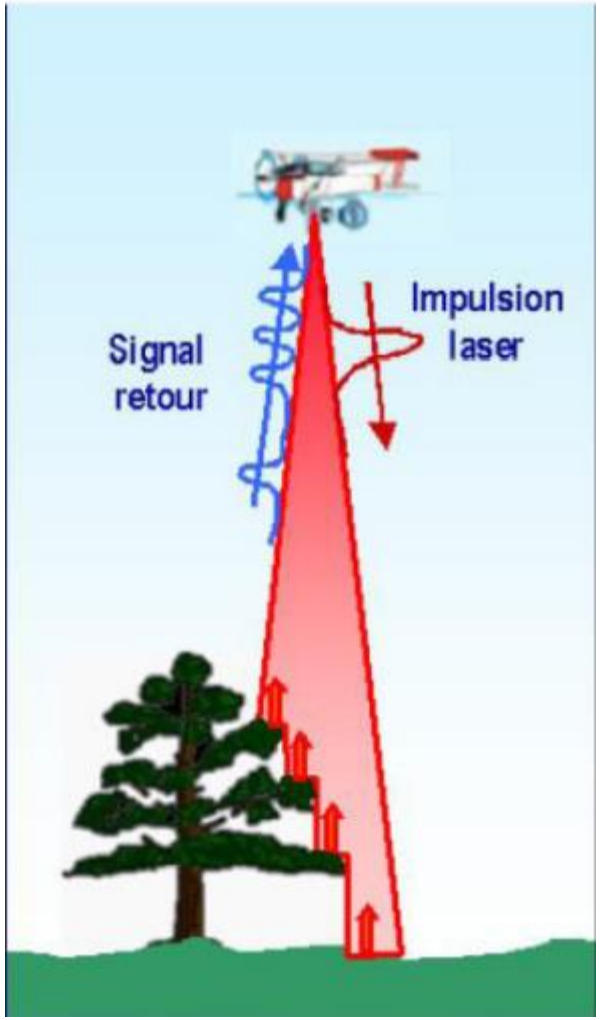
Sources des images numériques - 1

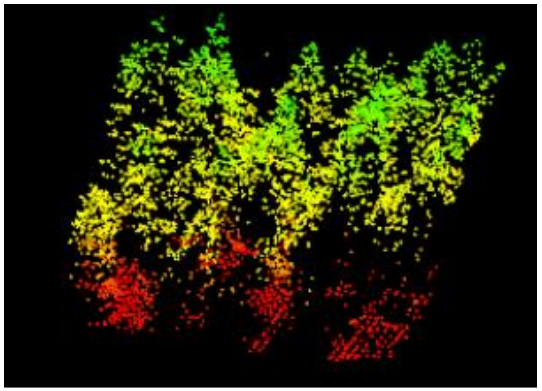
- Capteur CCD (Charge-Coupled Device ou dispositif à transfert de charge)
  - Composant électronique constitué de capteurs photosensibles – le rayonnement perçu est converti en un signal électrique analogique. Ce signal est ensuite numérisé pour obtenir une image numérique
    - Appareil photo numérique
    - Camescope numérique
    - Téléphone portable
    - Scanner à plat
    - …

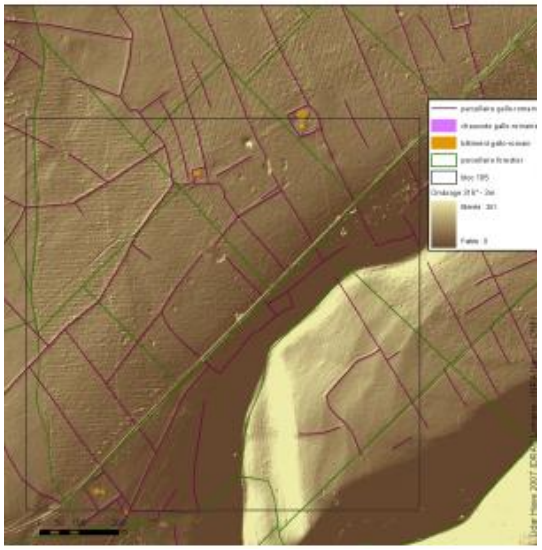
Sources des images numériques - 2

- Télémétrie
  - Radar (radio)
  - Lidar (laser)
  - Sonar (son)
  - …

Procédé technique permettant de calculer ou de mesurer la distance d'un objet lointain par utilisation d'éléments optiques, acoustiques ou radioélectriques

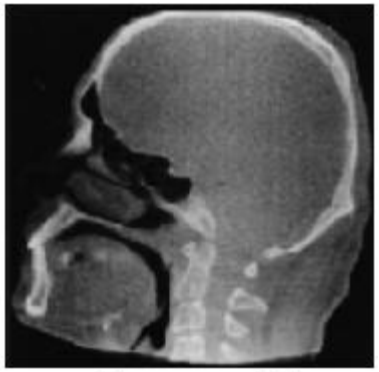




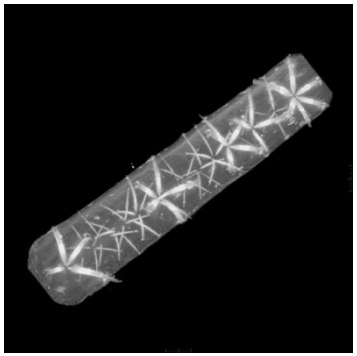


Sources des images numériques - 3

- Imagerie médicale
  - Scanner X
  - Imagerie par Résonnance Magnétique (IRM)
  - Echographie
  - …



Scanner X

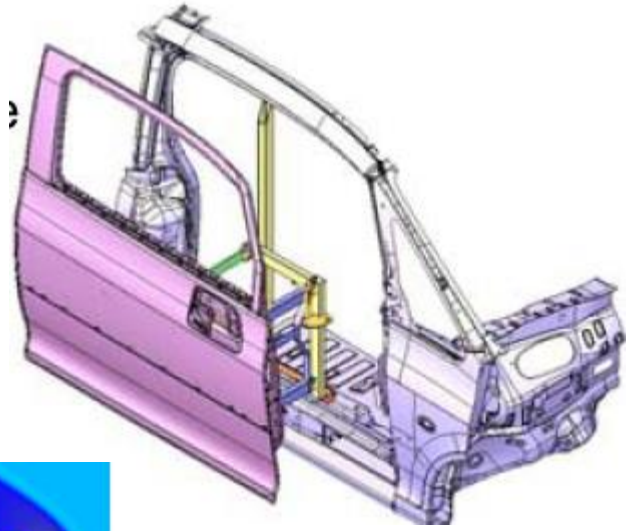


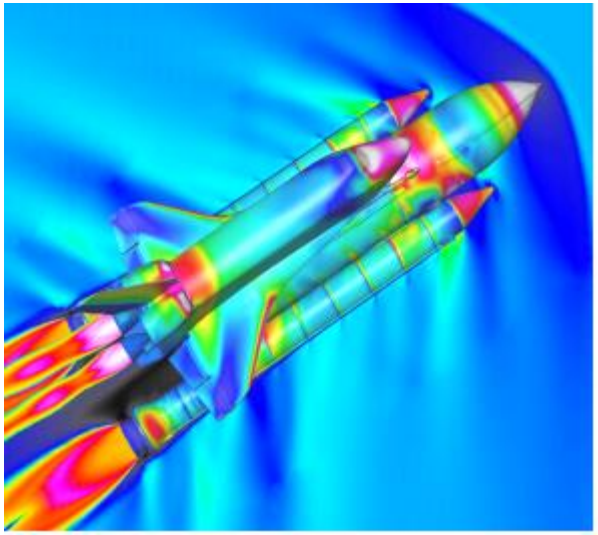
IRM

Sources : Sébastien Thon

Sources des images numériques - 4

- Le calcul : images de synthèse
  - Conception Assistée par Ordinateurs
  - Visualisation scientifique
  - Simulation
  - Réalité virtuelle
  - Cinéma
  - Jeu vidéo
  - …

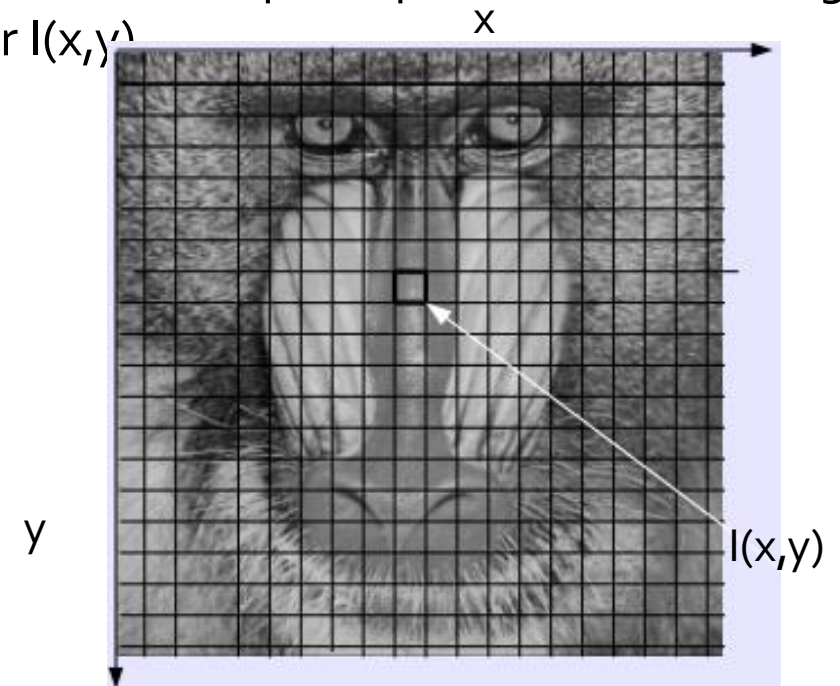




Sources : Sébastien Thon

Echantillonnage des images

- Procédé de **discrétisation** des images consistant à transformer une information analogique en une information digitale
  - **Numérisation = Echantillonnage** et **quantification** : associe à chaque zone rectangulaire (définie par le pas d'échantillonnage), nommée **pixel**, une unique valeur I(x,y)



Sous-echantillonnage des images

- On parle de ***sous-échantillonnage*** lorsque l'image est déjà discrétisée et que le nombre d'échantillons est diminué

Sources : Sébastien Thon

Quantification

- La ***quantification*** désigne le nombre de valeurs que peut prendre chaque pixel

- Exemple:
  4 quantifications différentes de la même image
  - 32, 16, 8, 4 niveaux de gris.

Images numériques - vocabulaire

- Ne pas confondre la *définition* d'une image et sa *résolution*
  - ***Définition d'une image***
    - Dimension de l'image en pixels – par ex, une image de 800 pixels de largeur et de 600 pixels de hauteur a une définition de **800 pixels par 600**, notée **800x600**
  - ***Résolution d'une image***
    - Nombre de pixels par unité de surface, exprimé en *points par pouce* ou **PPP** (en anglais **DPI**: *Dots Per Inch*), un pouce=2,54cm
    - Exprime le lien entre le nombre de pixels d'une image et sa taille réelle sur un support physique

Résolution - Exemple

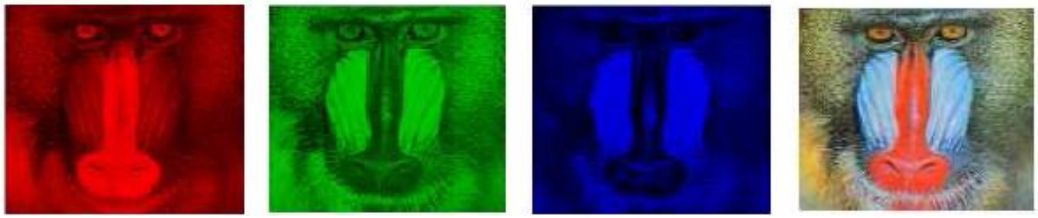
- Quelle est la définition d'une image de 15cm de large sur 10 cm de haut avec une résolution de 300 dpi ?

