

Introduction au Traitement d'Images

Séance 5

Isabelle Debled-Rennesson
debled@loria.fr

Contenu

- 6 cours/TD/TP de 2h
 - Examen le 27 mai → **résumé manuscrit de 2 feuilles autorisé**
- Points abordés :
 - Codage des images
 - Histogrammes
 - Transformations géométriques sur les images
 - **Filtres**
 - Opérateurs morphologiques
 - **Détection de contours**
 - Segmentation

Correction TP S2

Q1

- Plugin calculant la mesure de contraste de l'image traitée

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

- Valeurs à obtenir :

- Image « mystere »: 0.432145...
- Image « boat »: 49.663829...
- Image « baboon »: 1.068963...

Q2

- Etirement linéaire des images – utilisation d'une LUT

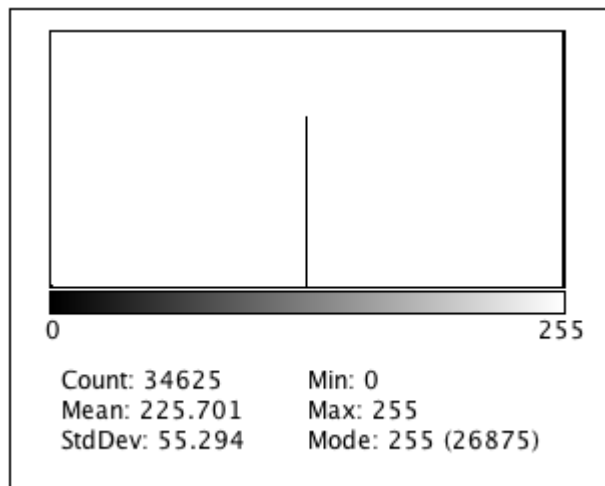
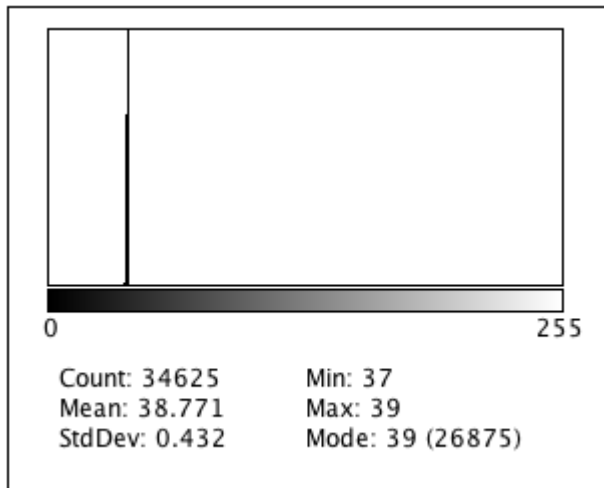
```
for(int ng = 0; ng < 256; ng++)  
    LUT[ng] = (int) 255*(ng-min)/(max-min);  
  
for (int i=0;i<N;i++)  
    ip.set(i,LUT[(pixels[i] & 0xff)]);
```

Q2

- Etirement linéaire des images – utilisation d'une LUT

```
for(int ng = 0; ng < 256; ng++)  
    LUT[ng] = (int) 255*(ng-min)/(max-min);  
  
ip.applyTable(LUT);
```

Q2



Sur l'image mystère

Avant transformation, 3 niveaux
37,38,39



Après étirement, 3 niveaux 0, 127 et
255 – contraste =55.29...

LA VIE
EST BELLE

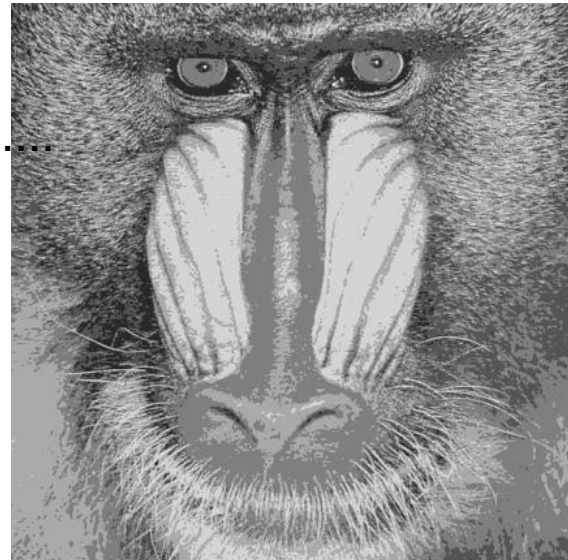
Q2



C=108.23..



C=45.41....



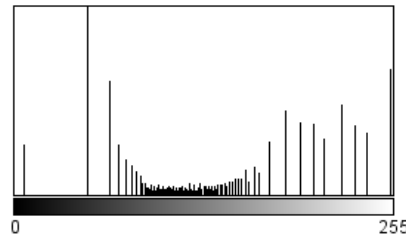
Q 3

- Egalisation d'histogramme
 - $I'(x,y) = (N-1) * R_I(I(x,y))$
 - Avec $R_I(n) = \sum_{k=0}^n PI(k)$
 - Calcul de l'histogramme normalisé
 - Calcul d'une LUT pour l'égalisation

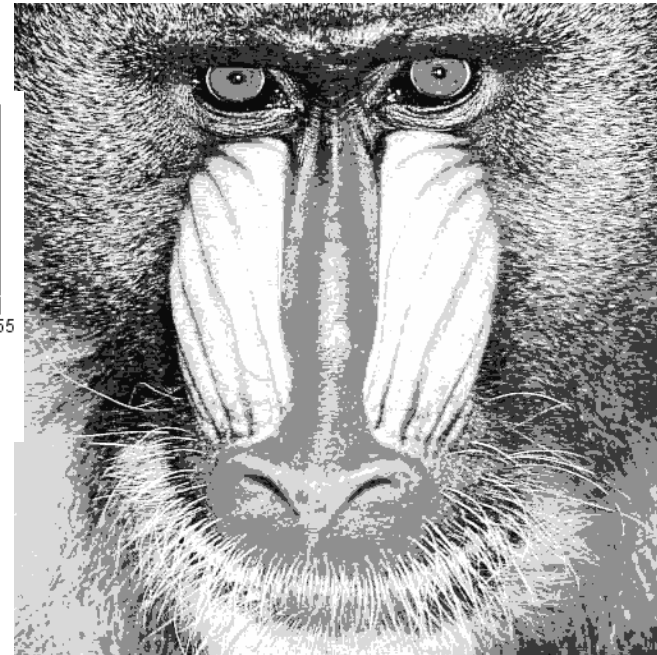
Q3 – sur les images

C=70.327..

