



République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université des Sciences et de la Technologie Houari Boumediene  
Faculté d'Electronique et d'Informatique  
Département d'Informatique



# La vision par ordinateur

## *Chapitre 2 : Techniques de base de traitement de l'image*

Master 2 : Systèmes Informatique Intelligents  
Lyes\_sii@yahoo.fr lyes\_abada@yahoo.fr

# Définition et Représentation de l'image

## Image :

Une image est **une fonction** , qui associe en un point donné une valeur (**niveau de gris**) qui dépend de la quantité de lumière :  $0 \leq f(x,y) \leq M$  pour tout  $(x,y)$  de l'image

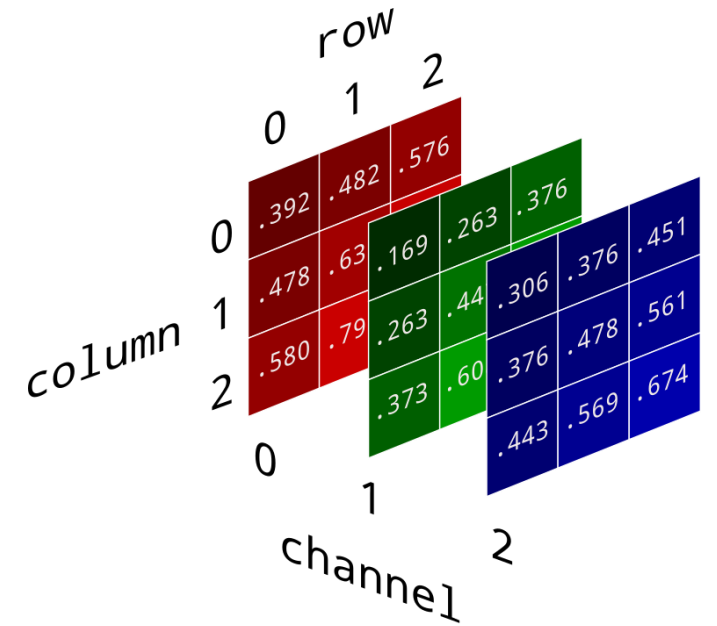
## Image analogique :

- L'image analogique obtenue par le système d'acquisition.
- L'image analogique ne peut pas être stockée puisque elle est continue.
- Il n'est pas possible de reproduire l'image originale à l'identique.
- Les copies sont nécessairement dégradées par rapport à l'original.

## Image Numérique :

Image analogique digitalisée (échantillonnage),

- Un seul canal (ex : 8 bits) pour le niveau de gris
- Trois ou quatre canaux pour les images couleur (ex : 24 bits, 8 bits pour chaque couleur RVB).

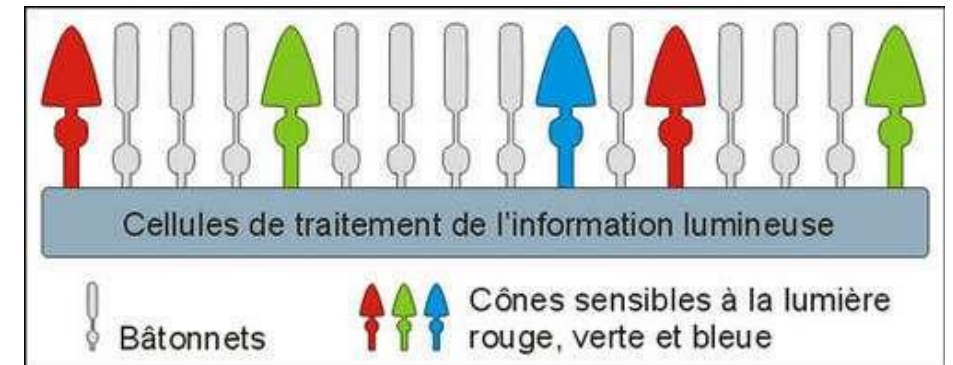
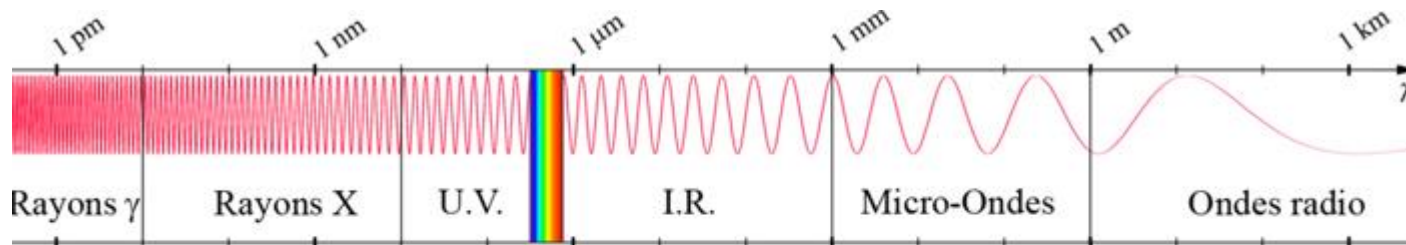