Image bitmap ou matricielle (1)

- Une image bitmap est composée d'une matrice de nombres (L*H) représentant (après échantillonnage et quantification) un signal continu, elle peut donc être vue comme une fonction :

$$I : \{0,1,\ldots,L-1\}*\{0,1,\ldots,H-1\} \rightarrow \Omega$$
$$(x,y) \rightarrow I(x,y)$$

  - Codage en niveaux de gris :
    - Codage 8 bits (le plus classique) $\Omega = \{0,1, \ldots, 255\}$, du plus foncé au plus clair
  - Codage couleur :
    - Codage 24 bits $\Omega = \{0,1, \ldots, 255\}*\{0,1, \ldots, 255\}*\{0,1, \ldots, 255\}$



Exemple d'image bitmap

y =	x = 58	59	60	61	62	63	64	65	66	67					
41	210	209	204	202	197	247	143	71	64	80					
42	206	196	203	197	195	210	207	56	63	58					
43	201	207	192	201	198	213	156	69	65	57					
44	216	206	211	193	202	207	208	57	69	60					
45	221	206	211	194	196	197	220	56	63	60					
46	209	214	224	199	194	193	204	173	64	60					
47	204	212	213	208	191	190	191	214	60	62					
48	214	215	215	207	208	180	172	188	69	72					
49	209	205	214	205	204	196	187	196	86	62					
50	208	209	205	203	202	186	174	185	149	71	63	55	55	45	56
51	207	210	211	199	217	194	183	177	209	90	62	64	52	93	52
52	208	205	209	209	197	194	183	187	187	239	58	68	61	51	56
53	204	206	203	209	195	203	188	185	183	221	75	61	58	60	60
54	200	203	199	236	188	197	183	190	183	196	122	63	58	64	66
55	205	210	202	203	199	197	196	181	173	186	105	62	57	64	63

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

Taille d'une image

- La taille d'une image dépend de sa discrétisation et de sa quantification

$$\text{Taille mémoire} = \underbrace{\text{largeur} \times \text{hauteur}}_{\textit{discrétisation}} \times \underbrace{\text{nb octets par pixel}}_{\textit{quantification}}$$

  - Ex : image de 800x600 pixels avec 24 bits par pixel,
    taille mémoire = 800x600x24
    = 11520000 bits
    = 1 440 000 octets
    = 1,37 Mio

Taille d'une image

- La taille d'une image dépend de sa discrétisation et de sa quantification

$$\text{Taille mémoire} = \underbrace{\text{largeur} \times \text{hauteur}}_{\textit{discrétisation}} \times \underbrace{\text{nb octets par pixel}}_{\textit{quantification}}$$

Multiples de l'octet : préfixes SI et mésusages

Nom	Symbole	Valeur	Mésusage[2]
kilooctet	ko	10^3	2^{10}
mégaoctet	Mo	10^6	2^{20}
gigaoctet	Go	10^9	2^{30}
téraoctet	To	10^{12}	2^{40}
pétaoctet	Po	10^{15}	
exaoctet	Eo	10^{18}	
zettaoctet	Zo	10^{21}	
yottaoctet	Yo	10^{24}	

Multiples de l'octet : préfixes binaires

Nom	Symbole	Valeur
kibioctet	Kio	2^{10}
mébioctet	Mio	2^{20}
gibioctet	Gio	2^{30}
tébioctet	Tio	2^{40}
pébioctet	Pio	2^{50}
exbioctet	Eio	2^{60}
zébioctet	Zio	2^{70}
yobioctet	Yio	2^{80}

pixels avec 24 bits par pixel,

800x600x24

11520000 bits

1 440 000 octets

1,37 Mio

Sources http://fr.wikipedia.org/wiki/Octet et http://fr.wikipedia.org/wiki/Pr%C3%A9fixe_binaire

Images bitmap (2)

- Codage direct:
  - I(x,y) représente une valeur d'intensité lumineuse
  - I(x,y) peut être un scalaire (images à niveaux de gris) ou un vecteur(images couleurs)

- Exemple:
  - images à niveaux de gris, codage sur 8 bits: I(x,y) varie entre 0 et 255
  - images couleurs, codage sur 3x8= 24 bits: I(x,y) varie entre (0,0,0) et (255,255,255)

Images bitmap (3)

- Codage indirect :
  - I(x,y) ne représente plus *directement* une valeur d'intensité lumineuse
  - I(x,y) est une valeur (index) qui va correspondre à une couleur dans une table : ***la palette***

  - Avantage :
    - si l'image contient peu de couleurs différentes, gain de place
  - Inconvénient :
    - accès plus lent
    - pour que cela soit intéressant, nombre de couleurs limité

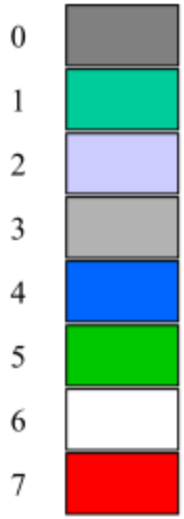
Images bitmap (4)

- Exemple de codage indirect avec une palette

Image

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

Représentation en mémoire de l'image



Palette

Les fichiers des images bitmap

- Un très grand nombre de formats de fichiers possibles
  - JPEG, TIFF, PNG, GIF, BMP, …
- Forme générale d'un fichier image
  - En tête:
    - "Magic Number" pour reconnaître le format
    - largeur, hauteur
    - quantification: noir et blanc, niveaux de gris, couleur
    - mode de compression (éventuellement)
    - présence de palette
    - autres informations …
  - Corps de l'image:
    - suite d'octets représentant les pixels, image éventuellement compressée

Quelques formats des fichiers contenant des images bitmap

- Le format Portable *Bit/Gray/Pixel* Map : PBM, PGM, PPM
  - Format sans compression
  - Peu utilisé car non compressé mais très simple à utiliser

Le format PGM

- En tête en ASCII:
  - “P2”: ASCII ou “P5”: binaire
  - Un caractère d’espacement (TAB, CR, LF, espace)
  - lignes de commentaires optionnelles commençant par “#”
  - “X Y” : largeur et hauteur de l'image écrit sous forme de texte (ASC II) suivi d’un caractère d’espacement
  - “MAX”: niveau de gris maximal de l'image (en général 255 (8 bits)) suivi d’un caractère d’espacement

- Corps de l'image:
  - Suite de caractères (si ascii) représentant le pixel suivi d’un caractère d’espacement
  - Suite d'octets (si binaire)
    chaque octet donne la valeur du niveau de gris du pixel

Le format PGM

- Un exemple

```
P2
19  7
11
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  3  3  3  0  0  0  7  7  7  7  0  0 11  0  0  0 11  0
0  3  0  3  0  0  0  7  0  0  0  0  0 11 11  0 11 11  0
0  3  3  3  0  0  0  7  0  7  7  0  0 11  0 11  0 11  0
0  3  0  0  0  0  0  7  0  0  7  0  0 11  0  0  0 11  0
0  3  0  0  0  0  0  7  7  7  7  0  0 11  0  0  0 11  0
0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

Le format PGM

- Un exemple

P2

19 7

11

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 3 3 3 0 0 0 7 7 7 7 0 0 11 0 0 0 11 0

0 3 0 3 0 0 0 7 0 0

0 3 3 3 0 0 0 7 0 7

0 3 0 0 0 0 0 7 0 0

0 3 0 0 0 0 0 7 7 7

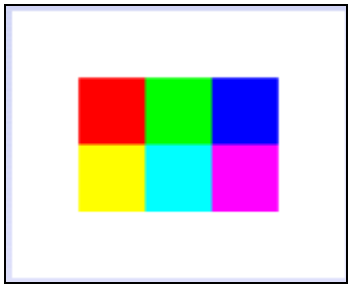
0 0 0 0 0 0 0 0 0 0

+ Copier/coller ces caractères dans un fichier texte nommé *ficPGM.pgm* puis l'ouvrir avec le logiciel gimp

+ Modifier un ou plusieurs des pixels de l'image, enregistrer les modifications (en ascii) dans gimp puis ouvrir le fichier dans un éditeur

Le Format PPM

- Concerne les images couleurs
- Même principe que le format PGM mais avec P3 en première ligne si en ascii et P6 si en binaire.
- Dans le tableau chaque pixel est défini par 3 nombres représentant les valeurs R, G et B.
- Exemple :



```
P3
5  4
15
15 15 15   15 15 15   15 15 15   15 15 15   15 15 15
15 15 15   15  0  0    0 15  0    0  0  15    15 15 15
15 15 15   15 15  0    0 15 15    15  0 15    15 15 15
15 15 15   15 15 15   15 15 15   15 15 15   15 15 15
```

Des logiciels libres

- Des logiciels de traitement d'images libres
  - Gimp
    - Documentation : http://docs.gimp.org/fr/index.html
  - ImageJ
    - Documentation: http://ufe.obspm.fr/offreFP/miniMan-IJ.pdf

Quelques formats avec compression

- Façon plus astucieuse de stocker les données
- Compression sans pertes : TIFF, GIF, PNG, BMP,…
  - Compression RLE (Run Length Encoding) – basée sur la répétition d'éléments consécutifs - Ex : si blanc blanc blanc blanc, on stockera plutot 4xblanc
    - Utilisée pour les images bitmap (par ex .bmp)
  - Compression LZW (Lempel – Ziv – Welch) – repérage des séquences répétées en construisant un dictionnaire de séquences – codage des séquences dans l'image par leurs indices dans le dictionnaire
    - Utilisée pour les formats GIF et TIFF
- Compression avec pertes : JPEG, JPEG2000, …
  - Compression JPEG (Joint Photographic Expert Group)
    - Permet de réduire la taille jusqu'à 1/25 ieme de l'original sans perte notable de qualité – utilisation de la transformée en cosinus discrète pour éliminer certaines composantes de hautes fréquences non visibles par l'oeil.

Images vectorielles

- Constituées de primitives géométriques simples
  - segments de droites, courbes, cercles… etc.
- Chaque primitive géométrique est décrite par une formule mathématique.
- Visualisation indirecte :

Image vectorielle :
• Primitive Cercle : r=3, c=(0,0)
• Segment AB
• A=(3,2) B=(5,7)

Interprétation :
• Définition de la résolution de l'image résultante
• Interprétation des primitives

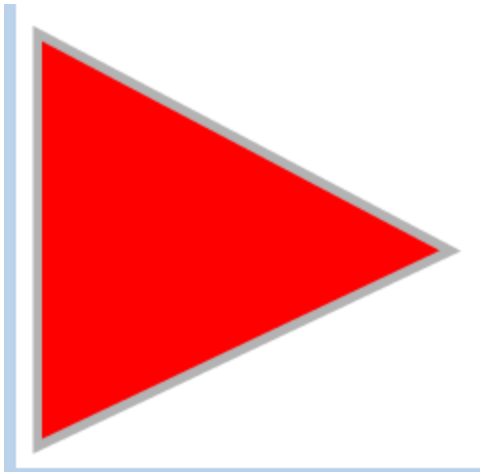
Image bitmap résultante :
• Visualisation
• Impression
• Sauvegarde

Images vectorielles

- Avantages :
  - Meilleur qualité d'impression/visualisation
  - Adaptées au stockage des images composées de formes géométriques
  - Prend moins de place qu'une image bitmap
- Différents types de formats de fichiers vectoriels :
  - EPS : Encapsuled PostScript.
  - SVG : Scalable Vector Graphics
  - Adobe pdf
  - Adobe illustrator AI
  - …

Un exemple de fichier postscript