..

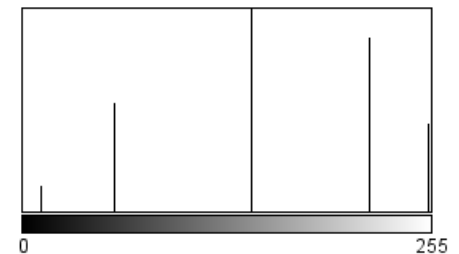


Count: 211848
Mean: 133.401
StdDev: 70.327
Min: 0
Max: 254
Mode: 49 (35619)



C=72.468

....



Count: 262144
Mean: 159.277
StdDev: 72.468
Min: 0
Max: 255
Mode: 143 (88787)

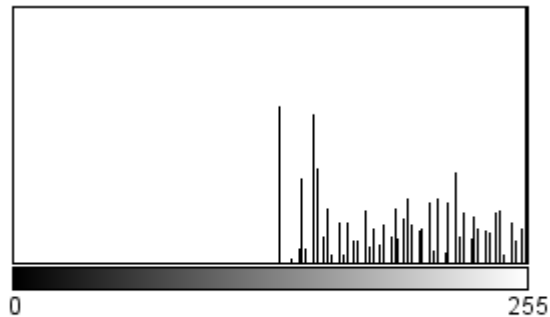
LA VIE
EST BELLE



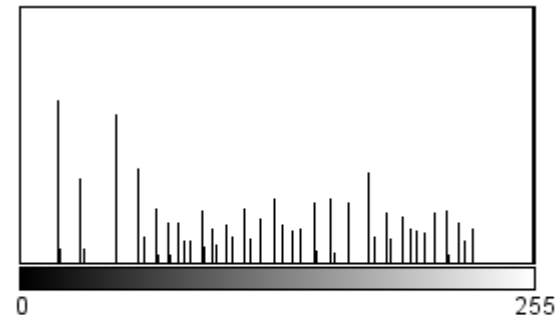
Count: 34625
Mean: 210.396
StdDev: 83.156
Min: 1
Max: 255
Mode: 255 (26875)

C=83.156..

Q3 – sur les images

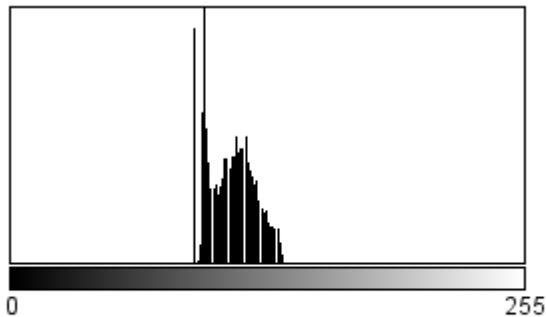
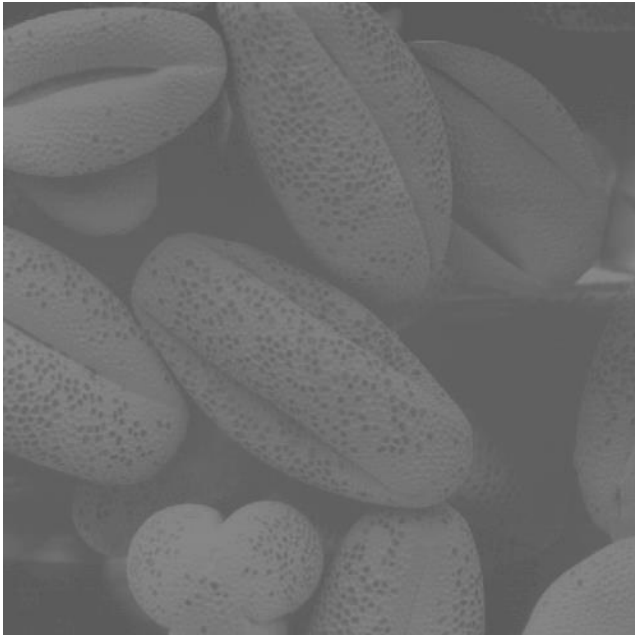


Count: 247506
Mean: 196.293
StdDev: 40.150
Min: 132
Max: 255
Mode: 255 (28035)

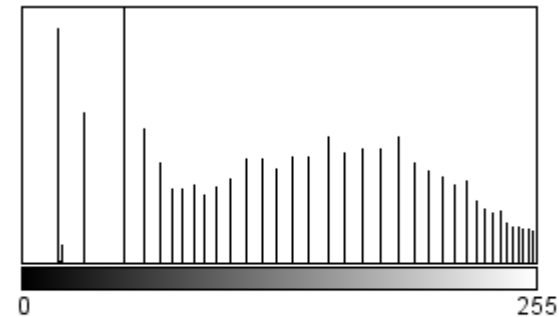
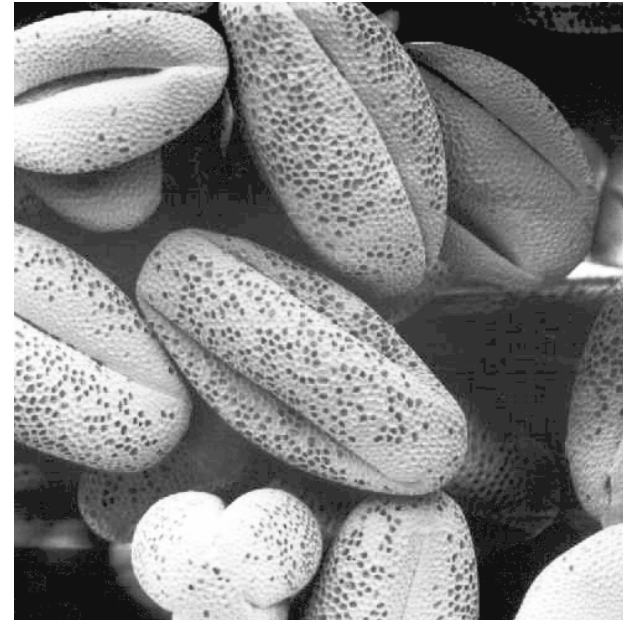


