

Vision par ordinateur

Techniques d'amélioration d'images

2^{ème} année GI et 3^{ème} année AE3S – ENSMR

Master *Intelligent Processing Systems* –FSR

I. BENMILOUD & N. ZRIRA

Année universitaire 2019/2020

Plan de la séance

- Traitement de bas niveau d'une image
 - Définition de l'histogramme d'une image
 - Expansion dynamique
 - Egalisation d'histogramme
- Opérations arithmétiques
- Opérations logiques ET/OU/XOR
- Transformation géométrique

But

- Modifier la dynamique de l'image qui est trop claire ou trop foncée
- Faire apparaître des régions et rendre visible certains détails
- Rendre les images plus facile à interpréter par l'être humain et par la machine

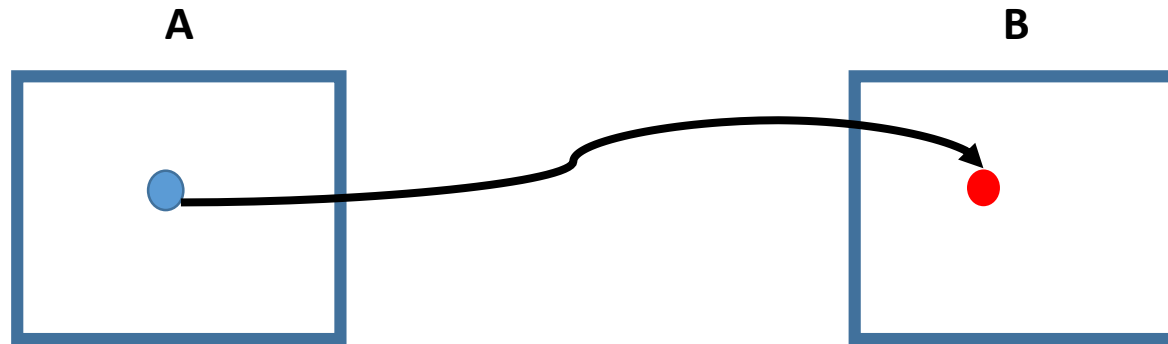
Amélioration d'images

Principe

- Changer la valeur de chaque pixel d'une image I pour obtenir une nouvelle image I' .
- Cette image résultat a la même taille que I , mais des propriétés plus intéressantes.

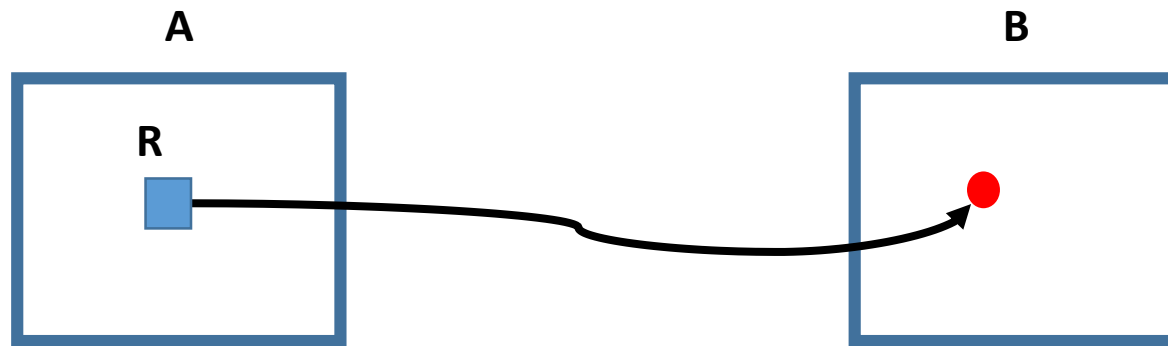
Types d'amélioration d'images (1)

- Transformation ponctuelle: $I(x, y) \xrightarrow{t} I'(x, y) = t(I(x, y))$
 - Seuillage, ajustement luminosité/Contraste
 - Manipulation d'histogramme



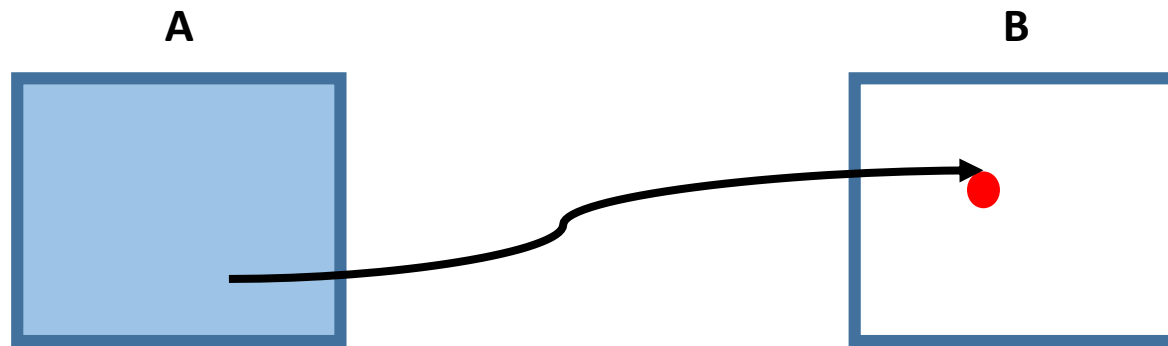
Types d'amélioration d'images (2)

- Transformation locale: $I(x, y) \xrightarrow{t} I'(x, y) = t(I(V(x, y)))$
 - Filtrage



Types d'amélioration d'images (3)

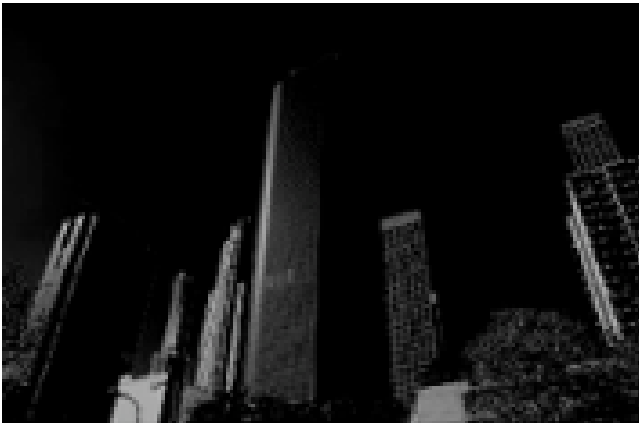
- Transformation globale: $I(x, y) \xrightarrow{t} I'(x, y) = t(I)$
 - Transformation dans l'espace de Fourier
 - Méthodes markoviennes



Propriétés de l'image: Brillance/Luminance

Elle est définie comme la moyenne de tous les pixels de l'image

$$B = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N I(i, j)$$



B=12



B=76



B=173

Propriétés de l'image: Contraste

Définition 1: Ecart type de la variance des niveaux de gris

$$C = \sqrt{\frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (I(i, j) - B)^2}$$

Définition 2: Variation entre niveaux de gris min et max

$$C = \frac{\max(I(i, j)) - \min(I(i, j))}{\max(I(i, j)) + \min(I(i, j))}$$

Propriétés de l'image: Contraste



Contraste faible



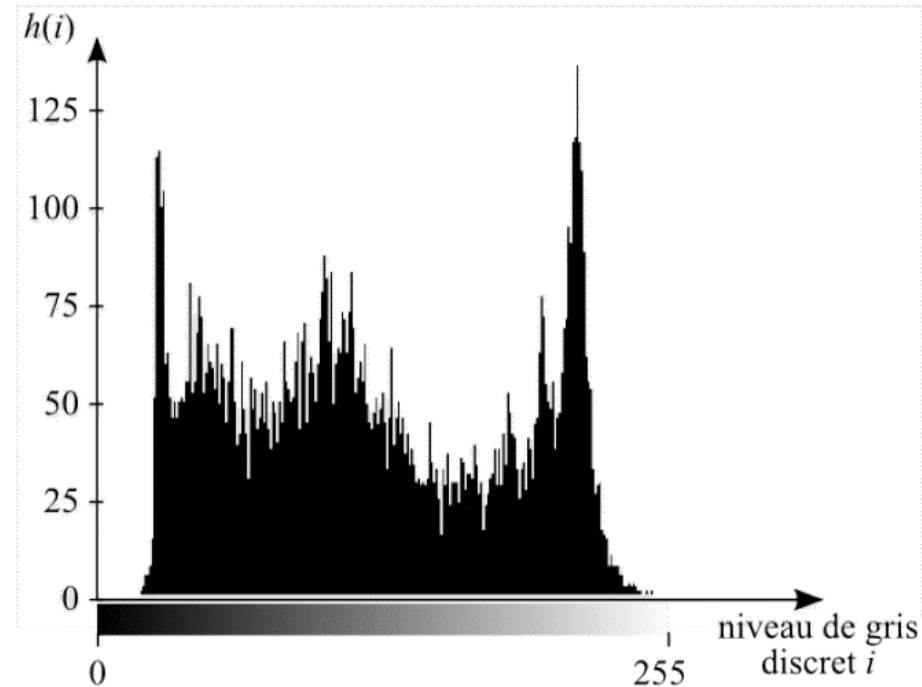
Contraste fort

Comment agir sur le contraste?

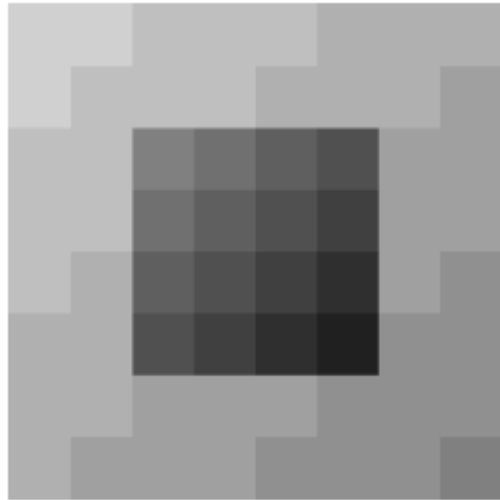
- Transformation linéaire
- Transformation linéaire avec saturation
- Transformation linéaire par morceau
- Transformation non-linéaire

Histogramme d'images: Définition et notation

- L'histogramme d'une image en niveau de gris dénombre les **occurrences** de chacun des niveaux.
- $h(i)$ = nombre de pixels dans l'image I ayant le niveau de gris i .