```
%!PS-Adobe-2.0
%%Title: file c2.ps
%%CreationDate: May 16 13:43:52 1995
%%Origin: [eg: 0 0]
%%BoundingBox: 0 0 210 210
%%EndComments
%% trace de 2 segment d'epaisseur 4 unites
%
/cm {72 mul 2.54 div} def
%   epaisseur de trait
0.15 cm  setlinewidth
newpath
10 10 moveto
200 100 lineto
10 200 lineto
closepath
gsave
1  0 0 setrgbcolor
fill
grestore
0.7  setgray
stroke
showpage
```



Langage postscript :

- Syntaxe préfixée (RPN= notation polonaise inversée)
- Les arguments d'une fonction sont placés devant son nom

Inconvénients des images vectorielles

- Peuvent difficilement stocker des images complexes comme des photographies
- L'affichage d'une image vectorielle peut prendre plus de temps que l'affichage d'une image bitmap de complexité égale

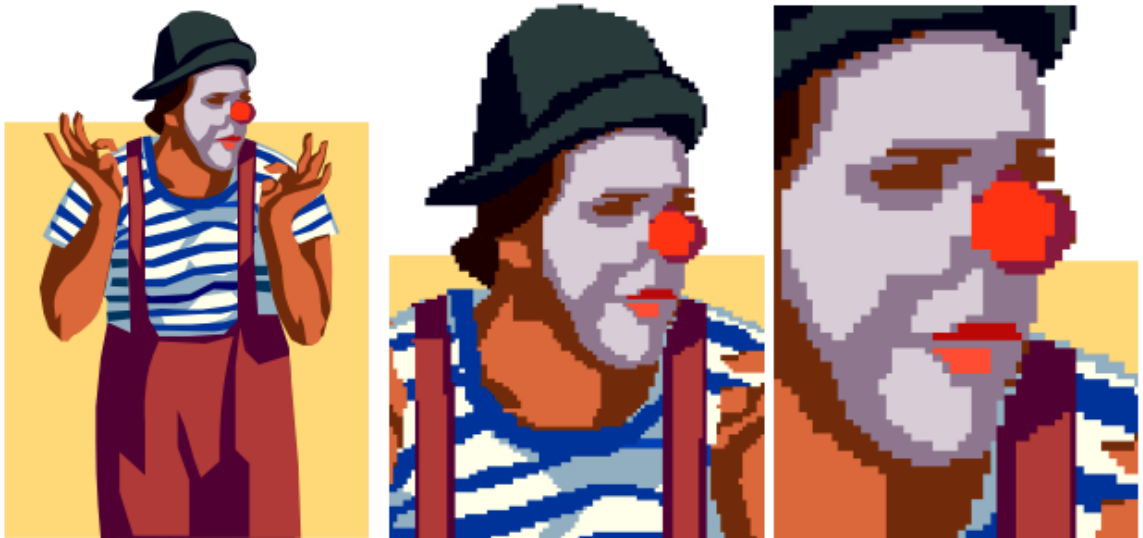
Comparaison vectoriel/bitmap

Vectoriel

Fichier : 10 Ko

Bitmap

Fichier : 74 Ko



Sources : Sébastien Thon

Quelques éléments de traitement des Images

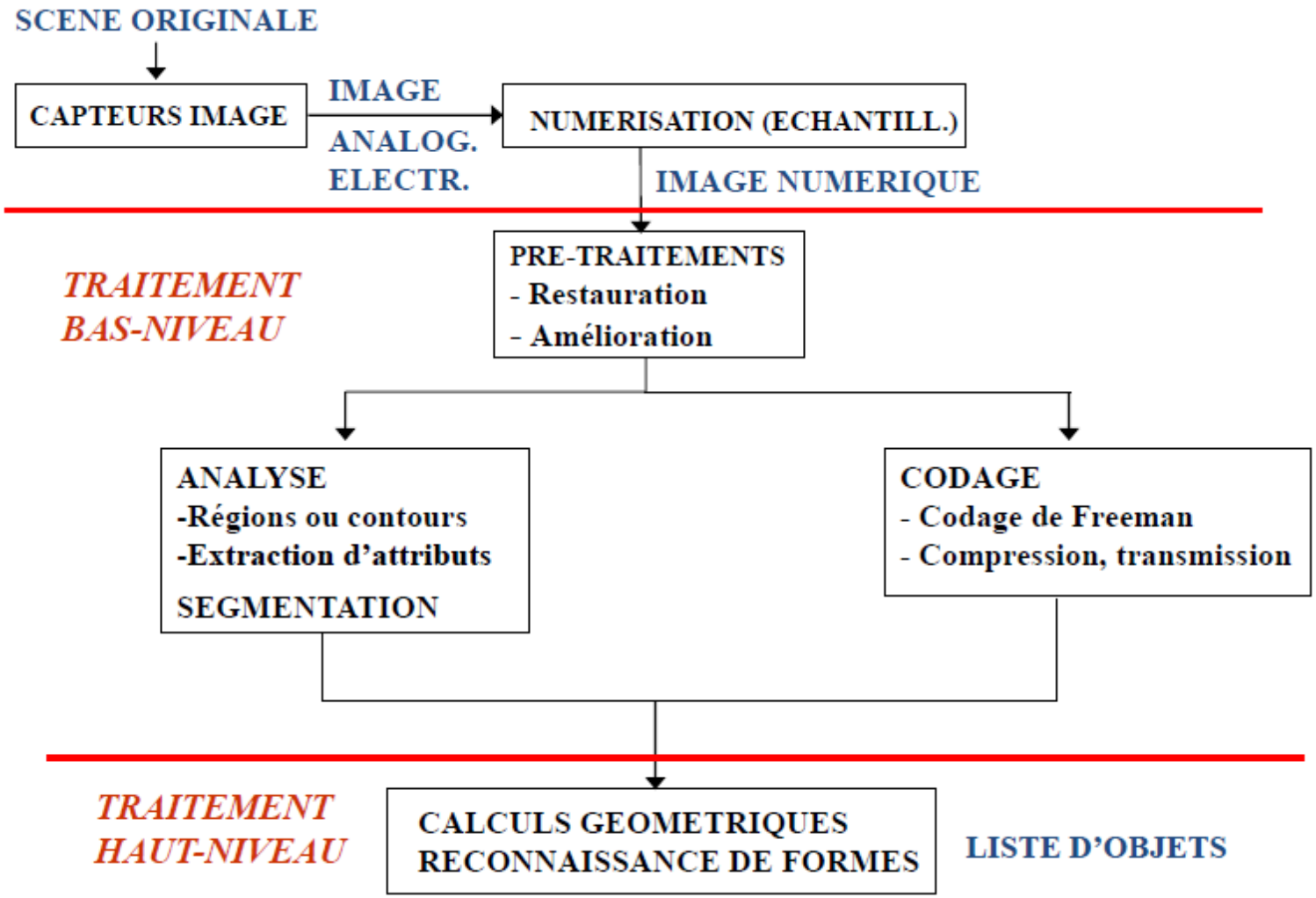
Introduction

Objectifs du traitement d'images

- Améliorer l'aspect visuel et esthétique des images.
  - Ex: restauration de films anciens, etc…
- Faciliter l'interprétation des images.
  - Ex: diagnostic médical, mesures de caractéristiques, aide à la décision, etc…

Traitement d'images

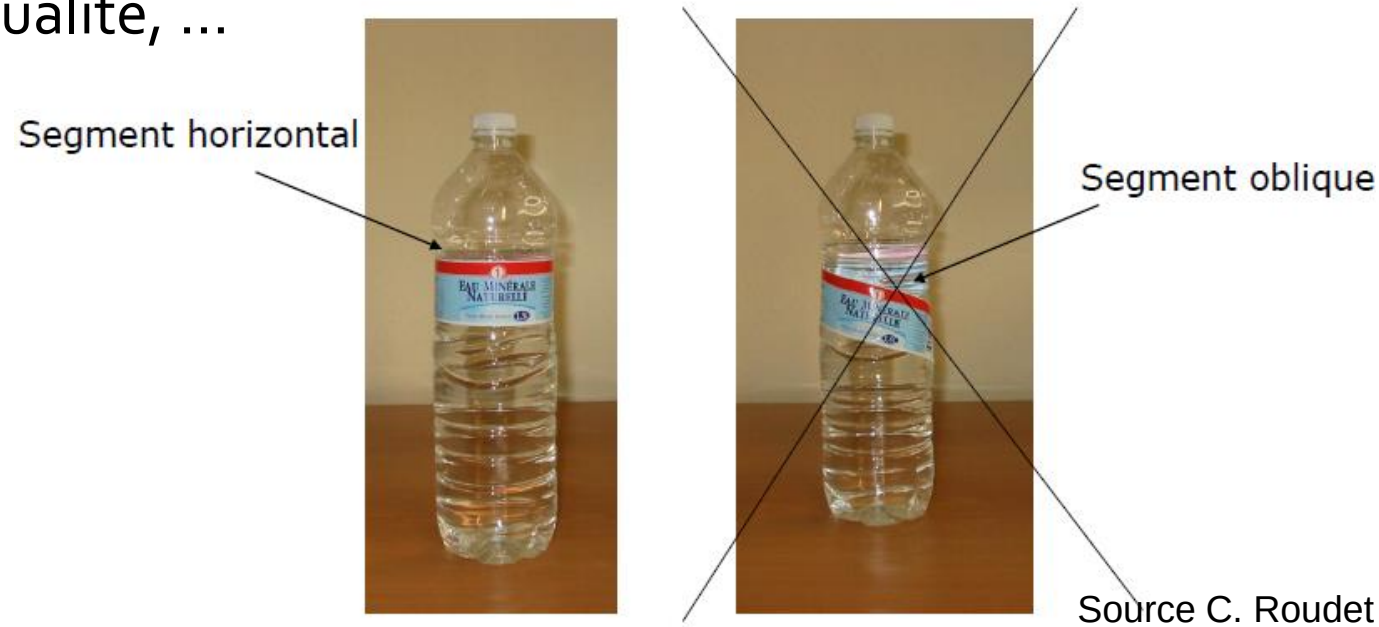
- Différents traitements



Source C. Roudet

De nombreux domaines d'application

- Imagerie médicale
  - Aide à la détection d'anomalies dans les images
    - Détection de maladies
- Contrôle industriel
  - Contrôle qualité, …



Source C. Roudet

De nombreux domaines d'application

- Biométrie
  - Reconnaissance de visage, d'empreintes digitales , de l'iris, ..
    - Contrôle d'accès, vérification d'identité, …
- Vidéosurveillance
  - Surveillance de lieux publics ou privé
    - Temps réel
- Télédétection
  - Imagerie aérienne et spatiale
    - Cartographie (recherche automatique des réseaux urbains, des forêts, …)
    - Surveillance

Source C. Roudet

De nombreux domaines d'application

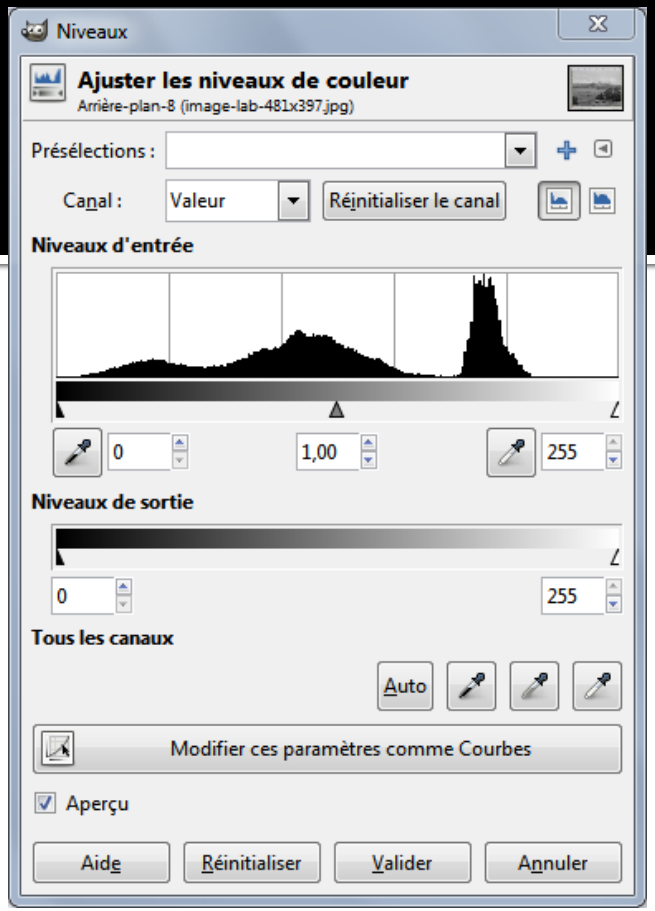
- Multimédia
  - Indexation, fouille dans des bases de données
  - Restauration d'images
- Imprimerie
  - OCR (optical character recognition)
  - Travaux sur les polices de caractères
- …

De très nombreux travaux existent dans le domaine du traitement d'image avec un vocabulaire et des approches différentes