Count: 247506
Mean: 131.398
StdDev: 74.425
Min: 18
Max: 255
Mode: 255 (28035)

Q3 – sur les images

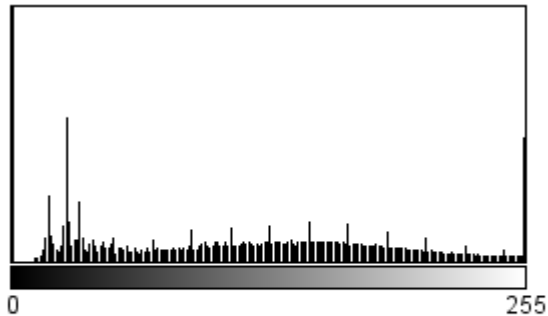


Count: 250000
Mean: 109.080
StdDev: 11.490
Min: 91
Max: 138
Mode: 96 (19078)

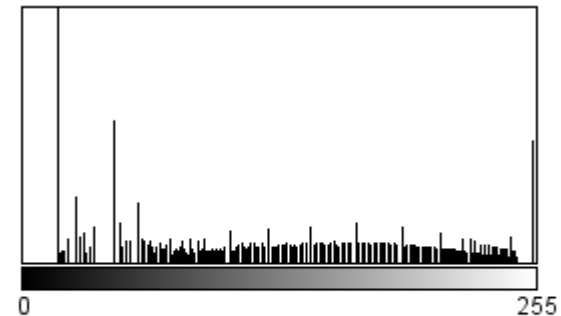
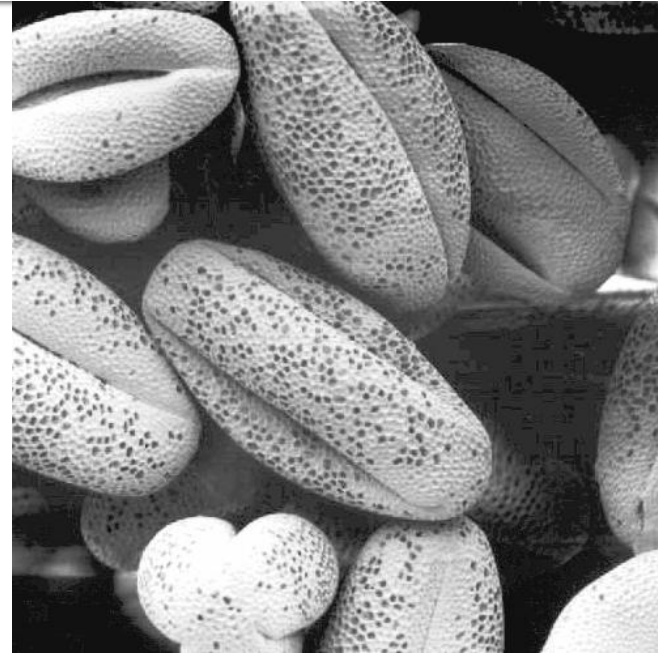


Count: 250000
Mean: 131.386
StdDev: 72.038
Min: 17
Max: 254
Mode: 50 (19078)

Q3 – sur les images

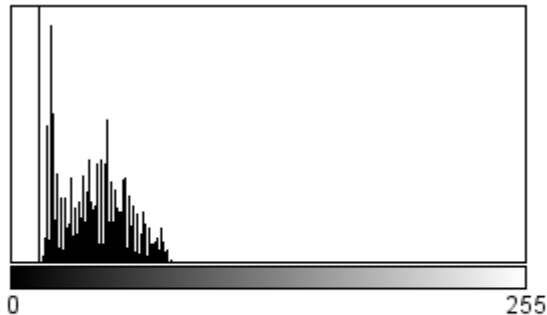
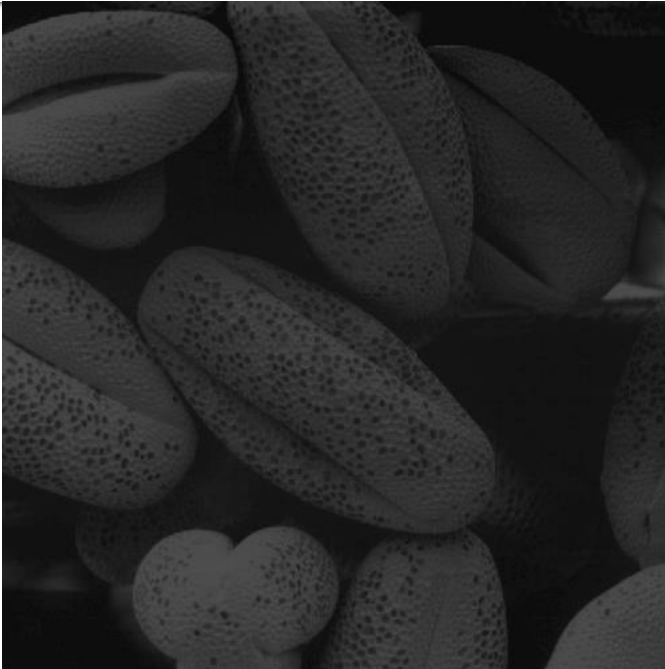


Count: 250000
Mean: 113.156
StdDev: 74.556
Min: 0
Max: 255
Mode: 0 (17496)