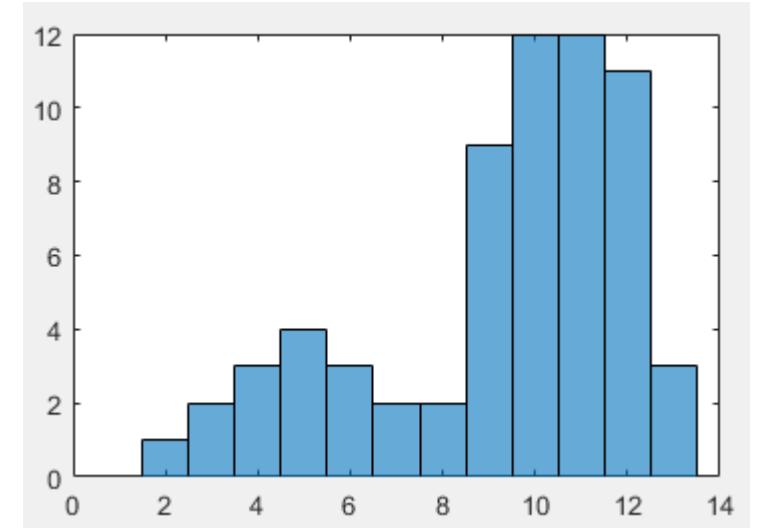
Histogramme d'images: Exemple



Les valeurs des
pixels



13	13	12	12	12	11	11	11
13	12	12	12	11	11	11	10
12	12	8	7	6	5	10	10
12	12	7	6	5	4	10	10
12	11	6	5	4	3	10	9
11	11	5	4	3	2	9	9
11	11	10	10	10	9	9	9
11	10	10	10	9	9	9	8



Pixel x	0	1	2	3	4	5	6	7	8	9	10	11	12	13
h	0	0	1	2	3	4	3	2	2	9	12	12	11	3

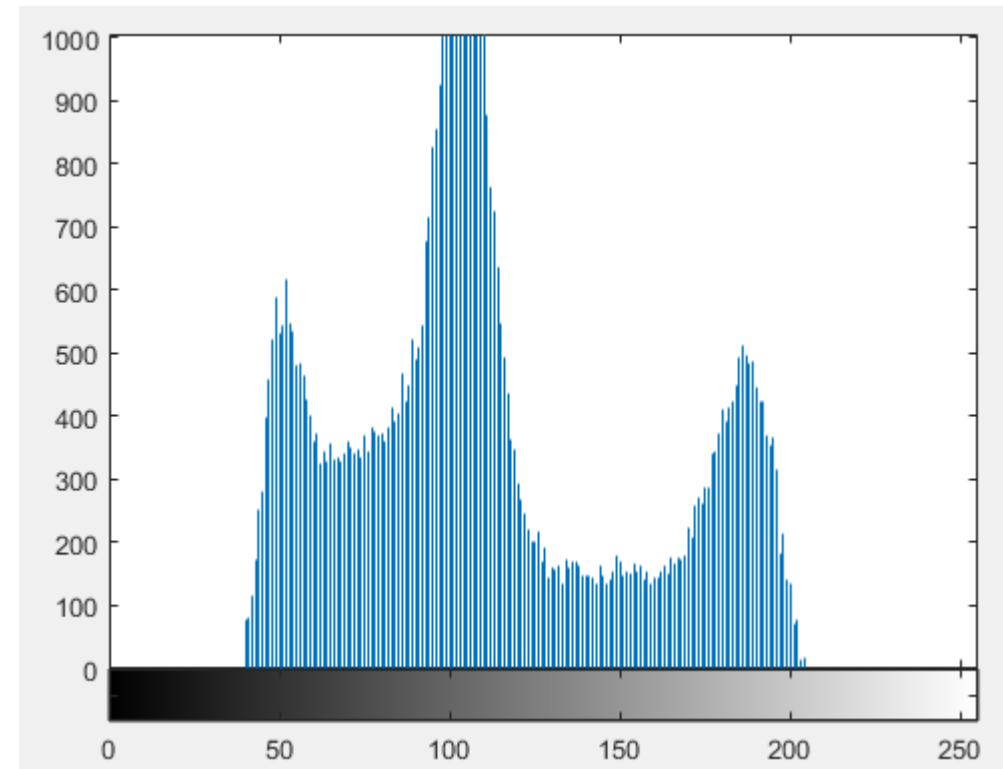
Histogramme d'images: En pratique

Le pseudo code pour calculer l'histogramme h d'une image I codée sur 8 bits

```
int h[256]=0, x,y,I;  
for ( x =0; x < M ; x++ )  
    for( y =0; y < N ; y++)  
        h[I[x,y]]++
```

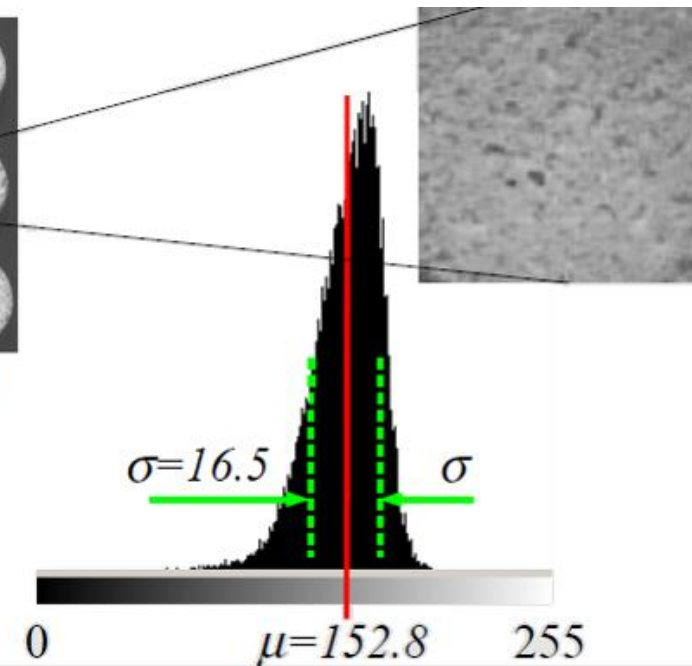
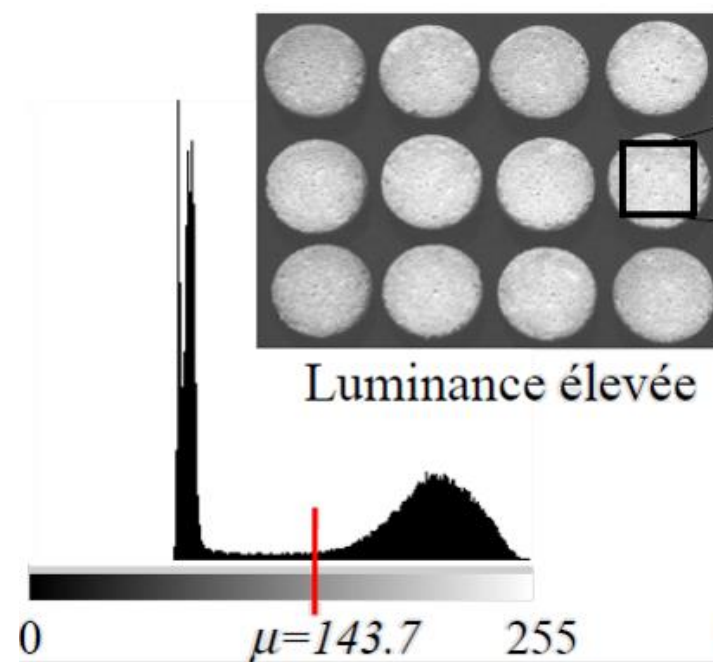
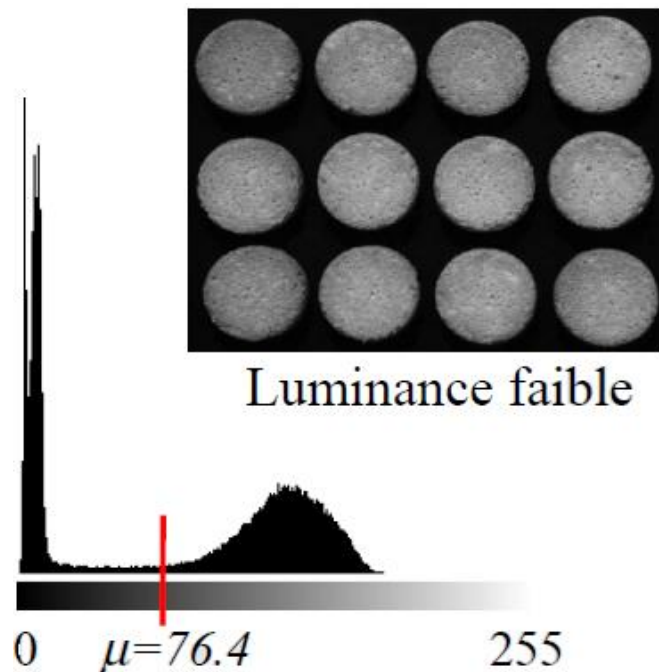
En Matlab

```
I=imread('rice.png');  
h=imhist(I);
```



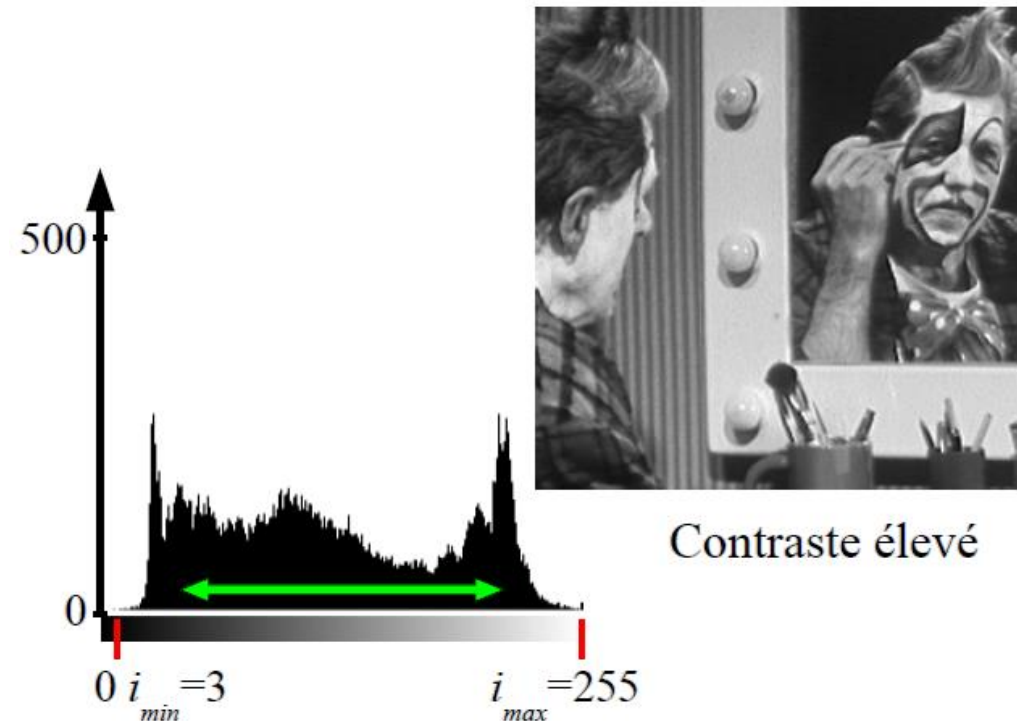
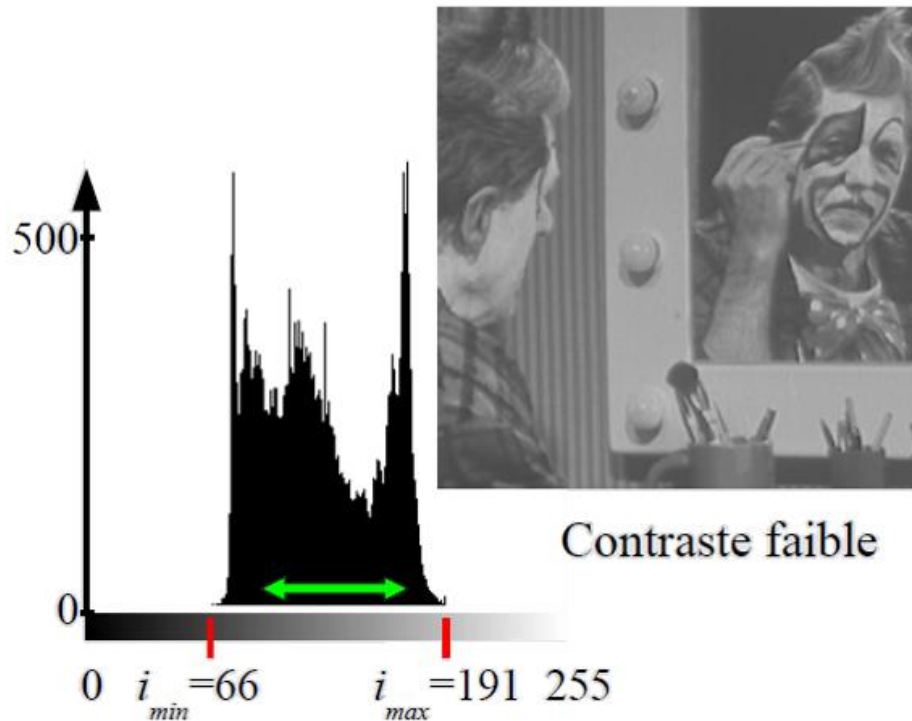
Histogramme d'images: Interprétation (1)