Source C. Roudet

Histogramme

Histogramme



- Distribution de l'intensité des pixels d'une image
- L'histogramme indique le nombre de pixels de l'image pour chaque niveau de gris

I

Histogramme de I :
En abscisse : niveaux de gris [0,255]
En ordonnée : nombre de points de I possédant le niveau de gris en abscisse

Dans Gimp, pour obtenir l'histogramme d'une image aller dans le menu Couleurs puis Information et Histogramme

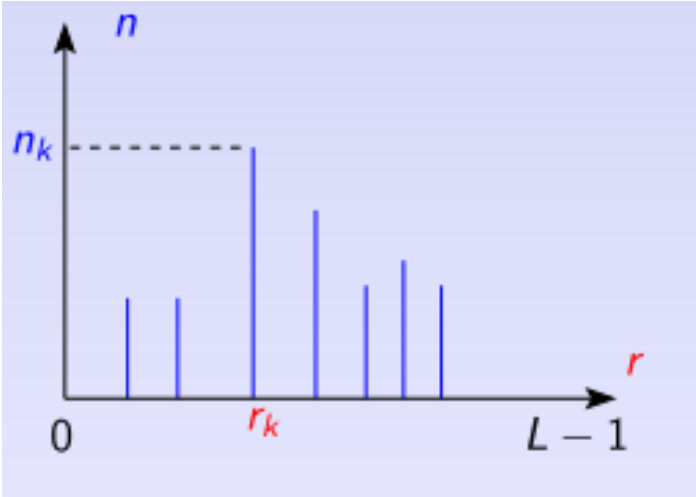
Histogramme

- Définition :

L'histogramme d'une image numérique possédant des intensités dans [0, L– 1] est une fonction discrète :

$$h(r_k) = n_k$$

- avec r_k représentant la $k^{ième}$ valeur d'intensité
- et n_k le nombre de pixels de l'image d'intensité r_k.



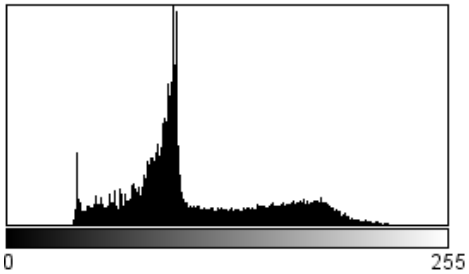
Dynamique d'une image

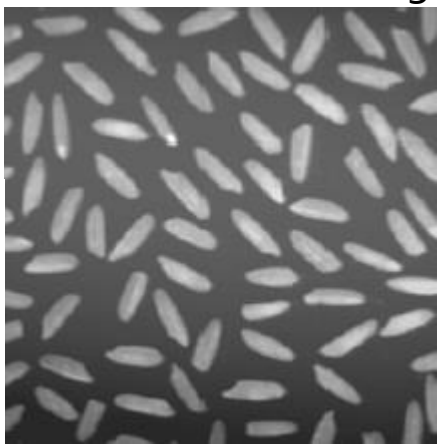
- La **dynamique d'une image** est l'intervalle

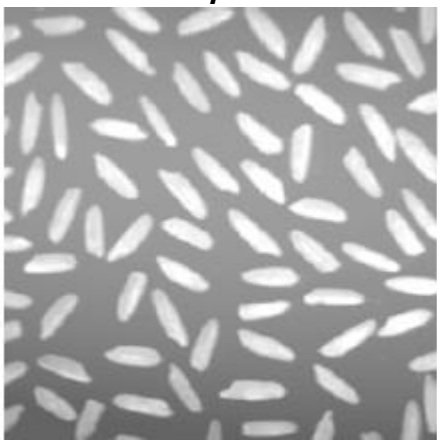
[valMin, valMax]

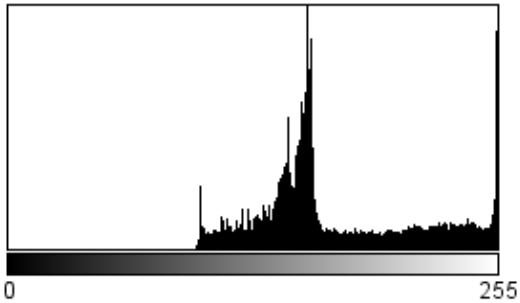
Luminance ou brillance d'une image

- La **luminance** (ou **brillance**) est définie comme la moyenne de l'intensité de tous les pixels de l'image.
  - Pour augmenter la luminance, il suffit de décaler l'histogramme : $I'(x,y) = I(x,y) + b$
  - Exemple : dans les deux images ci-dessous, seul la luminance change









Moy=108,4

Moy=170,5

Contraste d'une image

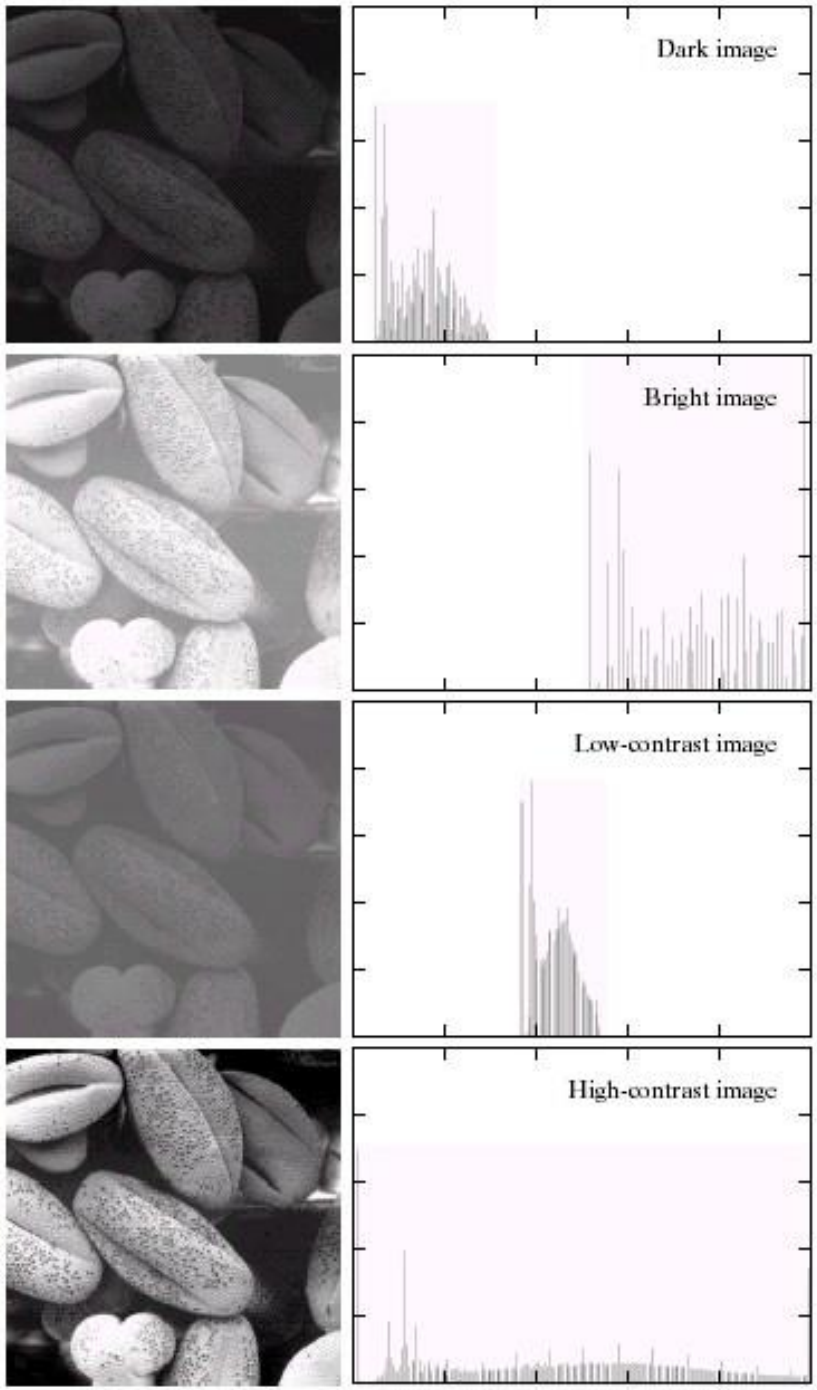
- Le **contraste d'une image** de taille L*H peut être défini de plusieurs façon :
  - Ecart-type des variations des niveaux de gris

$$C = \sqrt{\frac{1}{L * H} \sum_{y=0}^{H-1} \sum_{x=0}^{L-1} (I(x,y) - Moy)^2}$$

  - Variation entre niveaux de gris min et max

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

Exemples de contrastes d'une image



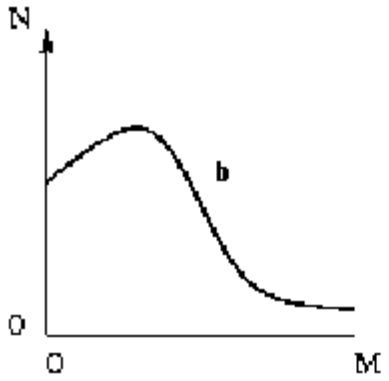
a b

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

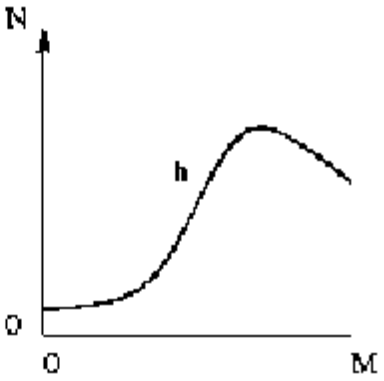
Source : Gonzalez and Woods. Digital Image Processing. Prentice-Hall, 2002.

Utilisation des histogrammes

- Permettent de mettre en valeur les défauts de contraste d'une image



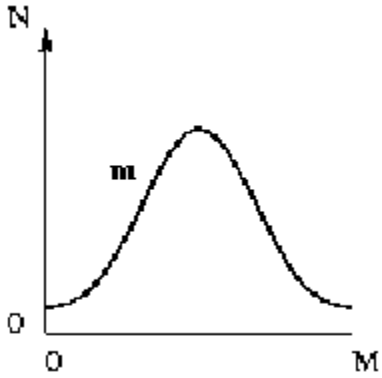
Les niveaux de gris sont concentrés vers la gauche: l'image est trop sombre



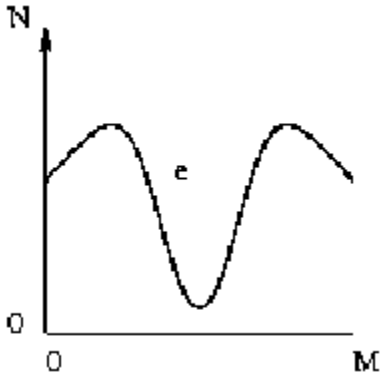
Les niveaux de gris sont concentrés vers la droite: l'image est trop claire