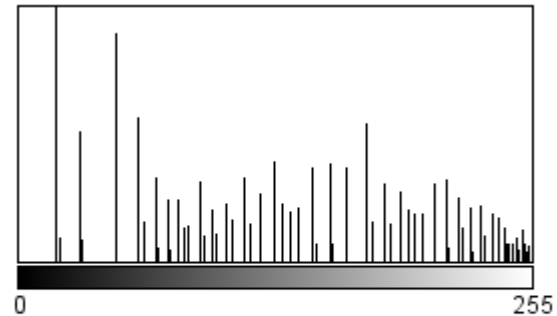
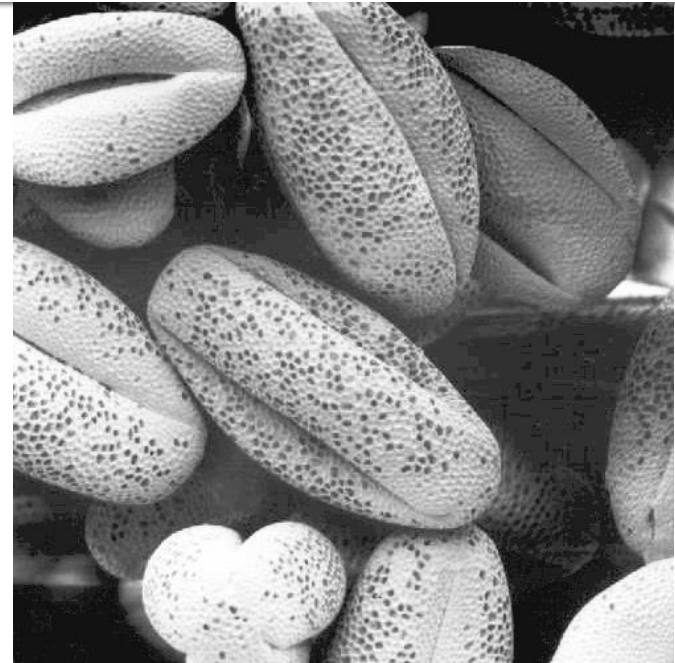


Count: 250000
Mean: 128.551
StdDev: 72.627
Min: 17
Max: 254
Mode: 17 (17608)

Q3 – sur les images



Count: 250000
Mean: 38.748
StdDev: 17.257
Min: 13
Max: 83
Mode: 13 (17718)



Count: 250000
Mean: 130.462
StdDev: 71.955
Min: 18
Max: 254
Mode: 18 (18320)

Q3 -ccl

- Les contrastes obtenus pour la série de 4 images sont entre 71.9 et 74.4. L'objectif de l'égalisation est d'obtenir un histogramme avec une bonne répartition des niveaux de gris qui implique un bon contraste.
- Les images résultantes obtenues semblent visuellement identiques.

Q4

- Même démarche que pour Q3 mais travailler sur l'image transformée en niveaux de gris
 - Une LUT d'égalisation est alors obtenue
 - Elle est appliquée sur chaque composante

Q4



Q4

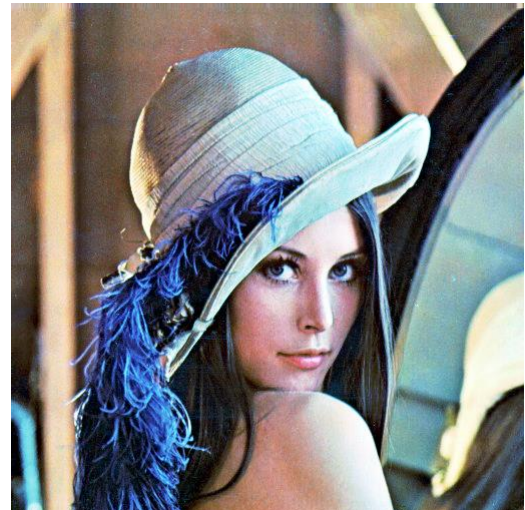


- Le contraste augmente en respectant la répartition des couleurs de l'image initiale

Q5

- Calculer une LUT par composant, chaque composant étant égalisé selon sa distribution.

Q5



Q5



- La répartition des couleurs de l'image initiale est modifiée en appliquant une égalisation différente sur chaque composante R, G, B.

Q6

```
for(int i=0; i<w*h;i++){  
    int imA=(pixelsDiffA[i]& 0xff);  
    int imB=(pixelsDiffB[i]& 0xff);  
    if(imA-imB >0)  
        pixelsRes[i]=(byte) (imA-imB);  
    else  
        pixelsRes[i]=0;  
}
```



Q7

```
for(int i=0; i<w*h;i++){  
    int imA=(pixelsDiffA[i]& 0xff);  
    int imB=(pixelsDiffB[i]& 0xff);  
    if(imB-imA !=0)  
        pixelsRes[i]=(byte) 250;  
    else  
        pixelsRes[i]=0;  
}
```

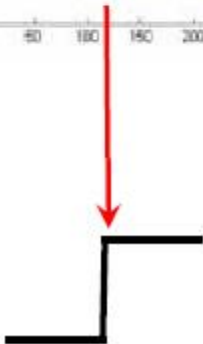
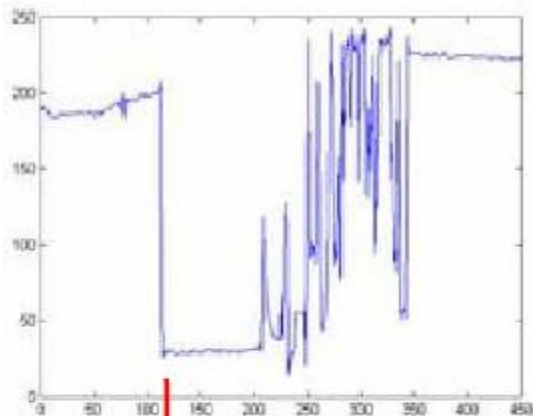


Détection de contours

Utilisation de filtres linéaires

Qu'est-ce qu'un contour ?

- Contour : frontière entre deux objets dans une image
- Définition plus large : discontinuité de l'image (variation brusque d'intensité).



Discontinuités de l'image

- Attention, toute discontinuité n'est pas nécessairement située à la frontière entre deux objets.

de profondeur



d'orientation
de surface

de réflectance

d'illumination

Différents types de contours



Marche d'escalier

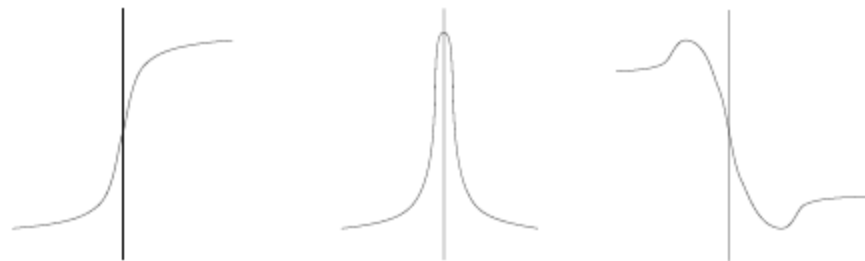
Rampe

Toit

Source : Gonzalez and Woods. Digital Image Processing 3ed. Prentice-Hall, 2008.