- **Luminance** de l'image = moyenne μ des niveaux de gris
- **Ecart-type** σ = amplitude moyenne de la variation des niveaux de gris d'une part et d'autre part de la moyenne



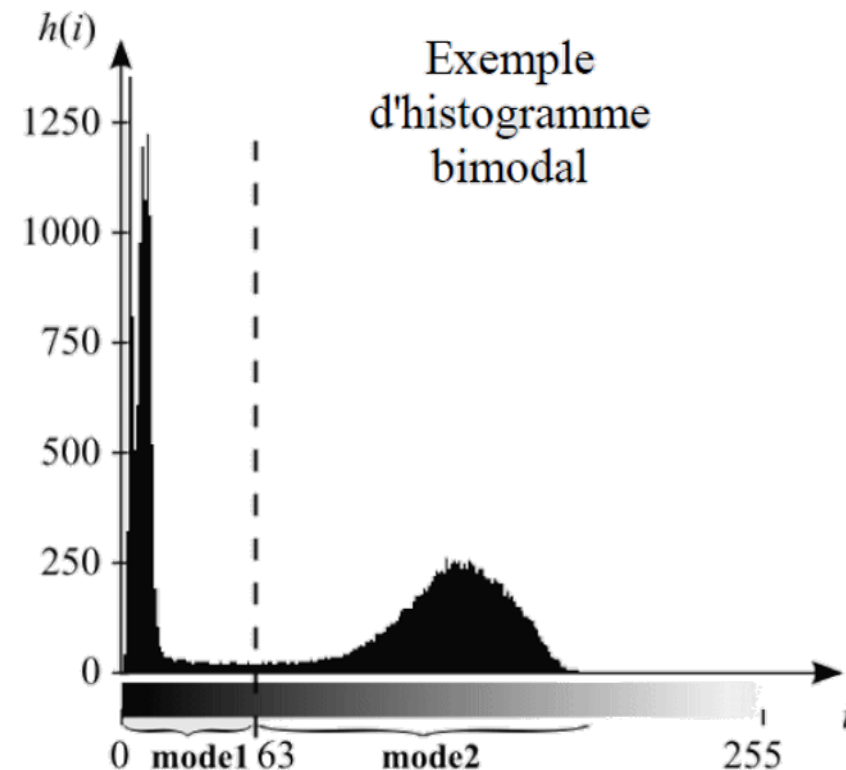
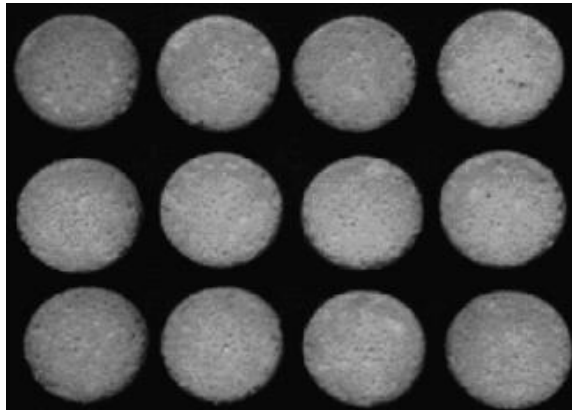
Histogramme d'images: Interprétation (2)

- **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

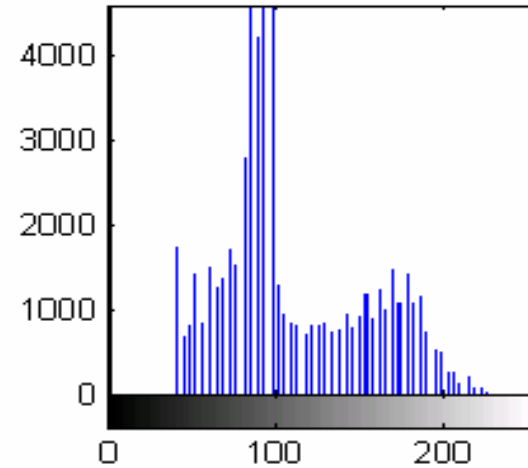
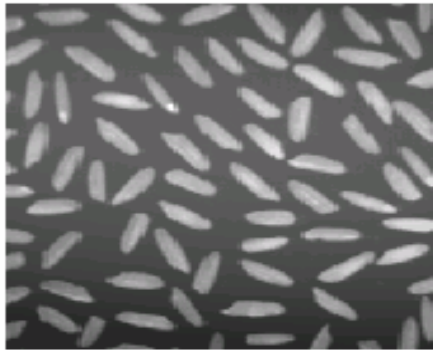


Histogramme d'images: Interprétation (3)

- Présence de **pics** significatifs pour certaines plages de niveaux de gris (appelées **modes**), correspondent à des ensembles de pixels intéressants.



Histogramme d'images: Interprétation (4)



**Présence de deux pics
Histogramme bimodal**

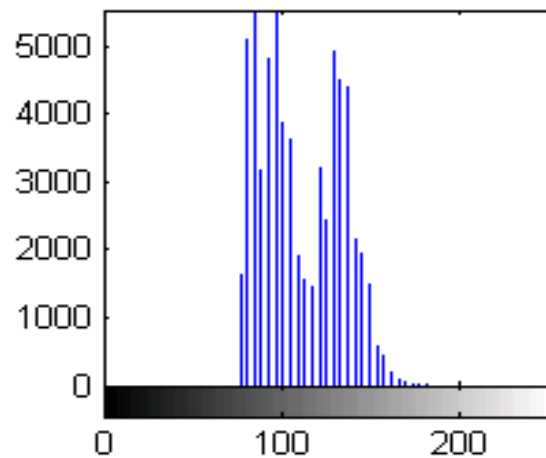
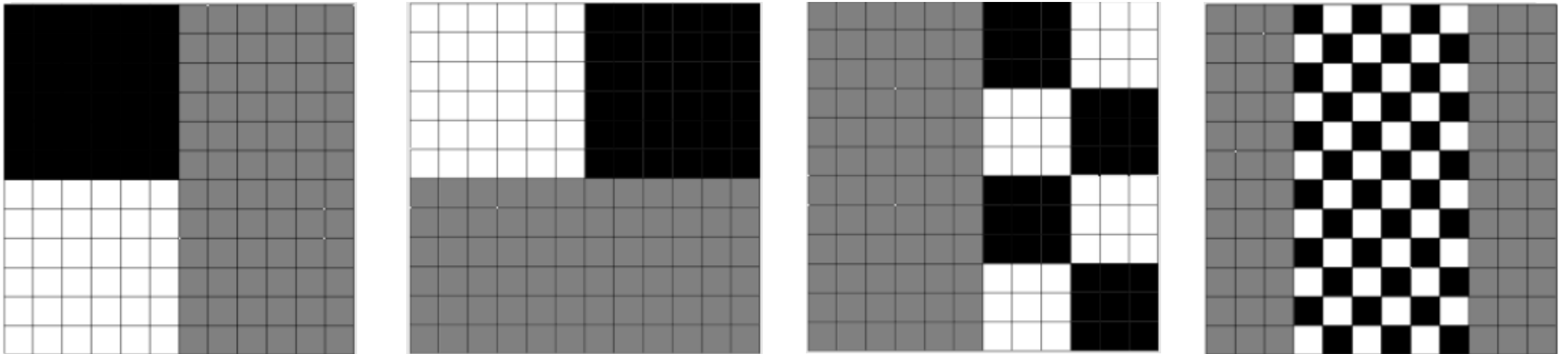


Image mal contrastée

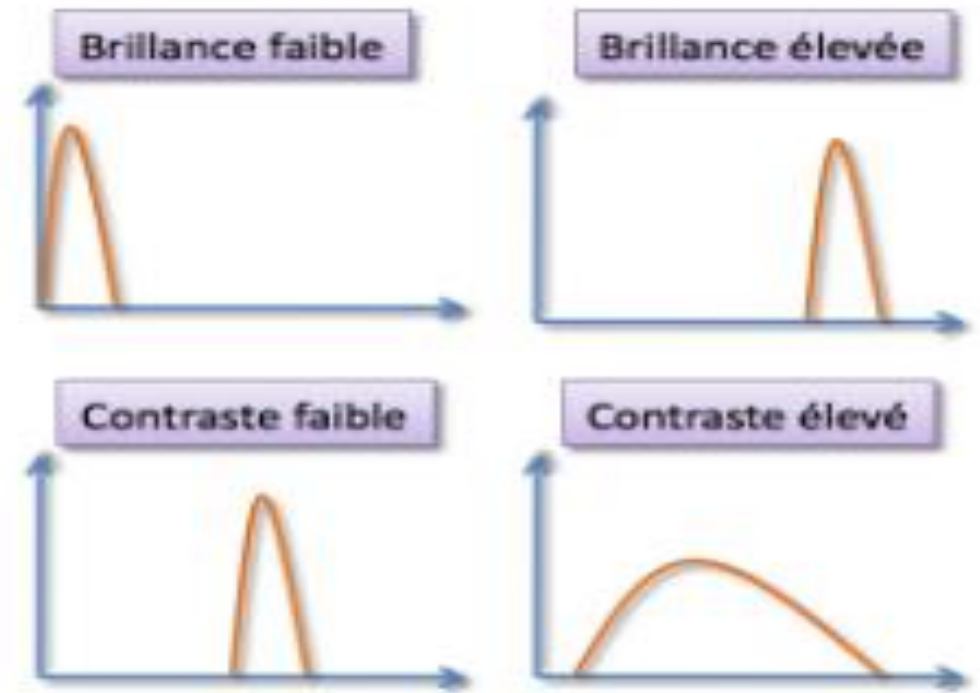
Histogramme d'images: Interprétation (5)

Question: Quelles sont les images qui ont le même histogramme?