Utilisation des histogrammes

- Permettent de mettre en valeur les défauts de contraste d'une image



Les niveaux de gris sont concentrés vers le milieu: l'image manque de contraste



Les niveaux de gris sont concentrés vers la droite et la gauche: l'image manque de nuances

Opérations sur les images

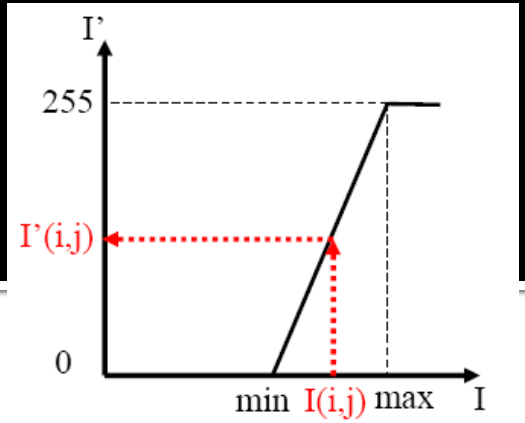
- Transformation des niveaux de gris d'une image
- Pour traiter le contraste d'une image **I**, on applique uniformément une transformation f à tous les pixels de l'image
- On obtient une nouvelle image **I'** telle que :

$$I'(p)=f(I(p)) \text{ pour tous les pixels } p$$

Opérations sur les images pour améliorer le contraste

- Transformations linéaires
- Transformations linéaires avec saturation
- Transformations linéaires par morceaux
- Transformations non-linéaires
- Égalisation de l'histogramme

Opération sur les images : Etirement de contraste



- Le niveau de gris de chaque pixel de l'image I est transformé par la fonction linéaire :

$$I'(x,y)=M(\ I(x,y)-\min\ )/(\ \max-\min)$$

Avec

- M une valeur constante fixée (par ex M=255)
- min (resp max) le niveau de gris minimal (resp maximal) de l'image I
- Une division entière

➢ La dynamique de I est étirée de I pour obtenir dans I', min =0 et max =M

Opération sur les images : Etirement de contraste

- Exemple : I

40	60	45
38	56	47
52	59	58

min =38 max =60

On souhaite effectuer un étirement de contraste entre 0 et 255 et obtenir une image I'

Opération sur les images : Etirement de contraste

- Exemple : I

40	60	45
38	56	47
52	59	58

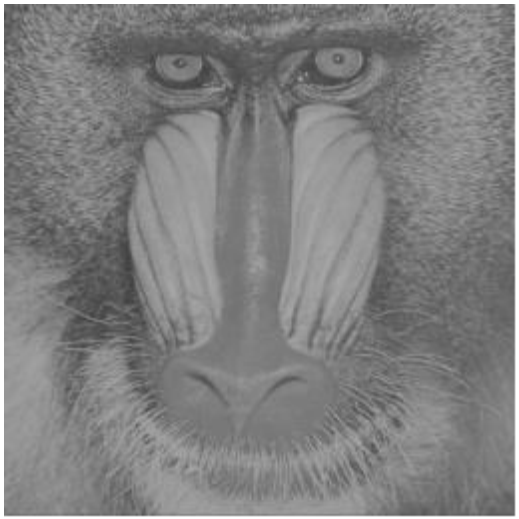
Etirement de contraste entre 0 et 255 :

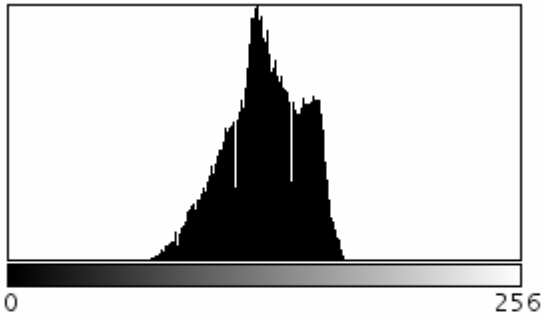
I'

23	255	81
0	208	104
162	243	231

Opération sur les images : Etirement de contraste – un exemple

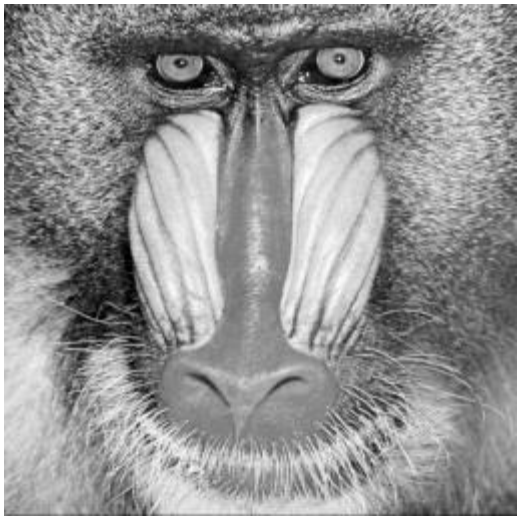
I

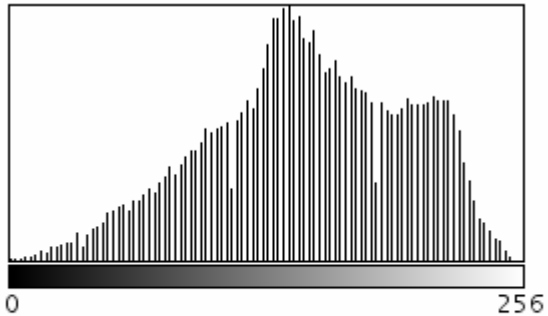




Count: 65536 Min: 65
Mean: 127.287 Max: 169
StdDev: 19.358 Mode: 124 (1544)

I'





Count: 65536 Min: 0
Mean: 147.073 Max: 255
StdDev: 49.860 Mode: 139 (1544)

Opération sur les images : Etirement de contraste – implantation

- Première idée :

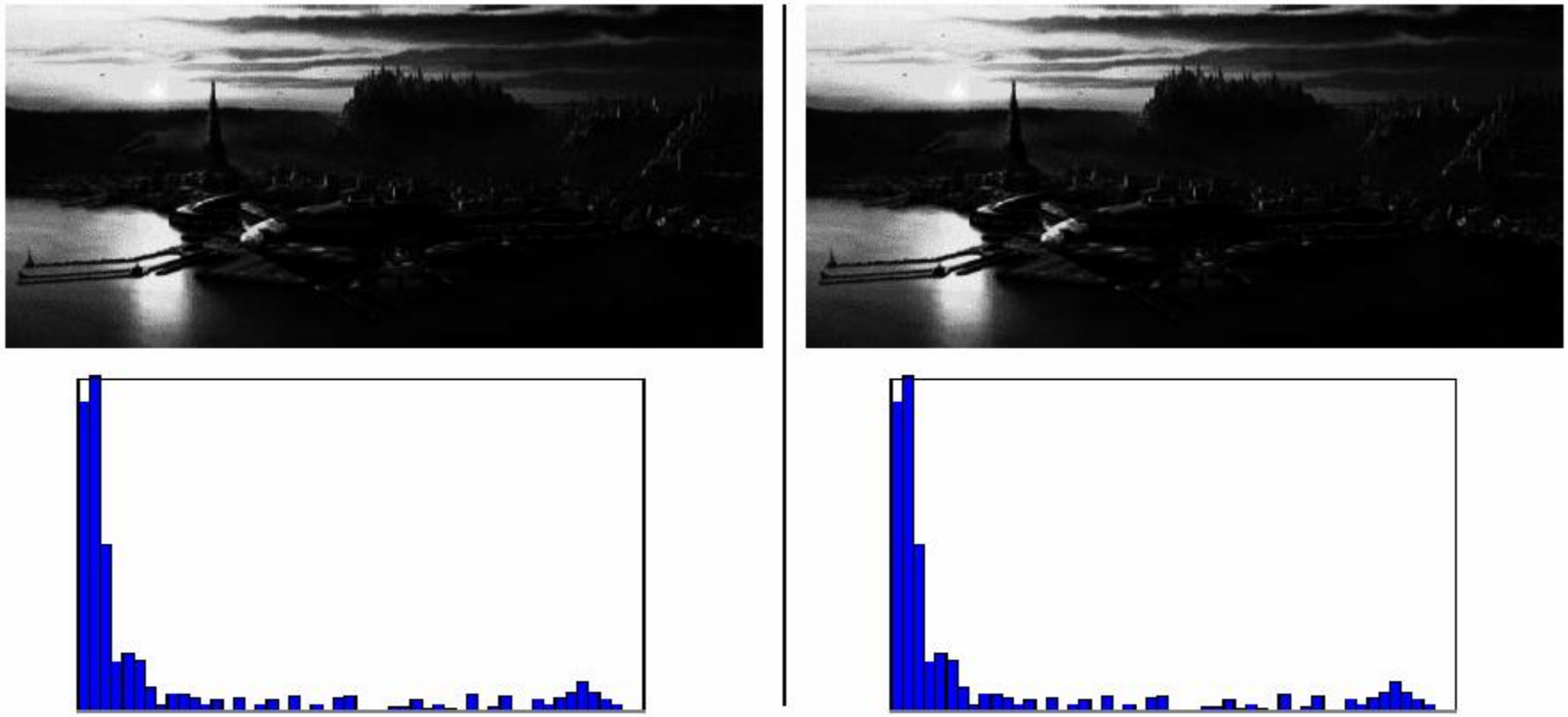
```
for(i = 0; i < L ; i++)
    for(j = 0; j < H; j++)
            I' (i , j) = 255( I(i,j) – min)/(max – min)
```

- Utilisation d'une "Look Up Table" :