Détection de contours

- Détection de contours = Etape préliminaire à de nombreuses applications de l'analyse d'images.
 - Les contours constituent des indices riches, au même titre que les points d'intérêts, pour toute interprétation ultérieure de l'image.
- Les contours dans une image sont caractérisés par des **discontinuités de la fonction d'intensité** dans les images.
 - Le principe de la détection de contours repose donc sur **l'étude des dérivées de la fonction d'intensité** dans l'image : les extréma locaux du gradient de la fonction d'intensité et les passages par zéro du laplacien.



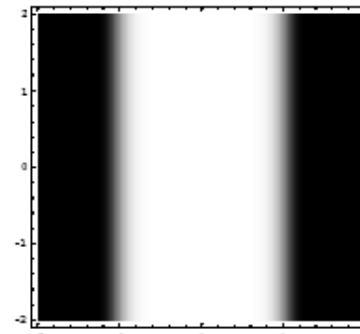
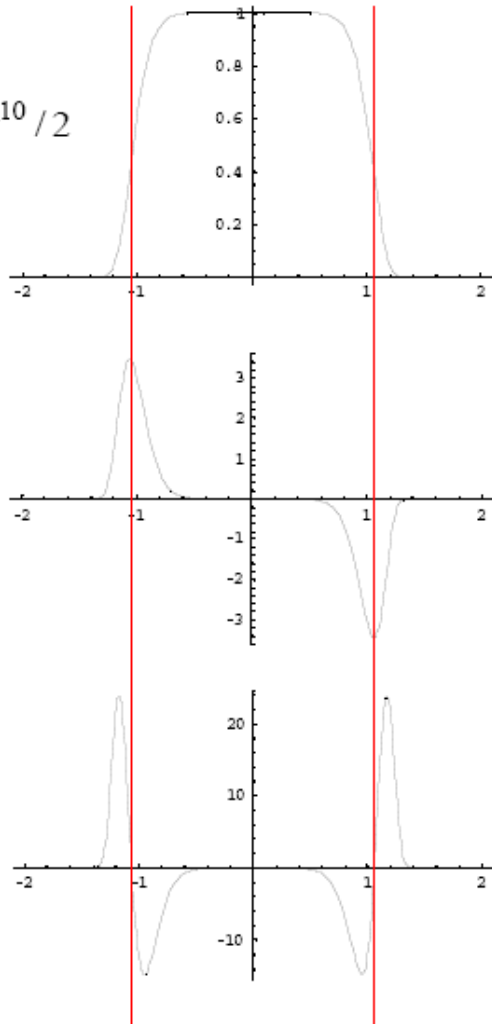
La fonction d'intensité au voisinage d'un contour en marche et ses dérivées première et seconde.

Dérivées d'une image et contours

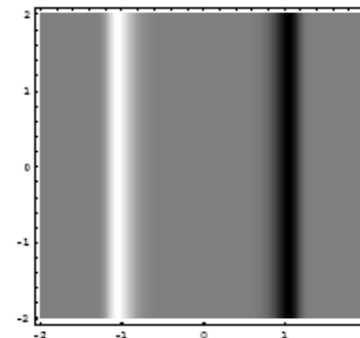
$$f(x, y) = e^{-x^{10}/2}$$

$$\frac{\partial f}{\partial x}$$

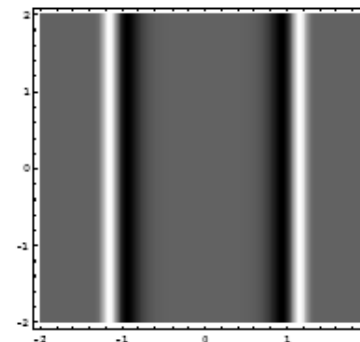
$$\frac{\partial^2 f}{\partial x^2}$$



Image



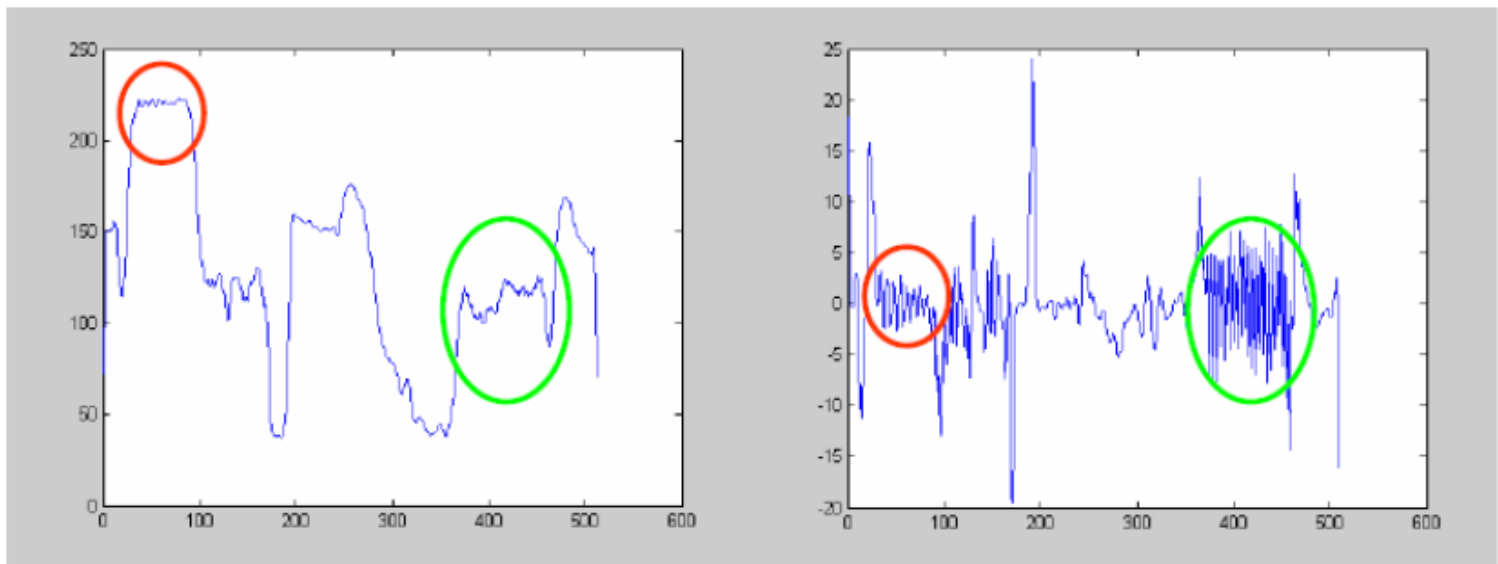
*Première
dérivée*



*Deuxième
dérivée*

Détection de contours - principe

- Etude des dérivées de la fonction d'intensité de l'image
 - les extréma locaux de la dérivée première
 - les passages par zéro de la dérivée seconde
 - Difficulté : la présence de bruit dans les images