Histogramme d'images: Interprétation (6)

- L'allure de h peut traduire des images trop sombres, trop claires ou encore mal contrastées.
- On peut agir sur la forme de l'histogramme
 - Transformation/modification (linéaire / par morceaux)
 - Egalisation



Deux images différentes (en termes de contenu sémantique) peuvent aussi avoir le même histogramme

Histogramme d'images: normalisé

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

$$h_n(i) = \frac{h(i)}{M \times N} \text{ avec } h_n(i) \in [0,1]$$

$h_n(i)$ = histogramme normalisé

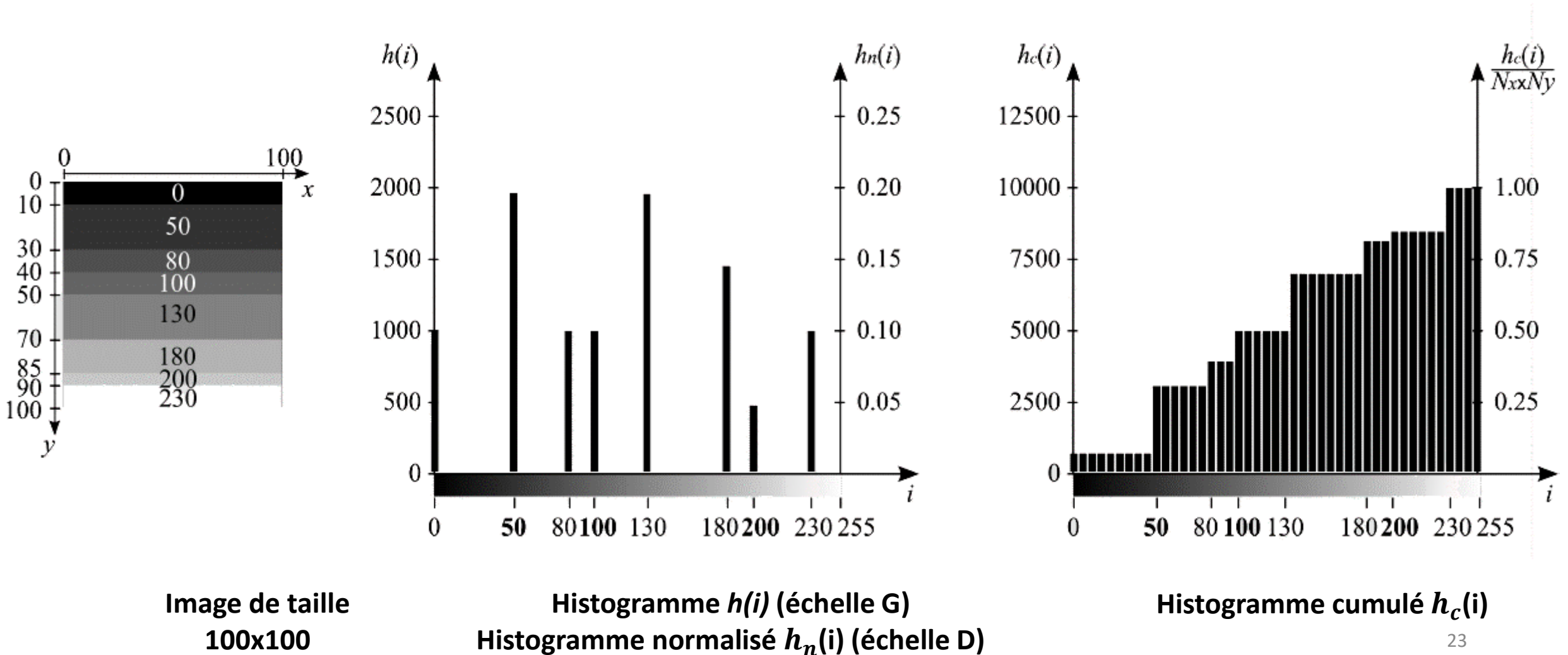
Histogramme d'images: cumulé

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

$$h_c(i) = \sum_{k=0}^i h(k) \quad \text{avec} \quad h_c(i) \in [0, M \times N]$$

- 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
$$\frac{h_c(i)}{M \times N} \quad \text{avec} \quad \frac{h_c(255)}{M \times N} = 1$$
 pour une image codée sur 1 octet

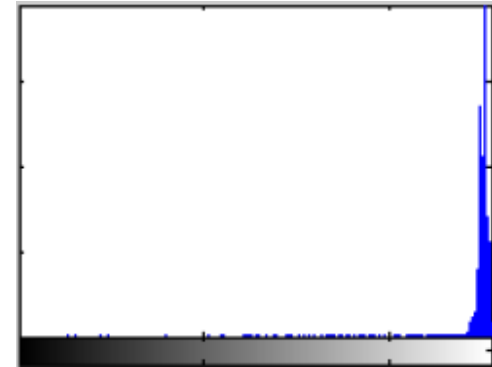
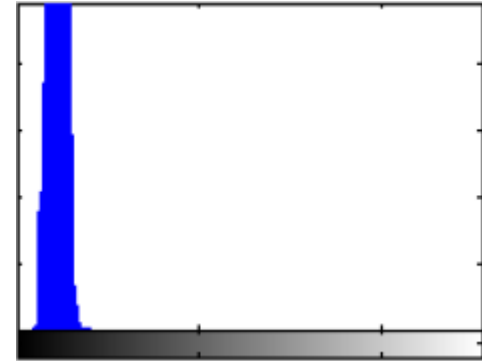
Histogramme d'images: normalisé et cumulé



Revenons à notre problème du départ!

- Des images trop claires ou trop foncées
- D'une manière générale: l'histogramme est trop concentré

Solution: application des méthodes ponctuelles travaillant sur les niveaux de gris ou sur les histogrammes, en général, ne modifiant pas l'information contenue dans les images



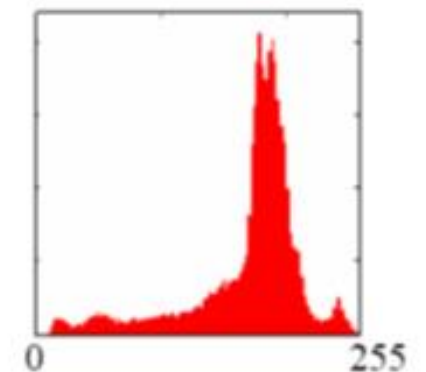
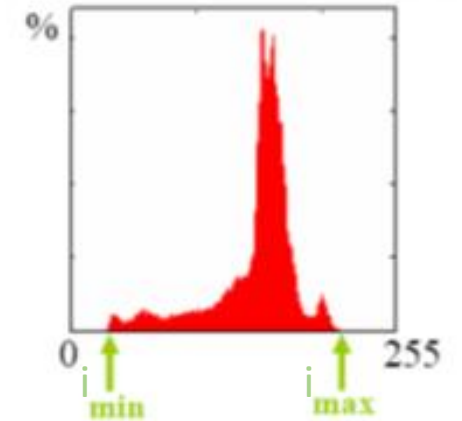
Transformation linéaire simple

- Soit $[i_{min}, i_{max}]$ la dynamique de l'image
- Fonction de la transformation est:

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

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

- Effet:
 - rehaussement du contraste par **expansion de la dynamique**



Expansion dynamique : Exemple Matlab



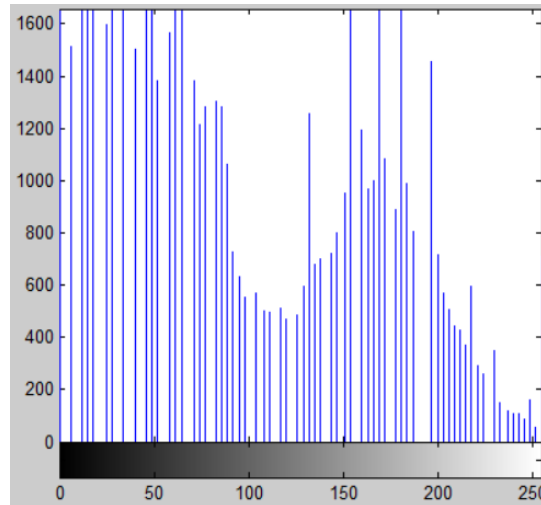
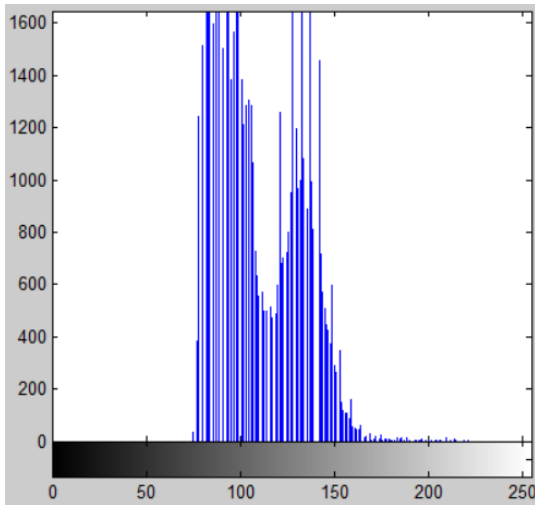
En Matlab :

```
I = imread('pout.tif');
```

```
J = imadjust(I);
```

```
figure, subplot(1,2,1), imshow(I); subplot(1,2,2), imshow(J);
```

```
figure, subplot(1,2,1), imhist(I); subplot(1,2,2), imhist(J);
```



Limite de la transformation linéaire

Si la dynamique est déjà maximale, la transformation n'apporte aucun changement
 $[i_{min} = 0, i_{max} = 255]$



Après
transformation



Transformation linéaire avec saturation

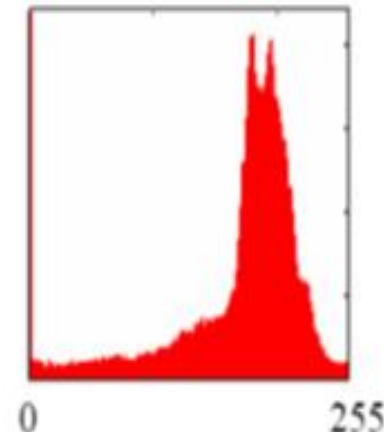
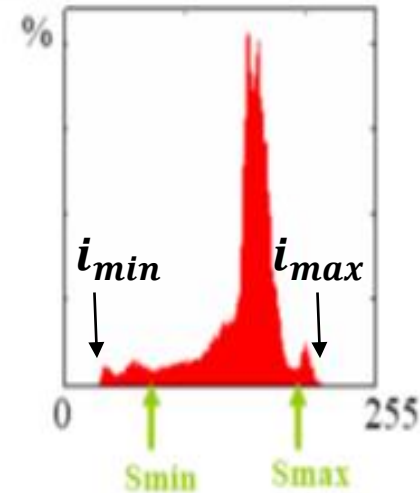
Définir deux seuils S_{max} et S_{min}

Tel que: $i_{min} < S_{min} < S_{max} < i_{max}$

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

Effets:

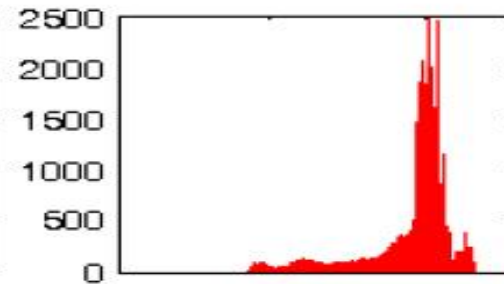
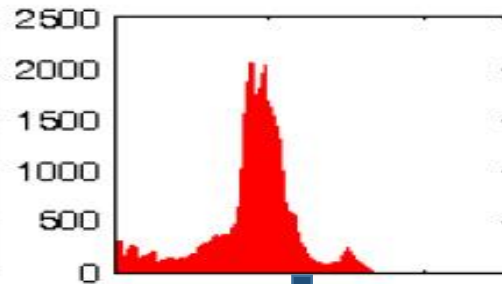
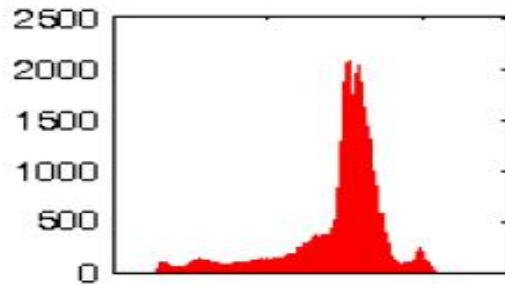
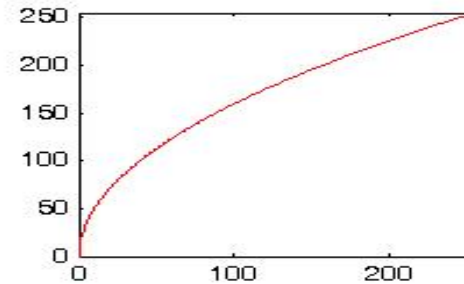
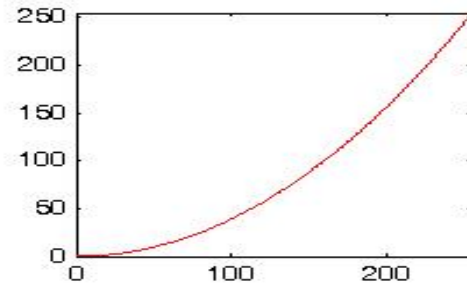
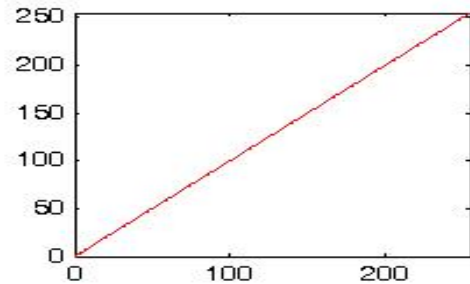
- **rehaussement du contraste** pour $S_{min} < i < S_{max}$
- **saturation**
 - à 0 pour $i_{min} < i < S_{min}$
 - à 255 pour $S_{max} < i < i_{max}$