```
// Initialisation de la LUT
for(ng = 0; ng < 256; ng++)
    LUT[ng] = 255 (ng – min) / (max – min)
// Calcul de la transformation
for(i = 0; i < L; i++)
    for(j = 0; j < H; j++)
        I' (i, j) =  LUT[I (i, j)]
```

Opération sur les images : Etirement de contraste – limite

- Si la dynamique est maximale, la transformation n'apporte aucun changement



Source A. Vialard

Opération sur les images : Transformation linéaire avec saturation

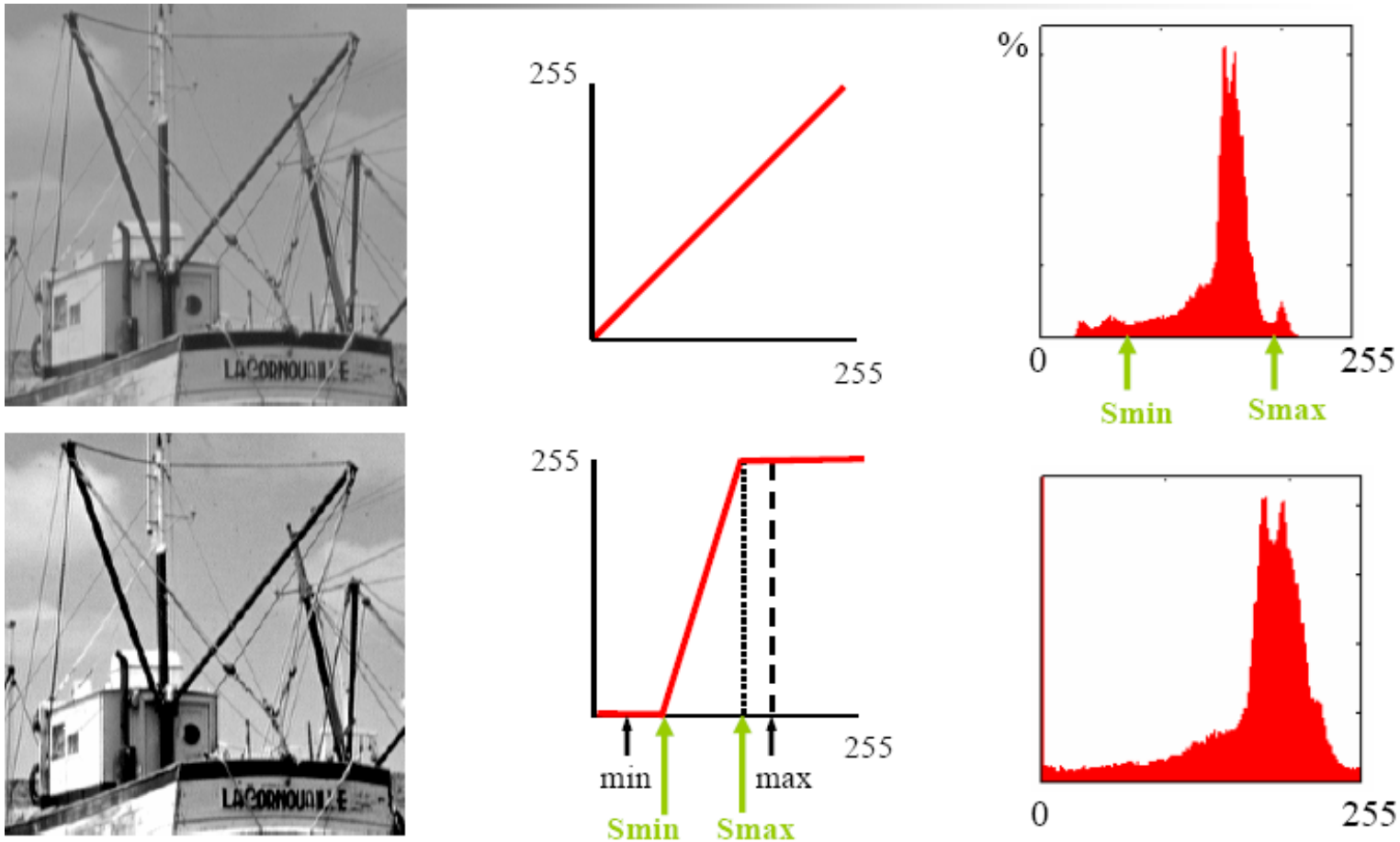
- Utilisation de valeurs Smin et Smax telles que :
  - $\min(I(i, j)) < Smin < Smax < \max(I(i, j))$

$I'(i, j) = 255(I(i,j) - Smin)/(Smax - Smin)$

Si $I'(i, j) < 0$ alors $I'(i, j) = 0$

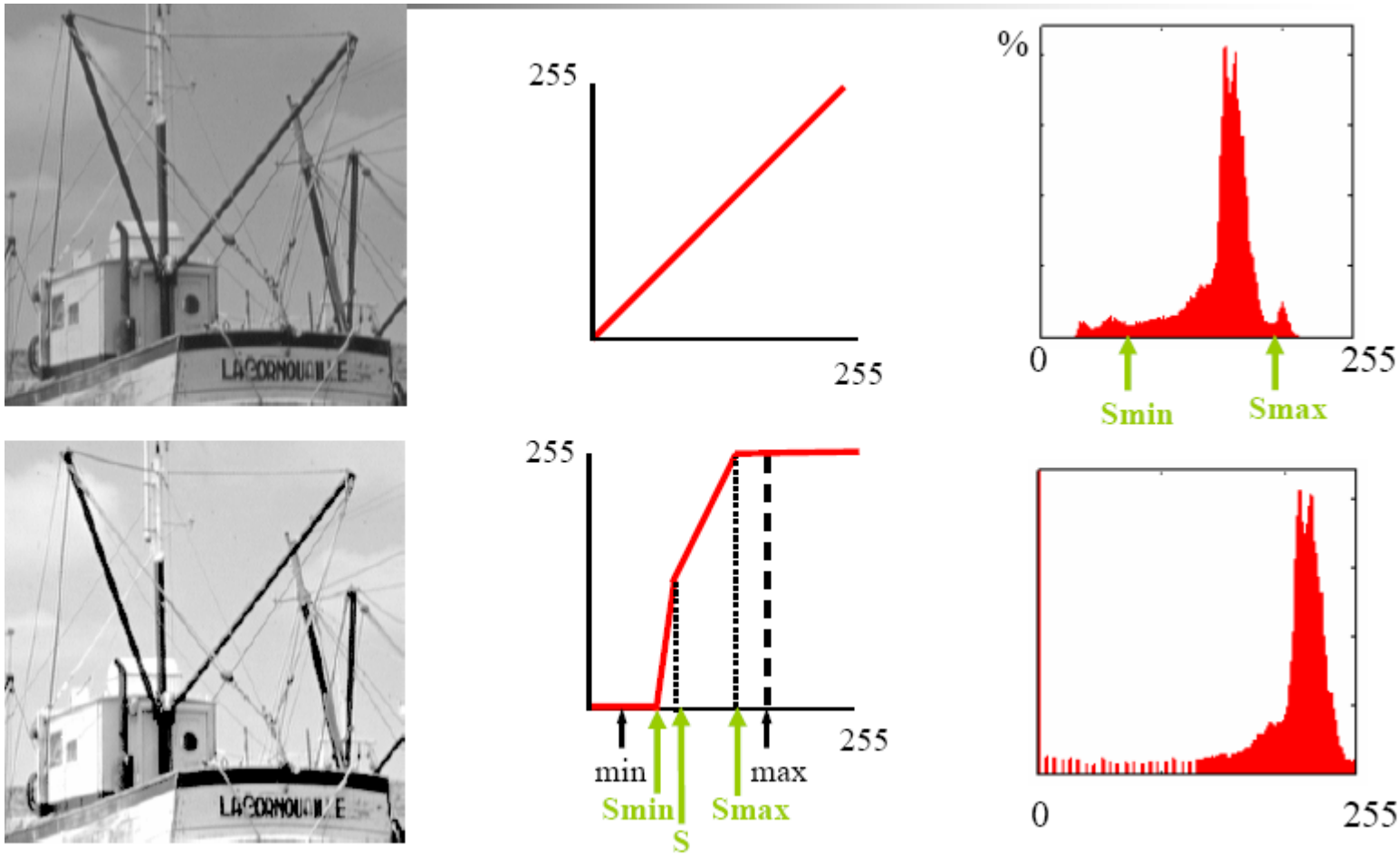
Si $I'(i, j) > 255$ alors $I'(i, j) = 255$

Transformation linéaire avec saturation - exemple



ource : Caroline Rougier. Traitement d'images (IFT2730). Univ. de Montréal.

Transformation linéaire par morceaux

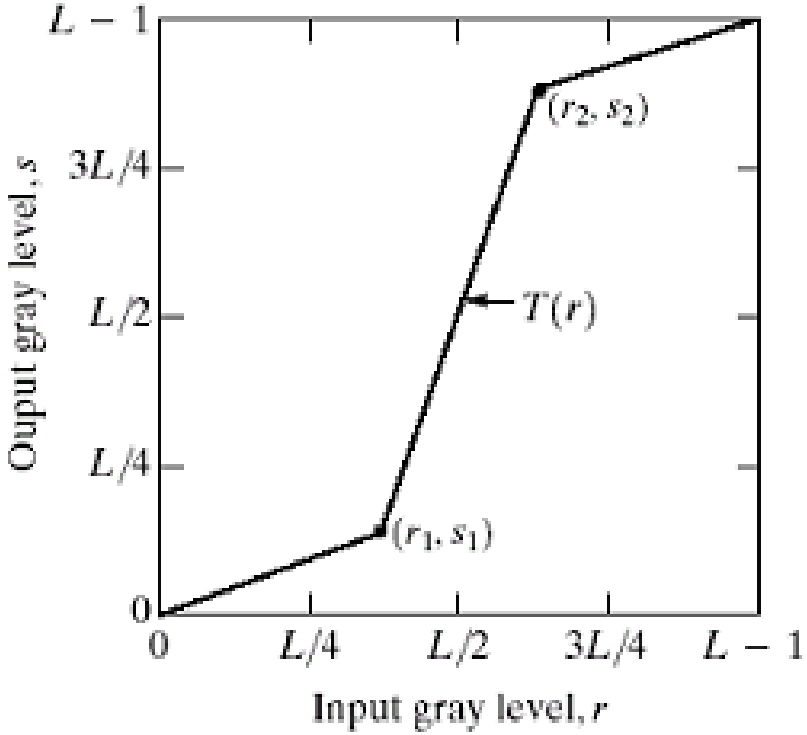


urce : Caroline Rougier. Traitement d'images (IFT2730). Univ. de Montréal.

Transformation linéaire par morceaux

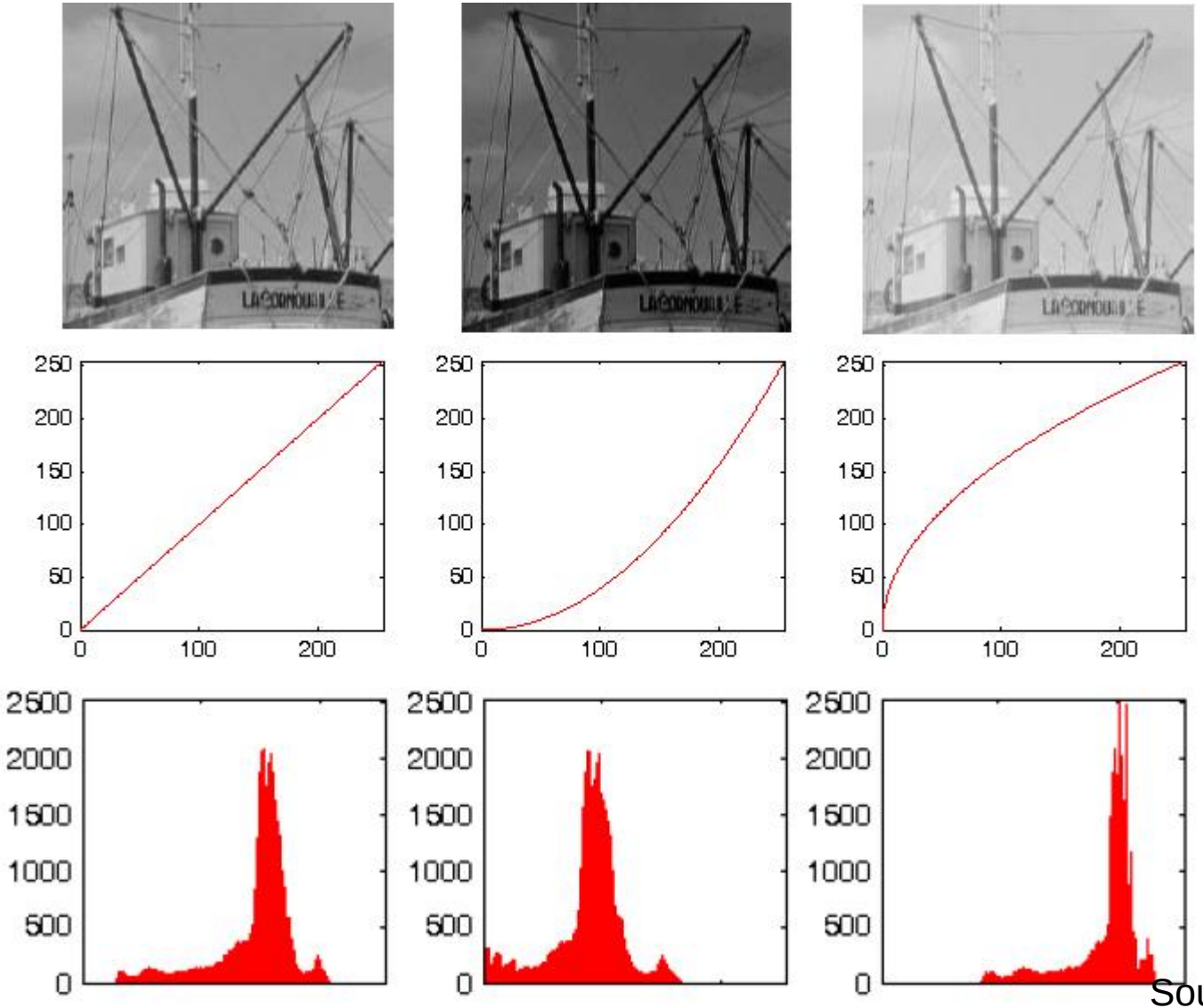
- Transformation par morceau
  - Peut être plus complexe si besoin
  - Nécessite la définitions de variables

- (r1,s1) et (r2,s2) contrôlent la fonction



Source S. Paris

Transformations non linéaires



Source A. Vialard

Transformations non linéaires

- Exponentielles

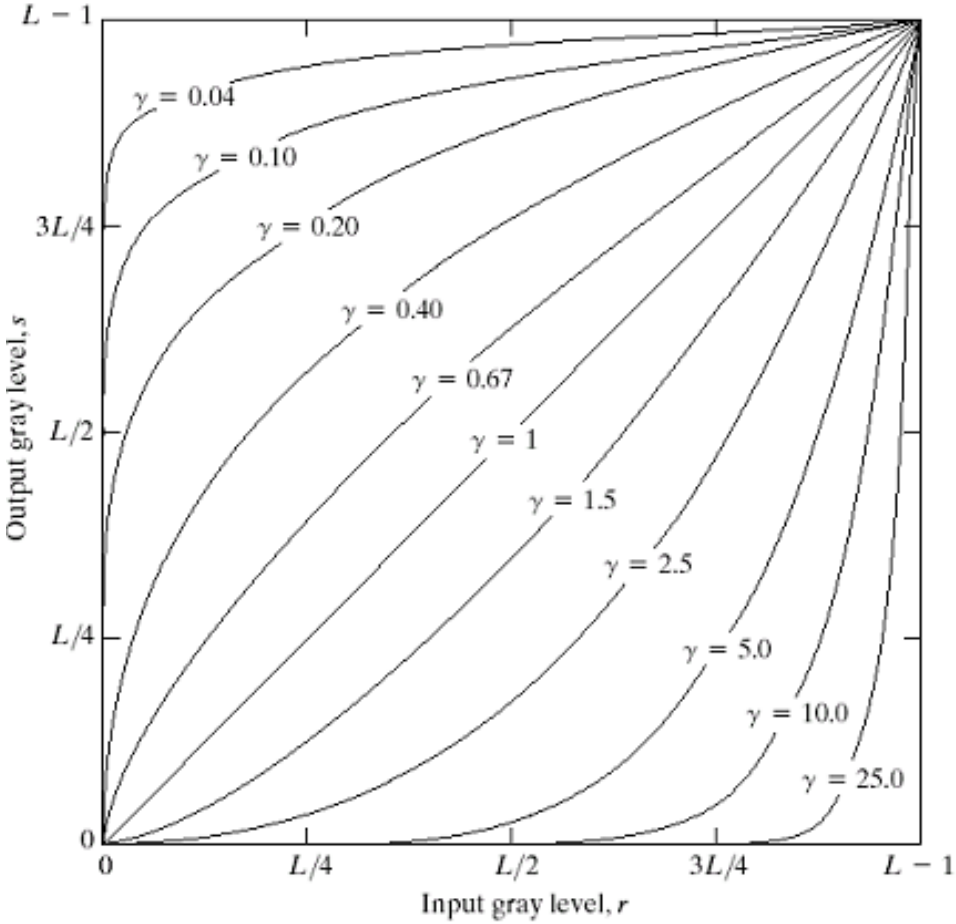
$$s = c\, r^{\gamma}$$

$\gamma > 1$

- Étire les valeurs pâles
- Compresse les valeurs foncés

$\gamma < 1$

- Compresse les valeurs pâles
- Étire les valeurs foncés



Source S. Paris

Transformations non linéaires

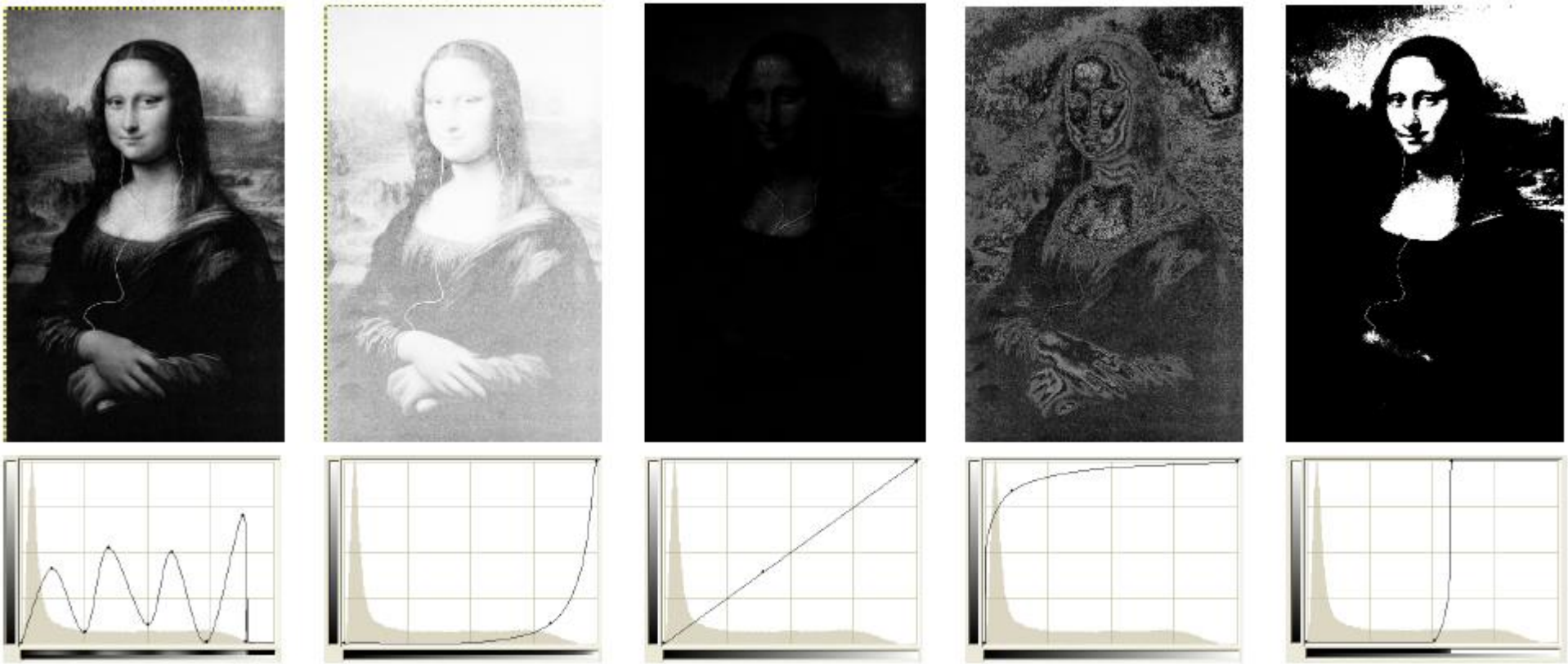
a b
c d

FIGURE 3.9 (a) Aerial image. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0$, and 5.0, respectively. (Original image for this example courtesy of NASA.)

Source : Gonzalez and Woods. Digital Image Processing. Prentice-Hall, 2002.

exercice

- Associer à chaque image la transformation qui lui a été appliquée



Egalisation d'histogramme (1)

- **Histogramme normalisé P_I**