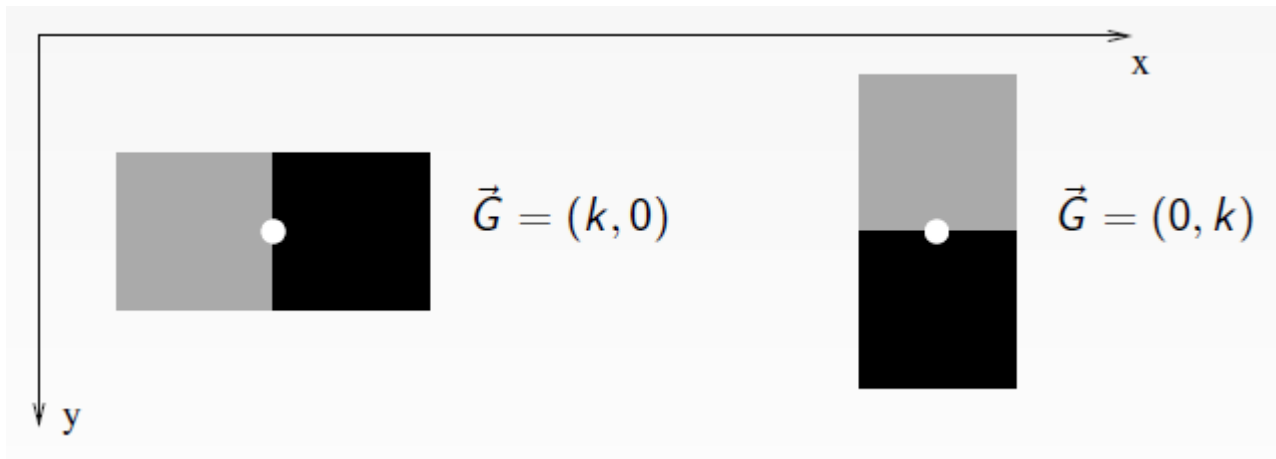
signal

dérivée première

Gradient d'une image

- Gradient d'une image ayant une fonction d'intensité I

- $\vec{G} = (G_x, G_y) = \left(\frac{\partial I(x,y)}{\partial x}, \frac{\partial I(x,y)}{\partial y} \right)$



Gradient d'une image

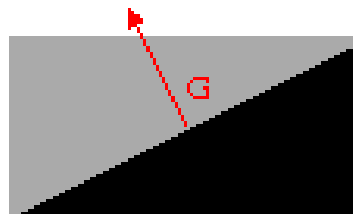
- Le gradient peut être représenté en coordonnées polaires par un **module m** et une **direction ϕ** dans l'image.

- le module du gradient mesure la force du contour

$$m = \sqrt{G_x^2 + G_y^2} \approx |G_x| + |G_y|$$

- le gradient est un vecteur perpendiculaire au contour

$$\phi = \arctan\left(\frac{G_x}{G_y}\right)$$



Approximation de la dérivée

- Une image est discrète par nature. Les premières approches ont donc consisté à approximer les dérivées par différences finies :

- $G_x(x, y) = I(x, y) - I(x - n, y)$

ou alors :

- $G_x(x, y) = I(x + n, y) - I(x - n, y)$

avec en général $n=1$.

Ces dérivées sont calculées par convolution de l'image avec un masque de différences.

Approximation de la dérivée

- Une image est discrète par nature. Les premières approches ont donc consisté à approximer les dérivées par différences finies :

- $G_x(x, y) = I(x, y) - I(x - n, y)$

ou alors :

- $G_x(x, y) = I(x + n, y) - I(x - n, y)$

avec en général $n=1$.

Ces dérivées sont calculées par convolution de l'image avec un masque de différences.

Quels masques de convolution pour $n=1$ dans les directions x et y ?

Approximation de la dérivée – filtre de Prewitt

- Filtre de Prewitt= approximation des dérivées (en tenant compte des diagonales)



1	0	-1
1	0	-1
1	0	-1

V_x

-1	-1	-1
0	0	0
1	1	1

V_y

A tester !

Approximation de la dérivée – filtre de Sobel

- Filtre de Sobel= approximation des dérivées (en tenant compte des diagonales)+ moyenneur



1	0	-1
2	0	-2
1	0	-1

Sx

-1	-2	-1
0	0	0
1	2	1

Sy

Approximation de la dérivée – filtre de Sobel

- Filtre de Sobel

- Approximation de la dérivée en x

Sx

1	0	-1
2	0	-2
1	0	-1



Approximation de la dérivée – filtre de Sobel

- Filtre de Sobel
 - Approximation de la dérivée en y

Sy

-1	-2	-1
0	0	0
1	2	1



Détection des contours

- Utilisation des filtres de Sobel

- Estimation de la norme du gradient

- Création d'une nouvelle image telle que

$$I'(x,y) = \sqrt{(I * S_x)^2 + (I * S_y)^2} \quad \text{avec } I = I(x,y)$$

- Seuillage pour obtenir une image en noir et blanc

Points à discuter-tester

- Amplitude du filtre S_x
 - Valeurs de -1020 à 1020
 - Normalisation
 - Division par 8 et partie entière à l'entier le plus petit : $[-128, 127]$
 - Translation de 128 $\rightarrow [0, 255]$
 - Division par 4 : $[-255, 255]$
 - Valeur absolue $\rightarrow [0, 255]$

1	0	-1
2	0	-2
1	0	-1

S_x

Points à discuter-tester

- Autres solutions (pas de diviseur ni de décalage) :
 - **écrêter** : tout résultat supérieur à 255 est arrondi à 255, et tout résultat inférieur à 0 est arrondi à 0 (cf. Gimp) ou bien on prend la valeur absolue
 - ramener l'ensemble des résultats (obtenus pour tous les pixels) dans l'intervalle $[0, 255]$ à l'aide d'une règle de trois, ce qui nécessite de les parcourir pour en rechercher la valeur max et la valeur min; il s'agit d'une démarche de **renormalisation**.