Transformation non-linéaire



$x' = \text{Log}(x)$ pour
l'éclaircissement global de
l'image: utilisé pour traiter
des images trop sombres

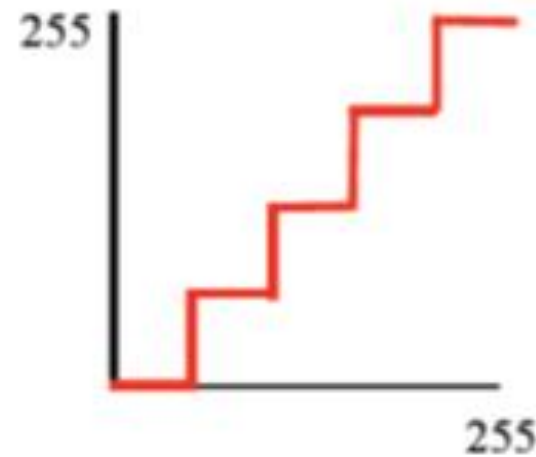


$x' = \exp(x)$ pour l'assombrissement global de
l'image: utilisé pour traiter des images trop claires

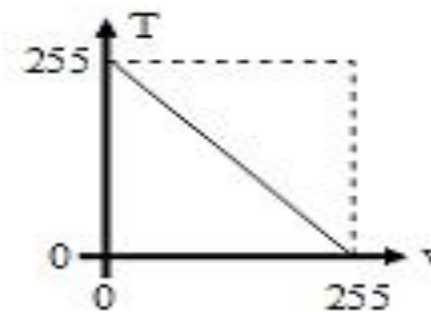
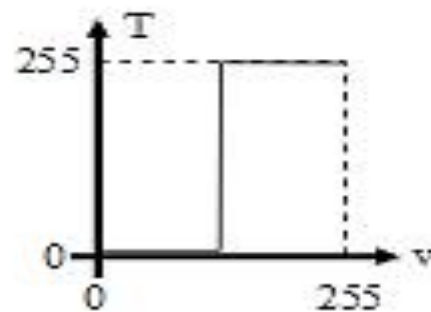
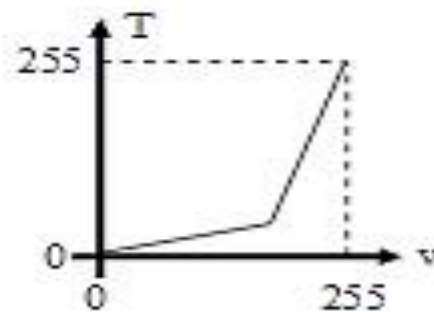
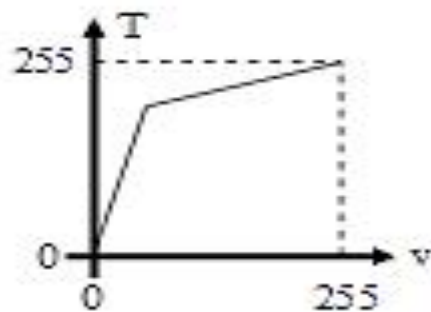
Autres transformations (1)

Quantification

- Transformation linéaire par morceaux utilisant des paliers
- Exemple: paliers de mêmes largeurs et de hauteurs réparties uniformément
- Résultat: seuls les niveaux de ces paliers sont conservés dans le résultat



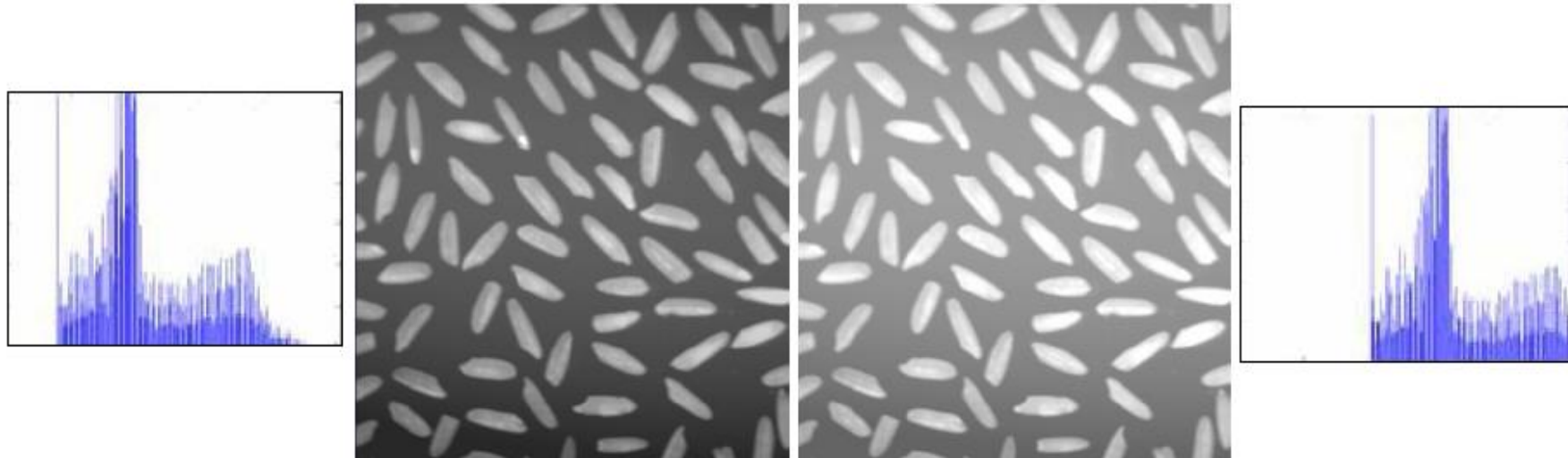
Autres transformations (2)



Translation de l'histogramme

- Permet de faire varier la luminosité de l'image sans en changer le contraste

$$I' = I + t \text{ avec } t \in \mathbb{R}$$



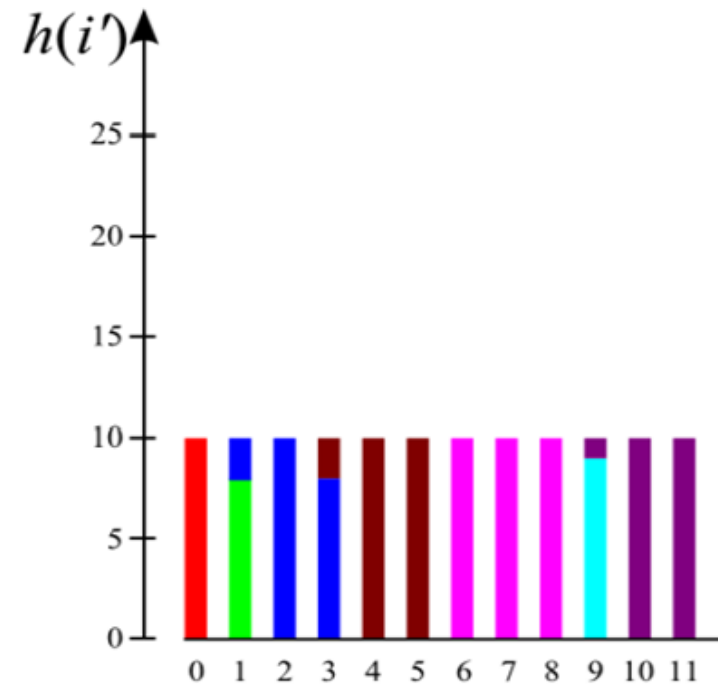
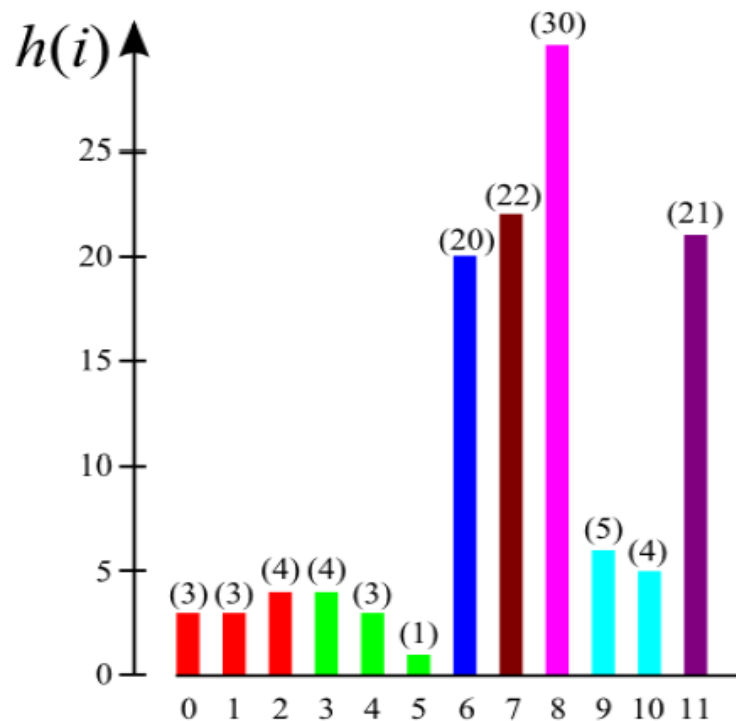
Egalisation d'histogramme

- On cherche à obtenir une image où les niveaux de gris sont répartis de manière la plus égalitaire possible (contraste maximal)
- Homogénéisation de la répartition des intensités des pixels
- Amplification des fluctuations dans les zones où elles sont faibles
- Etalement des détails concentrés dans un petit intervalle de niveaux de gris

Egalisation d'histogramme

On cherche une fonction de transformation $t : i \mapsto i'$

- 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



Egalisation d'histogramme

- Calcul de l'histogramme $h_n(k)$ avec $k \in [0,255]$

- Histogramme cumulé normalisé:

$$h_c(k) = \sum_{i=0}^k h_n(i)$$

- Transformation des niveaux de gris de l'image par:

$$I'(x, y) = h_c(I(x, y)) \times 255$$

- Avantages:

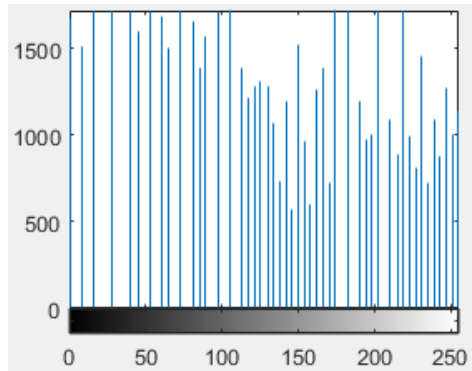
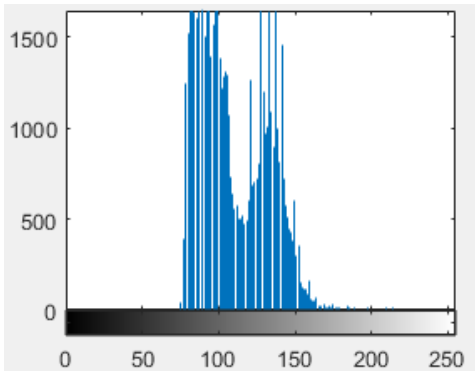
- Permet d'augmenter le contraste de l'image
- Pour chaque niveau de gris, il y a approximativement le même nombre de pixels.