  - $H_I(n)$ le nombre de pixels de l'image I de taille L*H qui ont la valeur n
  - P_I est défini de {0, …,N-1} dans [0,1] par

$$P_I(n) = H_I(n) / L*H$$

P_I est la proportion de pixels de I qui ont la valeur n

Egalisation d'histogramme (2)

- **Histogramme cumulé normalisé R_I**
  - R_I est défini de {0, …,N-1} dans [0,1] par

$$R_I(n) = \sum_{k=0}^{n} PI(k)$$

R_I est la proportion de pixels de I dont la valeur est inférieure ou égale à n

Egalisation d'histogramme (3)

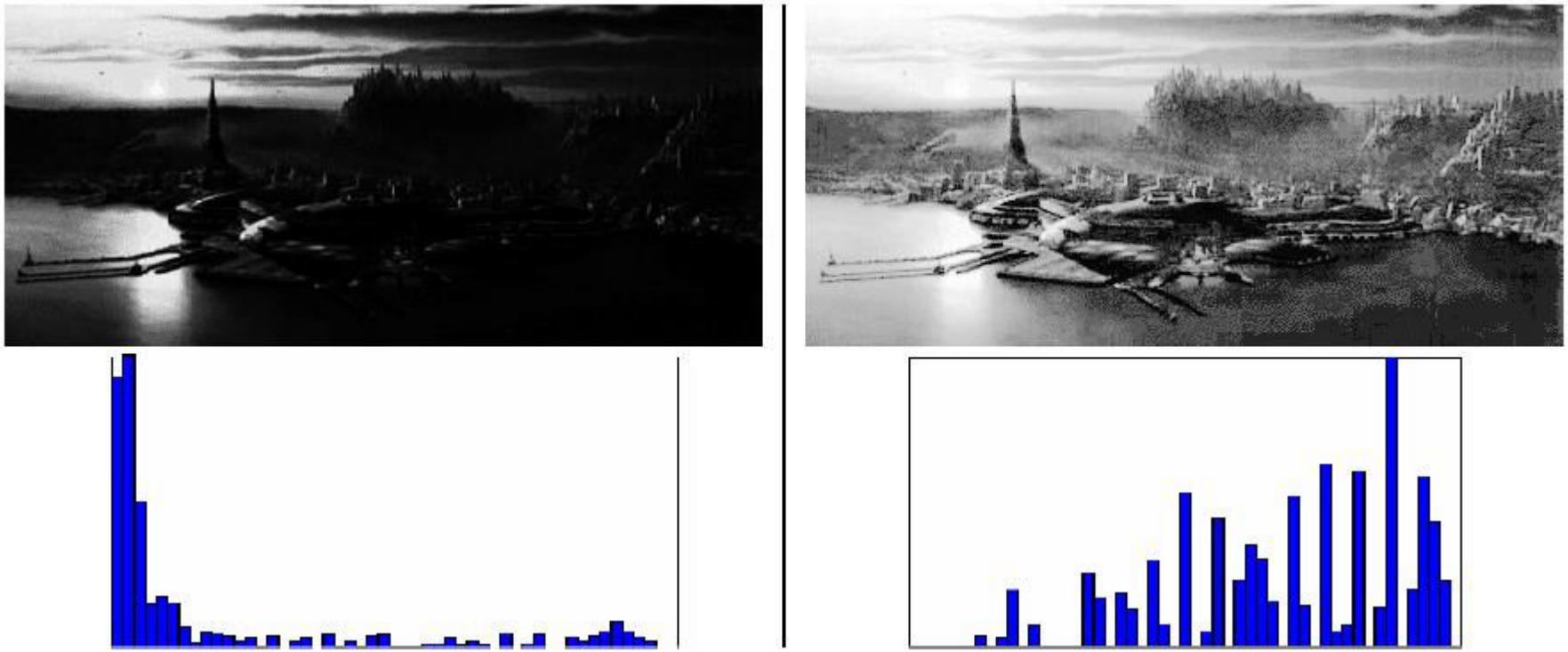
- **L'égalisation d'histogramme** est une transformation des niveaux de gris dont le principe est d'équilibrer le mieux possible la dynamique de I
  - Transformation à effectuer sur I pour obtenir l'égalisation de l'histogramme

$$I'(x,y)=(N-1)*R_I(I(x,y))$$

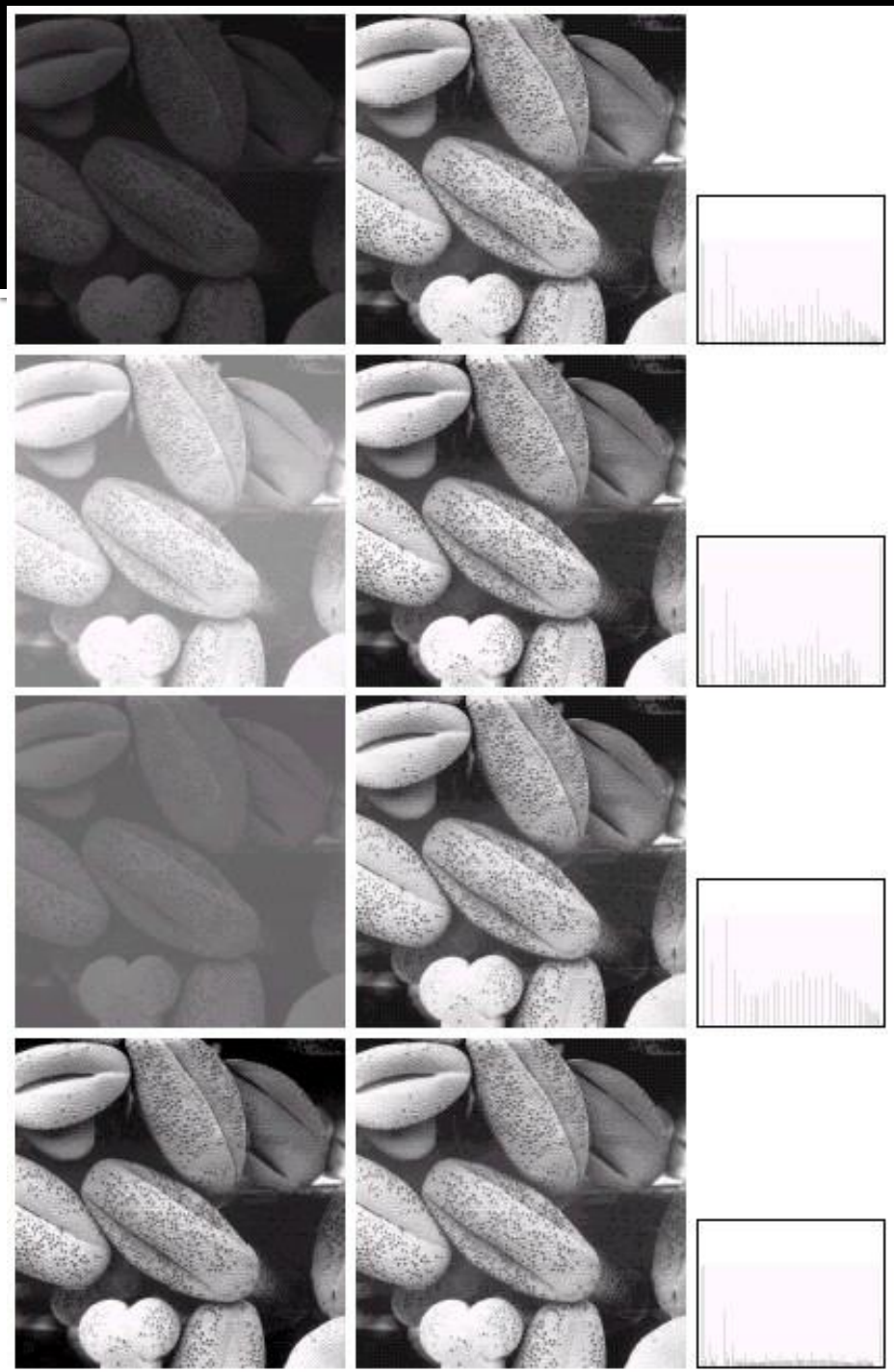
    - objectif histogramme plat ce qui traduit une équiprobabilité des différents niveaux de gris
    - Le nouvel histogramme cumulé normalisé est le plus linéaire possible

Egalisation d'histogramme (4)

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



Source A. Boucher



Egalisation d'histogramme (5)

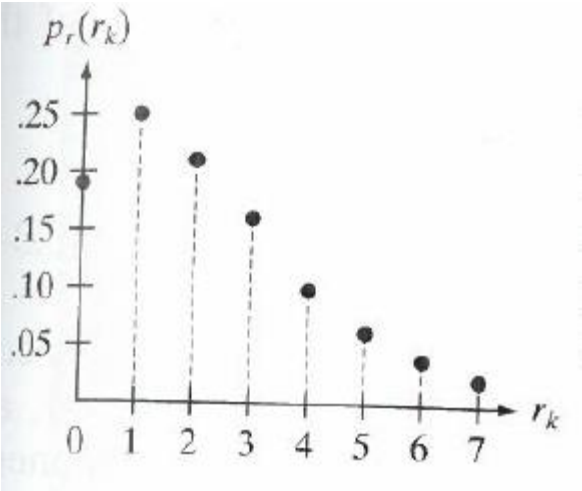
- *Si on prend la **même image** avec des **contrastes différents**, l'égalisation d'histogramme donne un **résultat visuel similaire** pour toutes les images.*

Source : Gonzalez and Woods. Digital Image Processing. Prentice-Hall, 2002.

Un exemple simple

- Une image I de 64*64 pixels , 8 niveaux d'intensité, L*H=4096 dont la distribution d'intensité est

Intensité k	HI(k)	PI(k)
0	790	0,19
1	1023	0,25
2	850	0,21
3	656	0,16
4	329	0,08
5	245	0,06
6	122	0,03
7	81	0,02



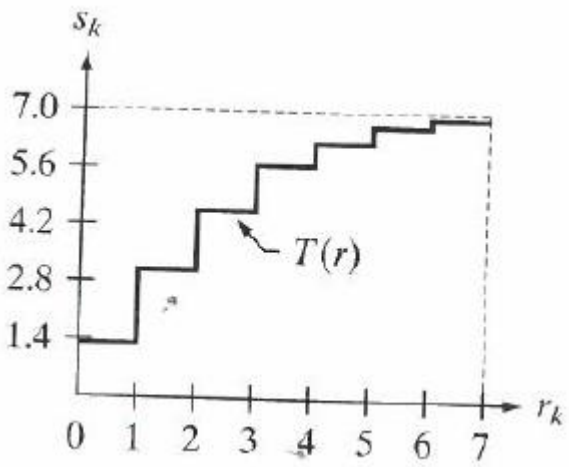
Histogramme normalisé

Calculer les premières intensités transformées par égalisation d'histogramme

Un exemple simple

- Une image I de 64*64 pixels , 8 niveaux d'intensité, L*H=4096 dont la distribution d'intensité est

Intensité k	HI(k)	PI(k)	Intensité obtenue
0	790	0,19	1,33 →1
1	1023	0,25	3,08 →3
2	850	0,21	4,55 →5
3	656	0,16	5,67 →6
4	329	0,08	6,23 →6
5	245	0,06	6,65 →7
6	122	0,03	6,86 →7
7	81	0,02	7 →7



Fonction de transformation d'intensité

Un exemple simple