Exemple



Détection avec Sobel
sans seuillage



Seuillage avec $S=25$



Seuillage avec $S=60$

Autre exemple



Gradient



Gradient seuillé ($|G| > G_{min}$)



Seuil faible



Seuil grand

Approximation du Laplacien

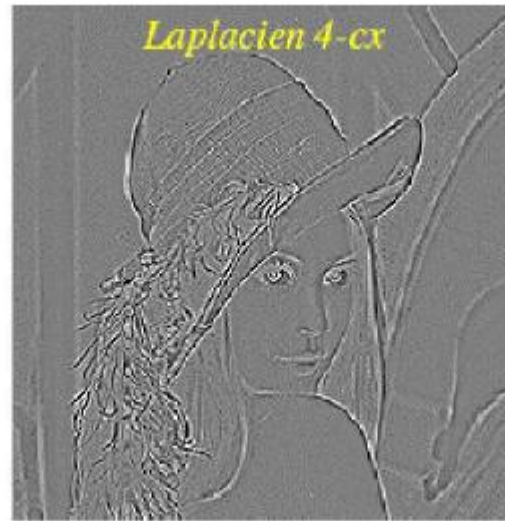
- Une autre approche pour trouver les contours de l'image est d'utiliser une approximation de la dérivée seconde
- Pour cela, on utilise le **Laplacien** comme opérateur

$$\nabla^2 I(x, y) = \frac{\partial^2 I}{\partial x^2}(x, y) + \frac{\partial^2 I}{\partial y^2}(x, y)$$

Approximation du Laplacien

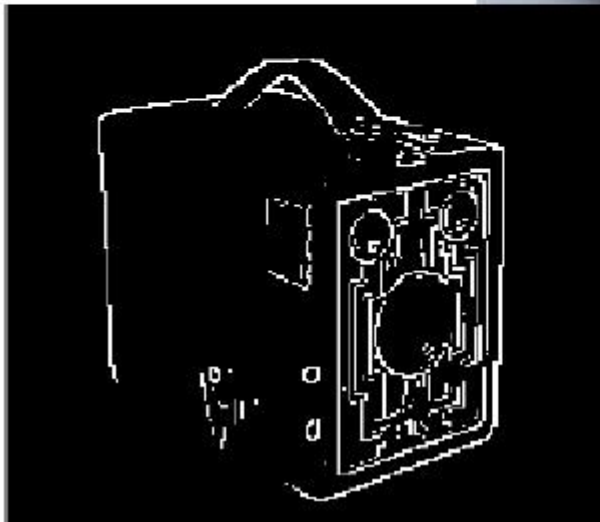
- Calcul par convolution

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \text{ ou } \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

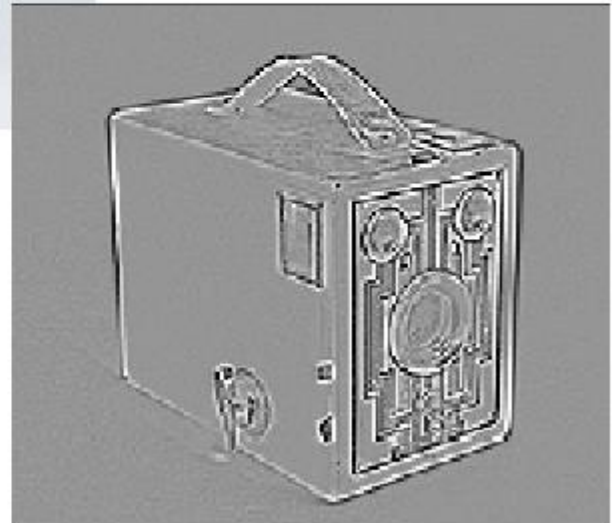


Comparaison Gradient Laplacien

Gradient



Laplacien



Que choisir ?

- Aucun opérateur n'est parfait pour détecter les contours
- En pratique, on obtient des contours incomplets
 - *il y a des pixels superflus*
 - *il y a des pixels manquants*
 - *il y a des erreurs de position et d'orientation des pixels des contours*
- Un opérateur de détection de contour n'est qu'une première étape dans la chaîne de segmentation

Détection des contours par seuillage du gradient : principe

- Idée : maximum local de gradient \rightarrow présence d'un contour
- Méthode pour obtenir une image des contours :
 - estimation du gradient en chaque point de l'image
 - extraction des maxima locaux de la norme du gradient dans la direction du gradient
 - sélection des maxima locaux significatifs par seuillage
 - fermeture des contours en traçant les chemins suivant une ligne de crête dans l'image de la norme du gradient

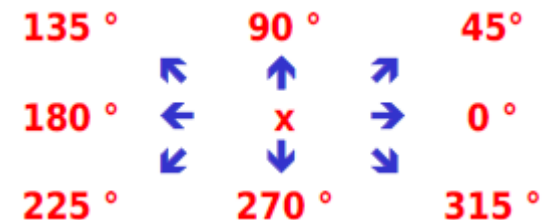
Détection des contours par seuillage du gradient -1-

1. Calculer l'intensité du gradient dans l'image

- Filtre de Sobel en X et Y
- Calcul de la norme $|G| = |G_x| + |G_y|$

2. Calculer les directions du gradient dans l'image

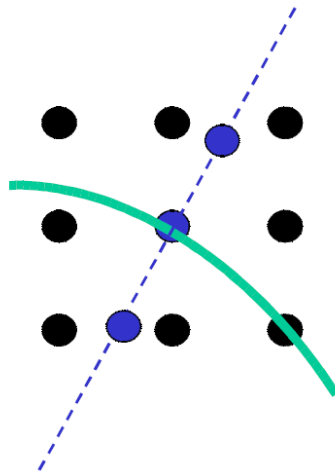
- Direction du gradient $\theta = \arctan (G_y / G_x)$
- Arrondi des directions par multiples de 45°