Egalisation d'histogramme: Exemple

0 0 0 1 1 1 2 2 2 2
 3 3 3 3 4 4 6 6 6 6
 7 7 7 7 4 5 6 6 6 6
 7 7 7 7 7 7 6 6 6 6
 7 7 7 7 7 7 6 6 6 6
 7 7 7 7 7 7 6 6 6 6
 8 8 8 8 8 9 9 9 9 9
 8 8 8 8 8 8 10 10 10 10
 8 8 8 8 8 8 11 11 11 11
 8 8 8 8 8 8 11 11 11 11
 8 8 8 8 11 11 11 11 11 11
 8 8 8 8 11 11 11 11 11 11

x	histogramme	Histogramme cumulé h_c	h_c normalisé	Nouvelle valeur	Valeur entière
0	3	3	0.03	0.3	0
1	3	6	0.05	0.6	1
2	4	10	0.08	1	1
3	4	14	0.12	1.4	1
4	3	17	0.14	1.7	2
5	1	18	0.15	1.8	2
6	20	38	0.32	3.8	4
7	22	60	0.50	6	6
8	31	91	0.76	9.1	9
9	5	96	0.80	9.6	10
10	4	100	0.83	10	10
11	20	120	1	12	12

Egalisation d'histogramme: Exemple Matlab



En Matlab :

```
I = imread('pout.tif');
```

```
J = histeq(I);
```

```
figure, subplot(1,2,1), imshow(I); subplot(1,2,2), imshow(J);
```

```
figure, subplot(1,2,1), imhist(I); subplot(1,2,2), imhist(J);
```

Egalisation d'histogramme: Exemple



Image originale

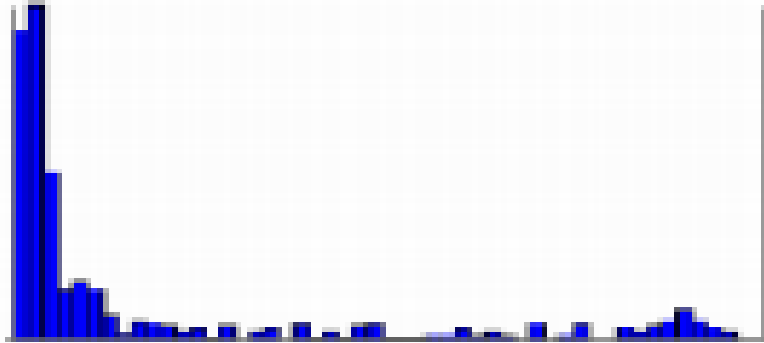


Expansion dynamique

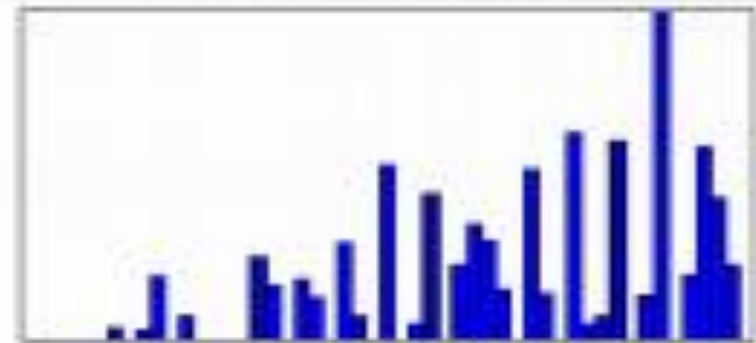


Egalisation d'histogramme

Egalisation d'histogramme: Exemple



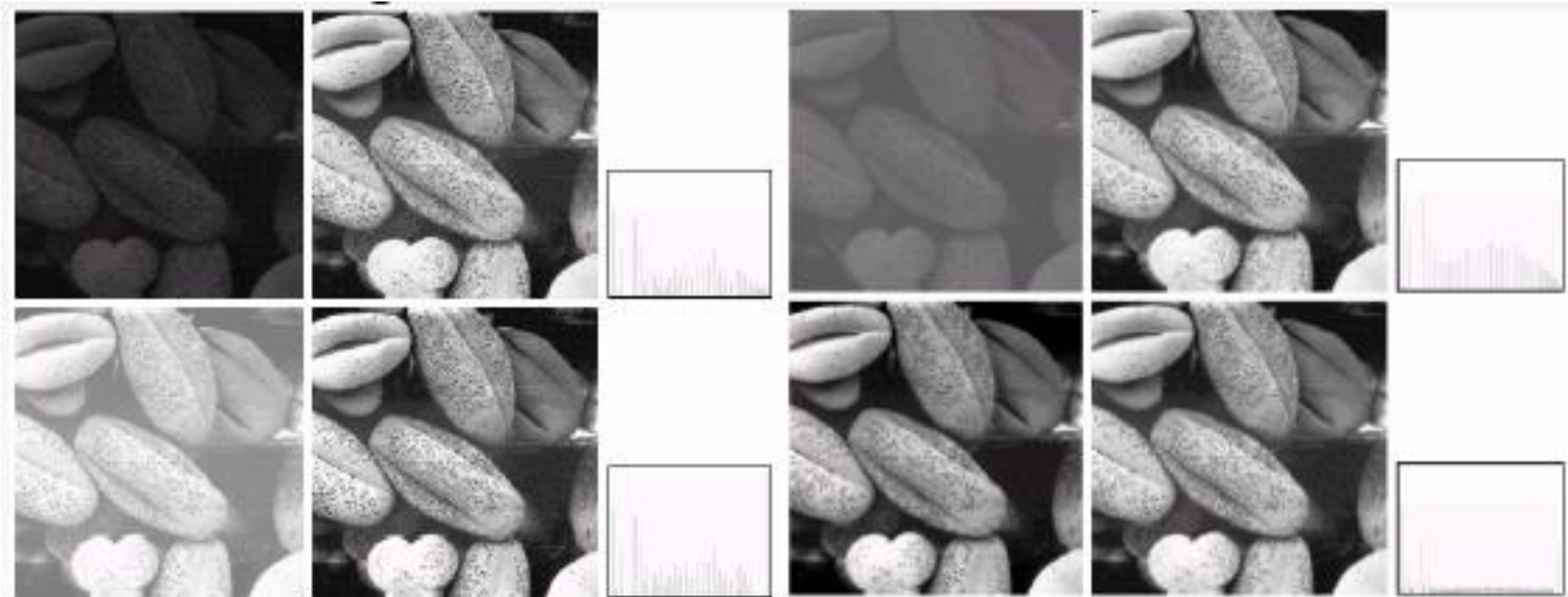
Expansion d'histogramme!



Egalisation d'histogramme

Intérêt d'égalisation d'histogramme (1)

L'égalisation d'histogramme sur une même image avec des **contrastes différents**, donne le même résultat pour toutes les images.



Intérêt d'égalisation d'histogramme (2)

- Un problème des variations d'illumination au sein de la base
- Normaliser l'ensemble des histogrammes pour que les images aient la même dynamique

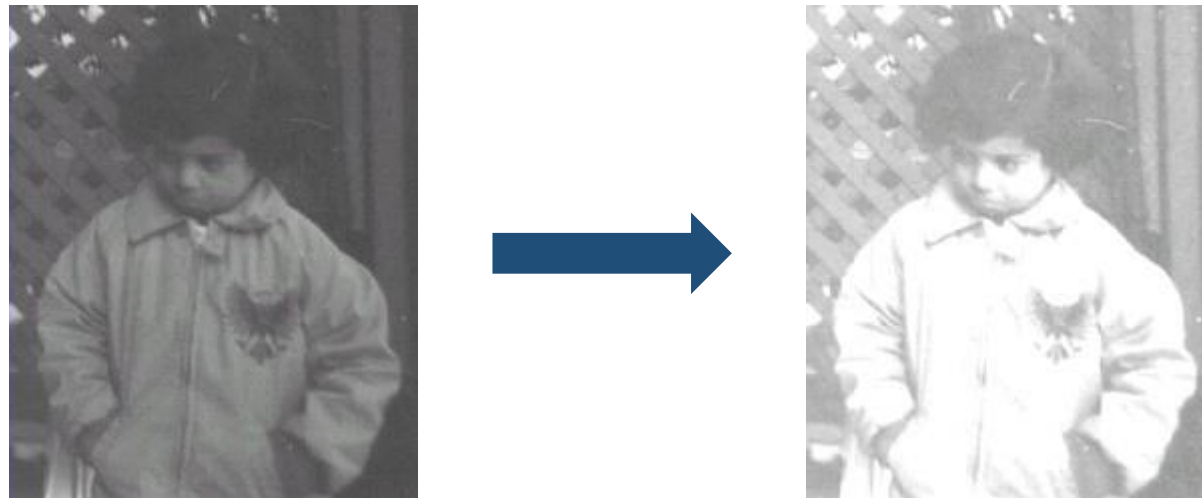


Opérations arithmétiques: Addition

- Soient f et g sont deux images, on peut définir l'addition R pixel à pixel de ces deux images par:

$$R(x, y) = f(x, y) + g(x, y)$$

- L'addition d'images peut permettre:
 - De diminuer le bruit d'une vue dans une série d'images
 - D'augmenter la luminance en additionnant une image avec elle-même



Opérations arithmétiques: Soustraction

- On peut définir la soustraction S pixel à pixel de deux images f et g par:

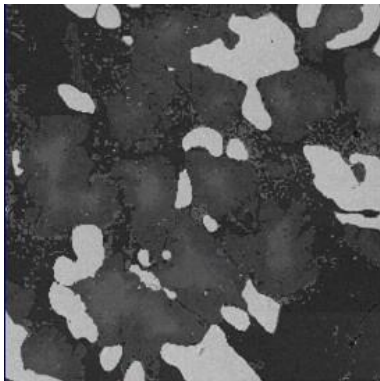
$$S(x, y) = f(x, y) - g(x, y)$$

- La soustraction d'images peut permettre:
 - Détection de défauts (par comparaison avec une image de référence)
 - Détection de mouvements

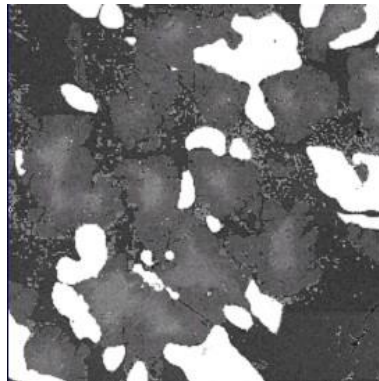


Opérations arithmétiques: Multiplication

- La multiplication d'une image f par un ratio (facteur) peut se définir par:
- Permet d'améliorer le contraste ou la luminosité



x 1.5



x 1.2



Opérations arithmétiques: Division

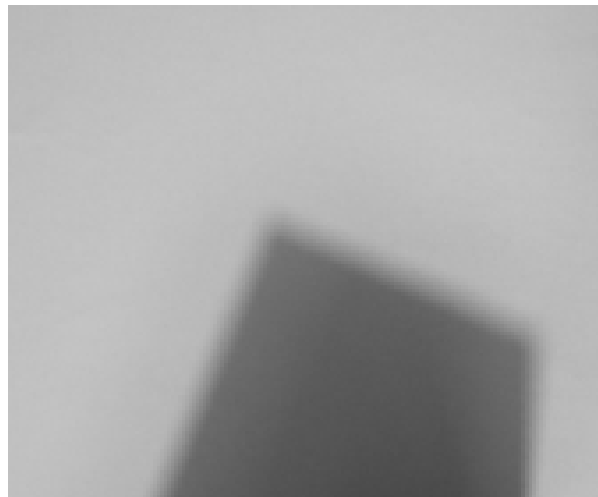
- La division de deux images permet de corriger une illumination non homogène:

$$D(x, y) = f(x, y) / g(x, y)$$

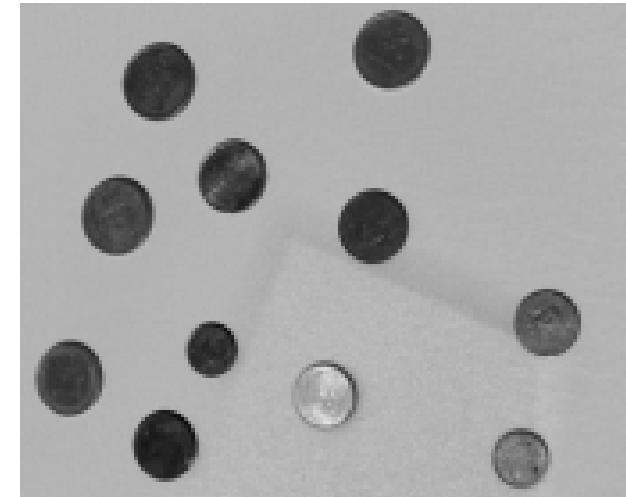
- Exemple d'application: la suppression de l'ombre sur une image