- Une image I de 64*64 pixels , 8 niveaux d'intensité, L*H=4096 dont la distribution d'intensité est

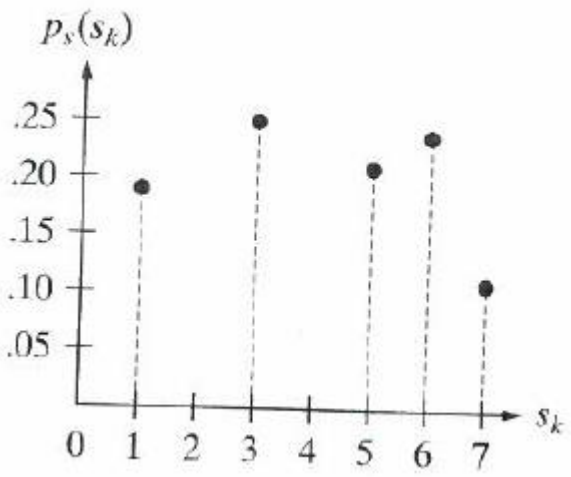
Intensité k	HI(k)	PI(k)	Intensité obtenue
0	790	0,19	1,33 →1
1	1023	0,25	3,08 →3
2	850	0,21	4,55 →5
3	656	0,16	5,67 →6
4	329	0,08	6,23 →6
5	245	0,06	6,65 →7
6	122	0,03	6,86 →7
7	81	0,02	7 →7

En déduire l'histogramme normalisé de l'image transformée correspondant à l'histogramme égalisé de l'image initiale .

Un exemple simple

- Une image I de 64*64 pixels , 8 niveaux d'intensité, L*H=4096 dont la distribution d'intensité est

Intensité k	HI(k)	PI(k)	Intensité obtenue
0	790	0,19	1,33 →1
1	1023	0,25	3,08 →3
2	850	0,21	4,55 →5
3	656	0,16	5,67 →6
4	329	0,08	6,23 →6
5	245	0,06	6,65 →7
6	122	0,03	6,86 →7
7	81	0,02	7 →7



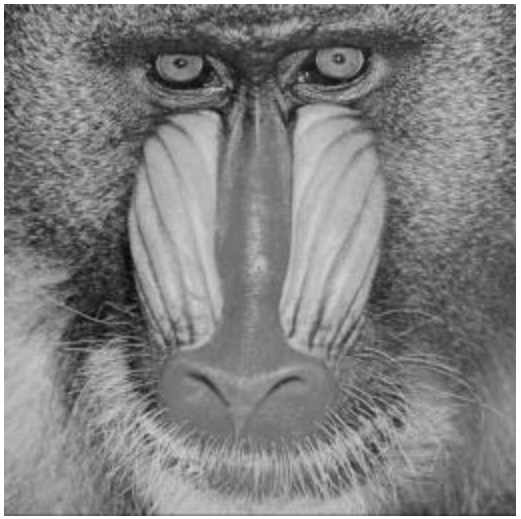
En déduire l'histogramme normalisé de l'image transformée correspondant à l'histogramme égalisé de l'image initiale .

Négatif

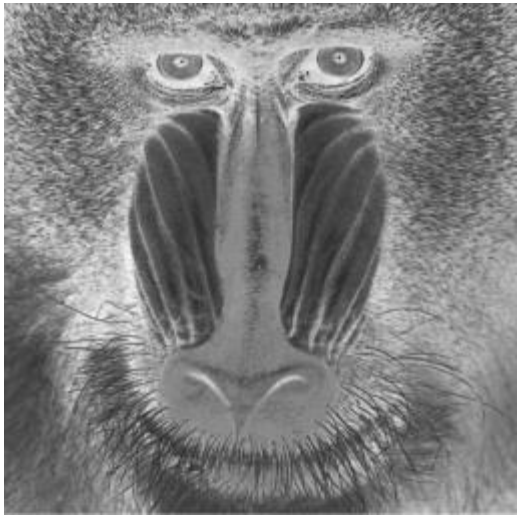
Le négatif N(I) d'une image I est obtenu en inversant l'échelle des niveaux de gris de 0 à M

En tout pixel **p** on a:

$$N(I)(p)=M-I(p)$$



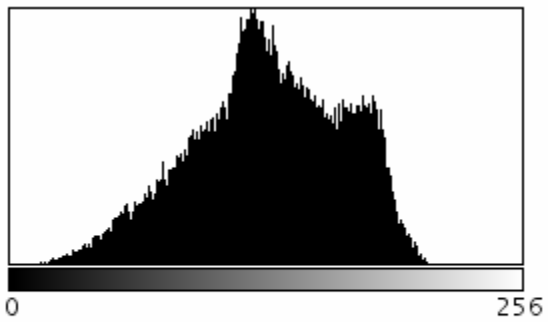
I



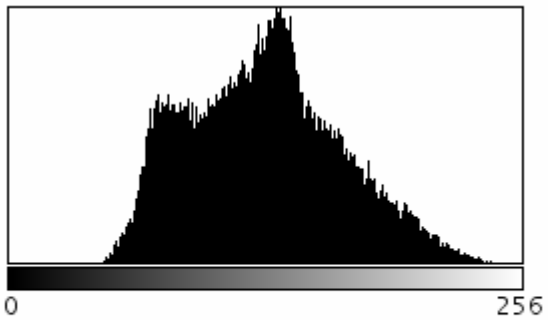
N(I)

Négatif

Conséquence sur l'histogramme:
Inversion



I



N(I)

Mes sources

- Livres
  - Introduction au traitement d'Images de Diane Lingrand chez Vuibert
  - Analyse d'images : filtrage et segmentation de JP Cocquerez et al chez Masson
  - Digital Image Processing, RC Gonzales RE Woods, Pearson International edition

Mes sources

- Cours en Traitement d'Images de
  - Alain Boucher
    - http://www2.ifi.auf.org/personnel/Alain.Boucher/cours/traitement_images/02-Traitement_de_base.pdf
  - Anne Vialard
    - http://dept-info.labri.fr/~vialard/Traitement/
  - Céline Roudet
    - http://ufrsciencestech.u-bourgogne.fr/~roudet/teaching.html
  - Stéphane Paris
    - http://sparis.free.fr/
  - Benoit Naegel
    - http://webloria.loria.fr/~naegelbe/index.php/enseignement
  - Bertrand Kerautret
    - http://www.loria.fr/~kerautre/
  - Sébastien Thon
    - http://www.iut-arles.up.univ-mrs.fr/thon/LP/index.php