Détection des contours par seuillage du gradient - 2

3. Suppression des non-maxima

- Si la norme du gradient en un pixel (x,y) est inférieure à la norme du gradient d'un de ses 2 voisins le long de la direction du gradient, alors mettre la norme pour le pixel (x,y) à zéro



Détection des contours par seuillage du gradient - 3

4. Seuillage des contours (hystérésis)

Objectif : limiter la fragmentation des contours obtenus

- Utilise deux seuils : un seuil haut (**Sh**) et un seuil bas (**Sb**)
- *Pour chaque pixel, calcul de la norme du gradient :*
 - Si **norme(x,y) < Sb**, alors le **pixel est mis à zéro** (non contour)
 - Si **norme(x,y) > Sh**, alors le **pixel est sur un contour**
 - Si **Sb ≤ norme(x,y) ≤ Sh**, alors le **pixel est un pixel de contour s'il est connecté à un autre pixel déjà accepté comme pixel de contour**

Exemple 1

Image
originale



Sobel



Suppression des
non-maxima



Seuillage



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

Exemple 2

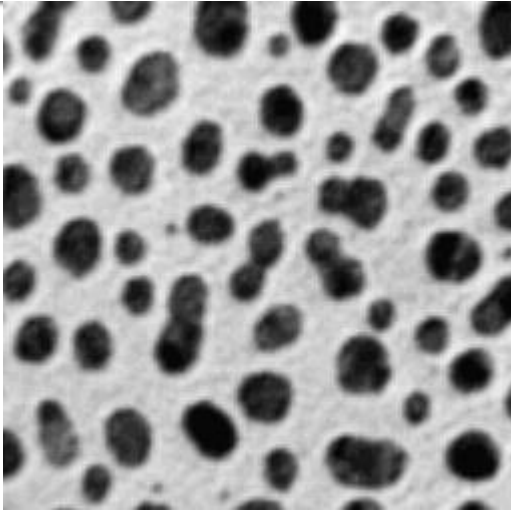
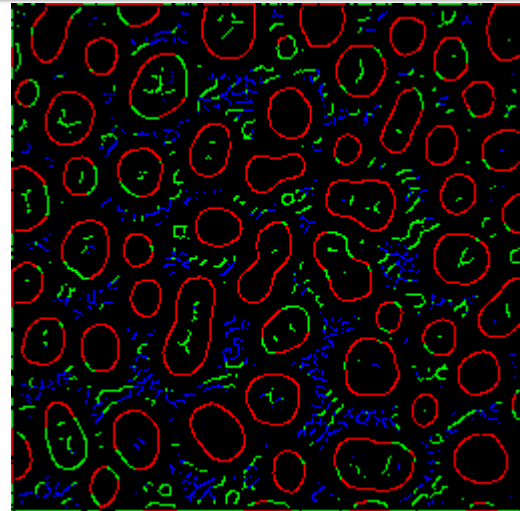
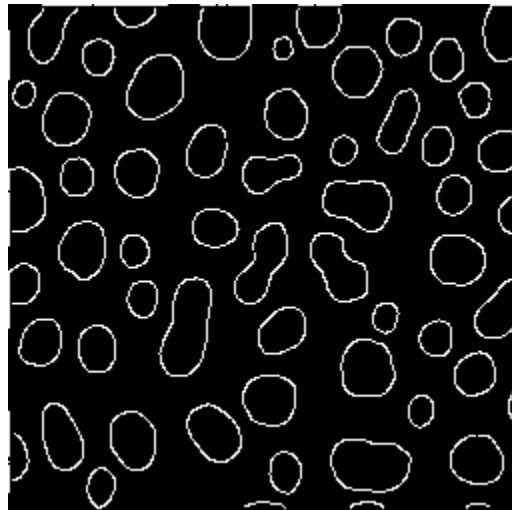


Image à segmenter

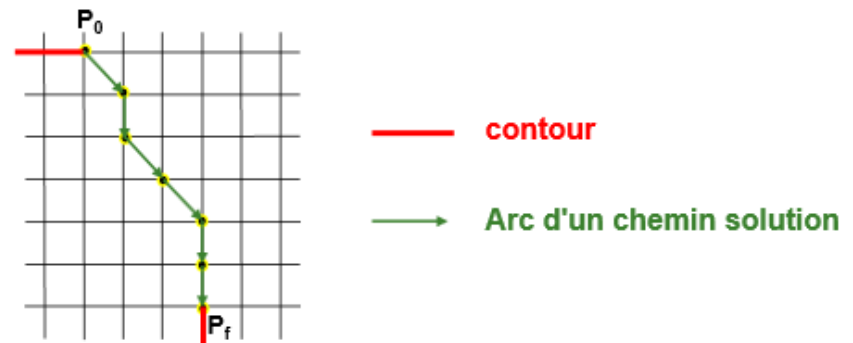


Maxima locaux du gradient



Détection des contours par seuillage du gradient -4

5. Fermeture des contours en traçant les chemins suivant une ligne de crête dans l'image de la norme du gradient
 1. Repérer les points extrémité (énumération des configurations possibles)
 2. Choix entre les points candidats : on explore tous les chemins possibles à partir de chaque point candidat. Le poids d'un chemin peut être défini comme la somme de la norme du gradient en chacun de ses points



Détection des contours par seuillage du gradient -4

5. Fermeture des contours en traçant les chemins suivant une ligne de crête dans l'image de la norme du gradient

Algorithme :

- Balayage de l'image des éléments essentiels
- Si extrémité, recherche parmi les voisins du meilleur candidat à la fermeture
 - Elaboration de l'arborescence de tous les chemins possibles
 - Attribution à chaque chemin d'un coût = \sum des normes des gradients du chemin
 - Prolongation du contour par le premier point du chemin au plus fort coût (ligne de crête)
- Réitération de l'algorithme à partir de ce nouveau point
- Arrêt de la fermeture si rencontre d'un point de contour ou nombre max d'itérations atteint

- Méthode à mettre en œuvre dans le TP5 à réaliser pour fin mai.