$f(x,y)$



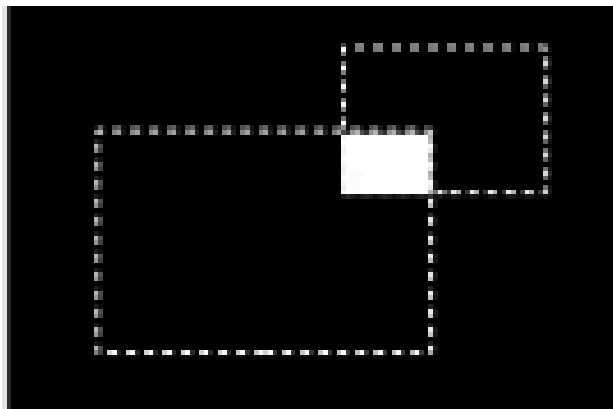
$g(x,y)$



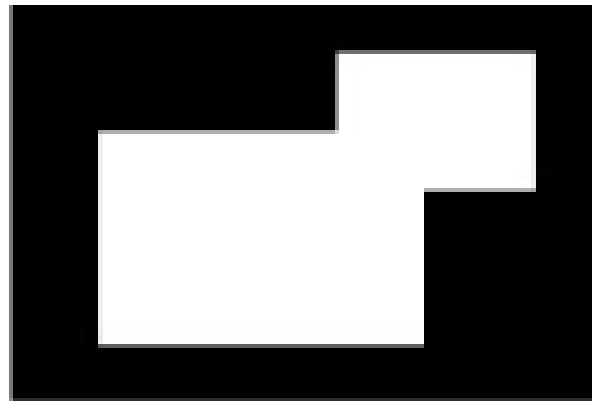
$D(x,y)$

Opérations logiques: ET/OU/XOR

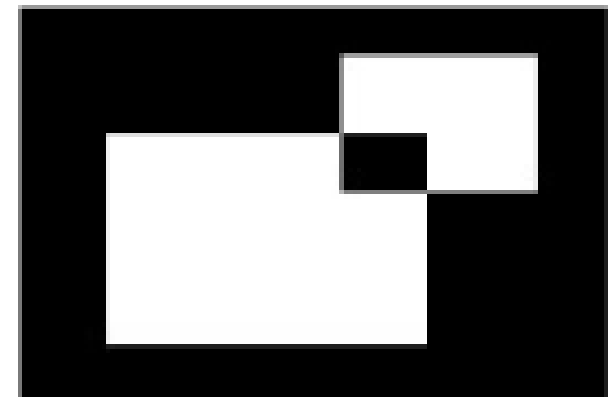
Nous avons les deux images binaires I1 et I2



ET logique: intersection



OU logique: union



OU exclusif logique: exclusion

Opérations logiques: ET/OU/XOR

Application: Masquage



Image 1

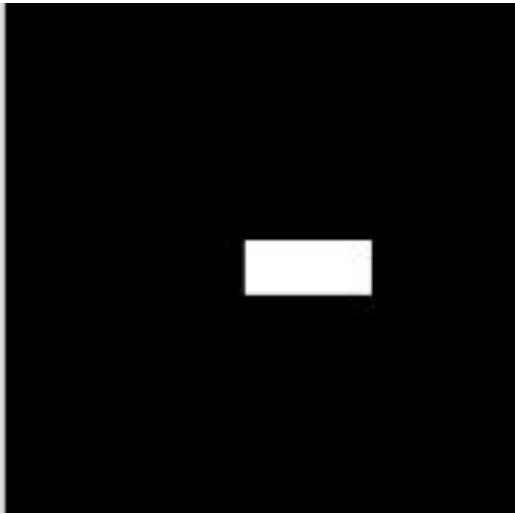
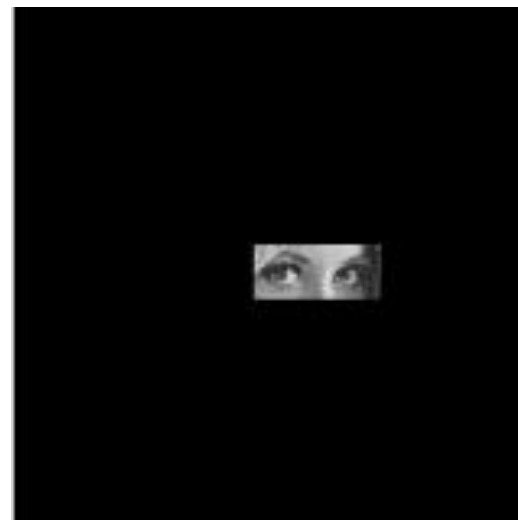


Image 2 (masque)



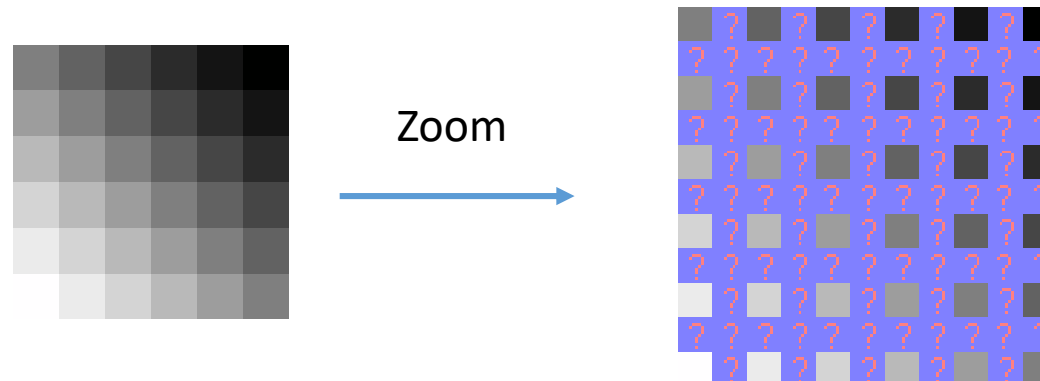
ET (Image 1, Image 2)



OU (Image 1, Image 2)

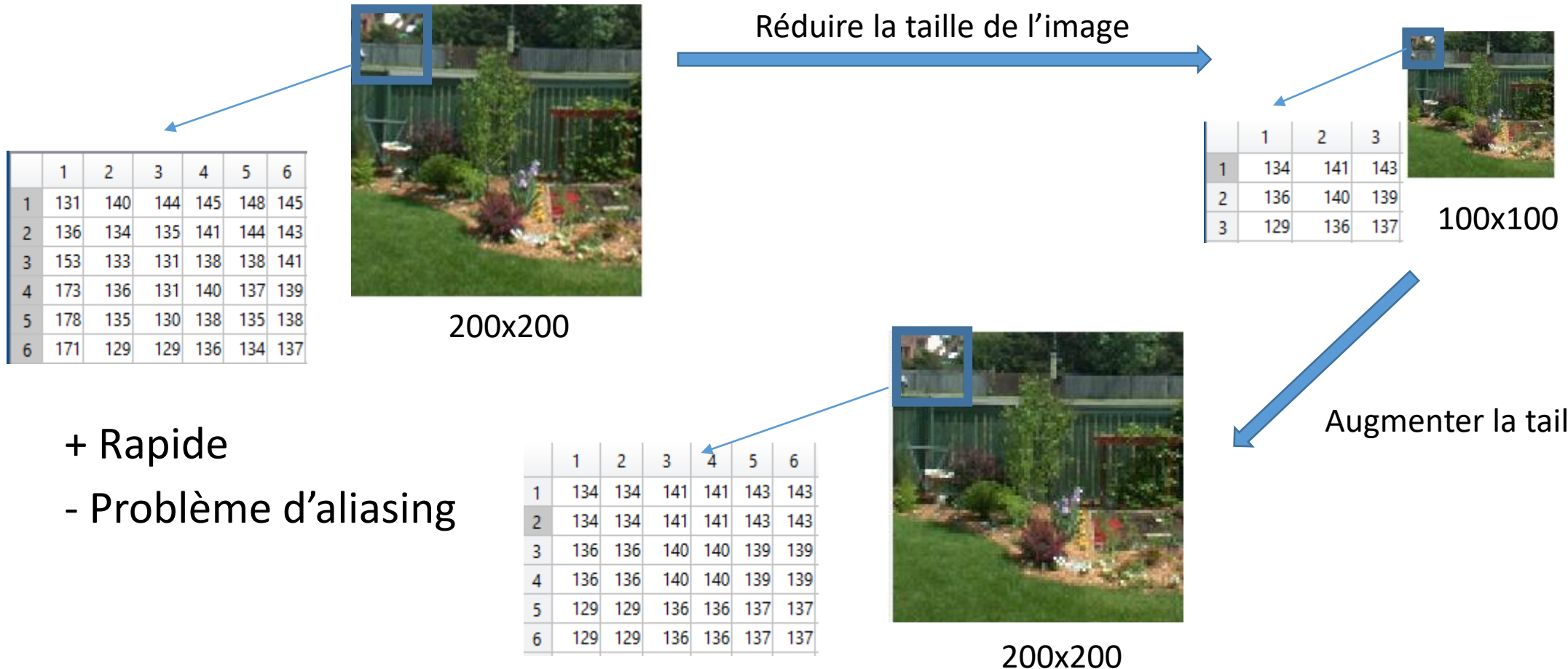
Réduire/ Augmenter la taille de l'image

- Plusieurs méthodes pour augmenter ou diminuer la taille d'une image:
 - Interpolation au plus proche voisin
 - Interpolation bilinéaire
 - Interpolation bicubique



Interpolation au plus proche voisin

- Le pixel aura la valeur de son plus proche voisin

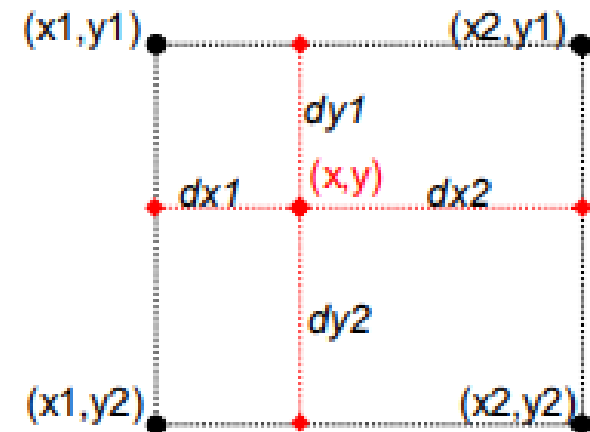


Interpolation bilinéaire

- Le pixel aura la valeur suivante :
- $f(x, y) = dy1(dx1.f(x1, y1) + dx2.f(x2, y1)) + dy2(dx1.f(x1, y2) + dx2.f(x2, y2))$

+ relativement rapide

- images floues



Interpolation bicubique

- Repose sur le même principe que l'interpolation bilinéaire
 - Pour calculer les pixels interpolés, elle utilise les 16 pixels voisins au lieu de 4
- + Peu de flou
- le temps de calcul est plus lent

Interpolation bilinéaire



200x200



50x50

Interpolation au plus proche voisin



50x50

Interpolation bilinéaire



50x50

Interpolation bicubique

Le code en Matlab :

```
A = imread('pic_org.jpg');  
lppv= imresize(A,0.5,'nearest');  
lbilineaire = imresize(A, 0.5,'bilinear');  
lbicubic = imresize(A, 0.5,'bicubic');
```

Objectifs de la transformation géométrique 2D

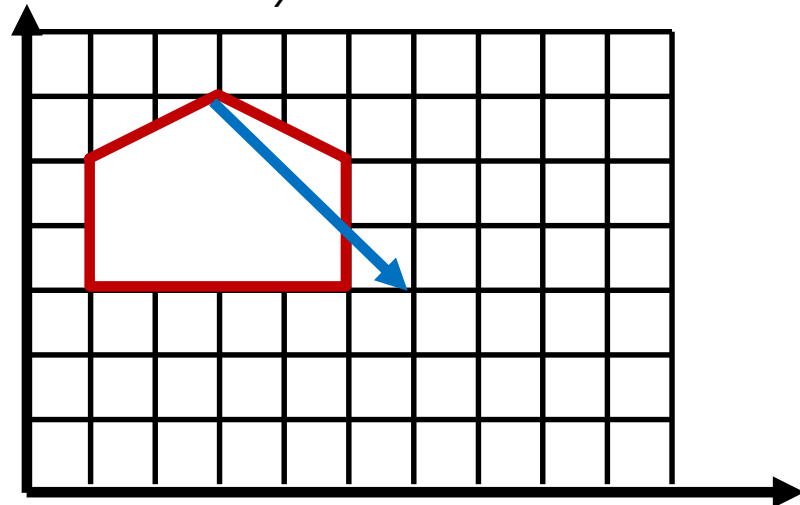
- Aligner une image scannée sur l'image de référence
 - Translation et rotation pour corriger le mauvais alignement
- Stabiliser les images d'une séquence vidéo
 - Corriger le mouvement parasite dû au bougé



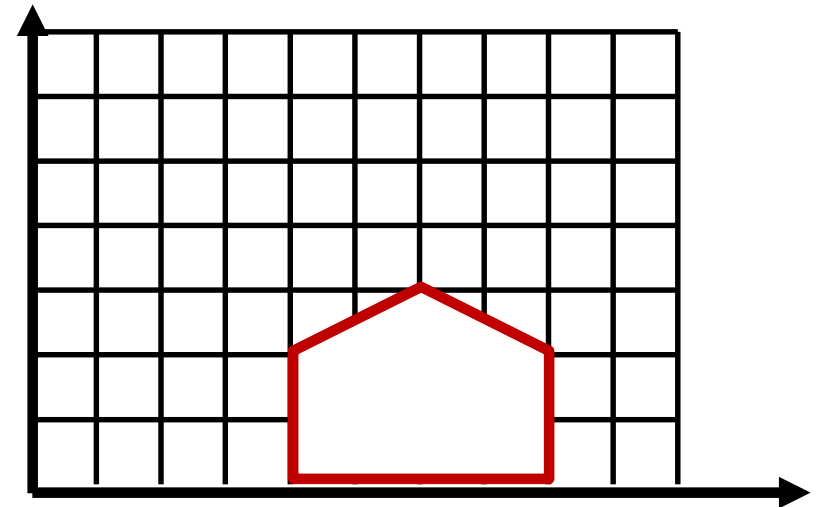
Translation

Transformation directe sur les coordonnées spatiales d'un pixel
exprimée de manière générale par:

- $x' = x + t_x$
- $y' = y + t_y$



Avant



Après

Translation

La translation d'un pixel (x, y) de vecteur (t_x, t_y) s'exprime:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$



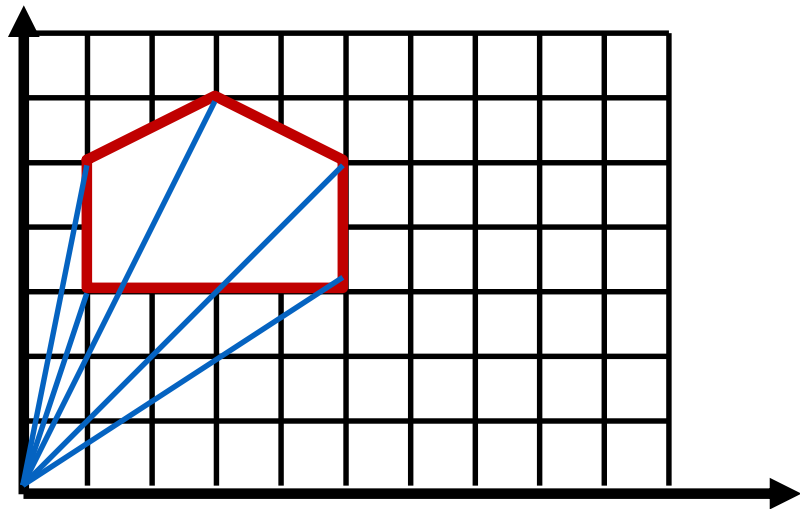
Code Matlab:

```
dx = 20;  
dy = 10;  
I = imread('cameraman.tif');  
T = imtranslate(I, [dx dy]);  
imshow(I), figure, imshow(T)
```

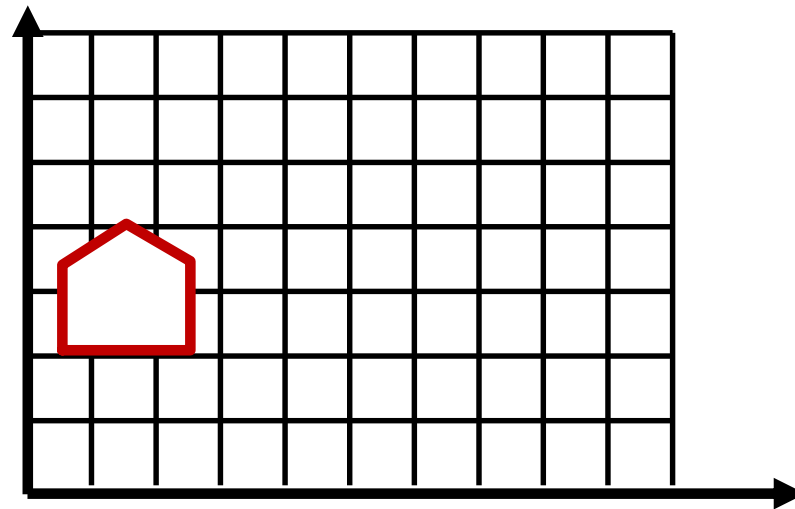
Homothétie

Les coordonnées sont multipliées par un facteur d'échelle selon l'axe des x et l'axe des y à partir d'une origine donnée

- $x' = s_x x$
- $y' = s_y y$



Avant



Après

Homothétie

Une homothétie de rapport s_x et s_y par rapport à l'origine s'exprime:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$



Code Matlab:

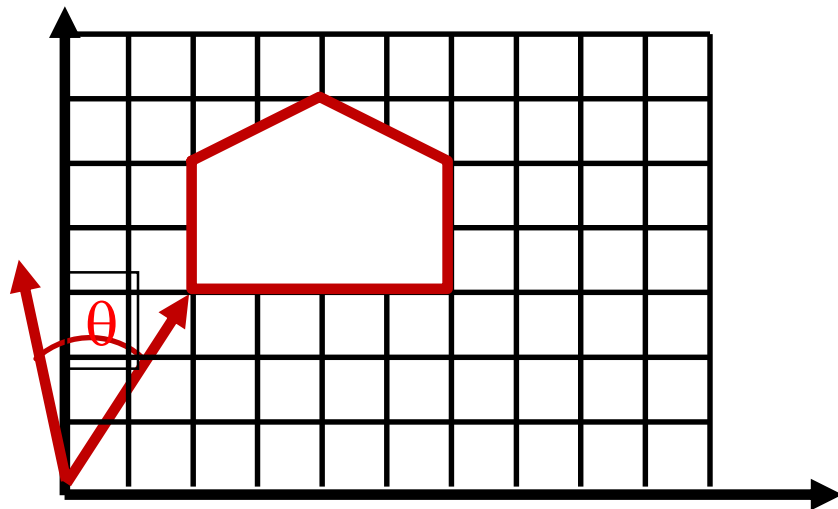
```
I = imread('cameraman.tif');  
H = imresize(I, 0.25);  
imshow(I), figure, imshow(H)
```

Rotation (1)

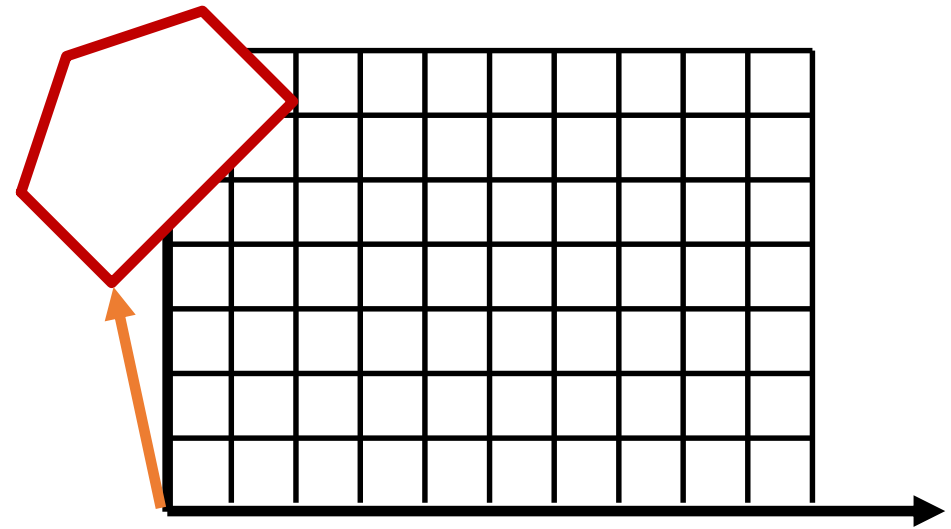
Une rotation d'un angle θ par rapport à l'origine s'exprime:

- $x' = \cos\theta x - \sin\theta y$
- $y' = \sin\theta x + \cos\theta y$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$



Avant



Après

Rotation (2)



Rotation de 25°
Plus proche voisin



Rotation de 25°
bilinéaire



Rotation de 25°
bicubique

Code Matlab :

```
angle = 25;  
I = imread('cameraman.tif');  
ppv=imrotate(I,angle,'nearest');  
bilinear=imrotate(I,angle,'bilinear');  
bicubic=imrotate(I,angle,'bicubic');
```

Rotation et translation

Les nouvelles coordonnées sont calculées comme suit

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$

Code Matlab:

```
I = imread('cameraman.tif');
```

```
J=imrotate(I,90);
```

```
imshow(J);
```

```
dx = 20;
```

```
dy = 10;
```

```
H=imtranslate(J, [dx dy]);
```

```
imshow(H);
```



Rappel: Opérations sur une image

- Chaque pixel de l'image est défini par sa position (x, y) et son intensité $I(x, y)$ dans l'image
- Il existe deux types de transformations sur les pixels de l'image:
 - **radiométriques** qui modifient les intensités des pixels
 - **géométriques** qui modifient les positions des pixels

TP1: Traitements basiques d'images sous MATLAB

Le but de ce TP est d'appliquer les traitements qui permettent d'améliorer la qualité de l'image

- Algorithmes classiques de traitement
 - recadrage dynamique,
 - égalisation d'histogramme,
 - transformation géométrique.