# Définition et Représentation de l'image

Les cônes sont des terminaisons nerveuses sensibles aux couleurs (6 à 7 millions)

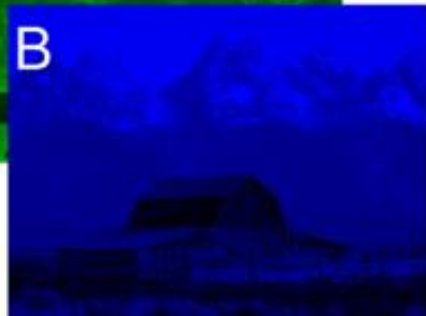
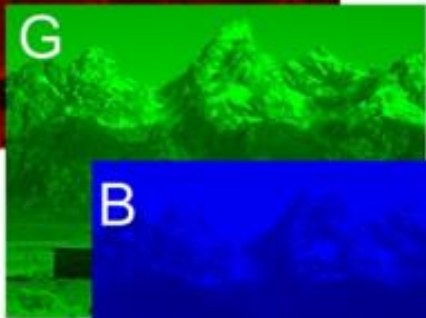
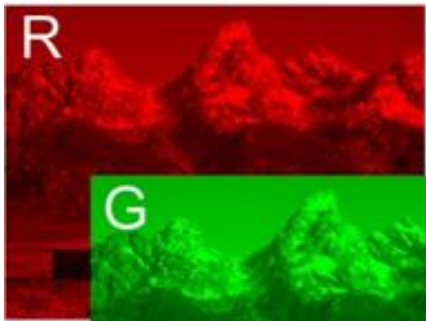
- Les cônes « bleus » sensibles à des longueurs d'onde d'environ 430nm
- Les cônes « verts » sensibles à des longueurs d'onde d'environ 530nm
- Les cônes « rouges » sensibles à des longueurs d'onde d'environ 630nm

Les bâtonnets (100 à 200 millions), sont 10 000 fois plus sensibles à la luminosité mais moins précis pour la perception des détails et des couleurs.



# Définition et Représentation de l'image

RVB(RGB)



width

height



$$H \times W \times 3$$

Niveau de gris  
(Gray Scale)

width

height

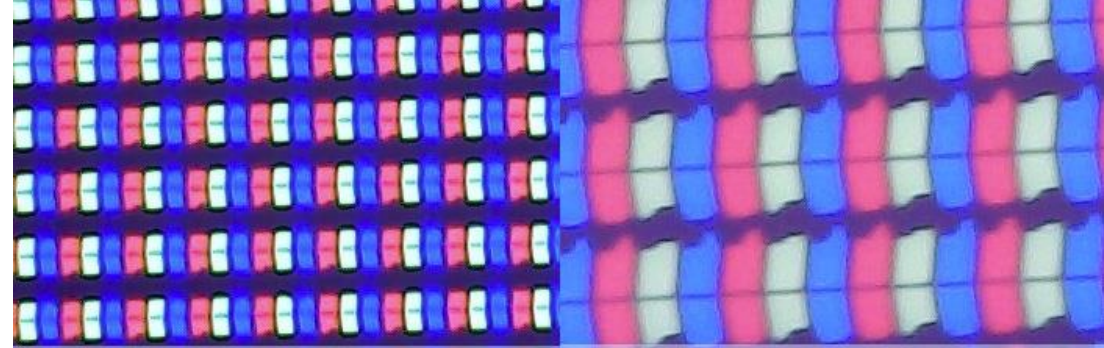
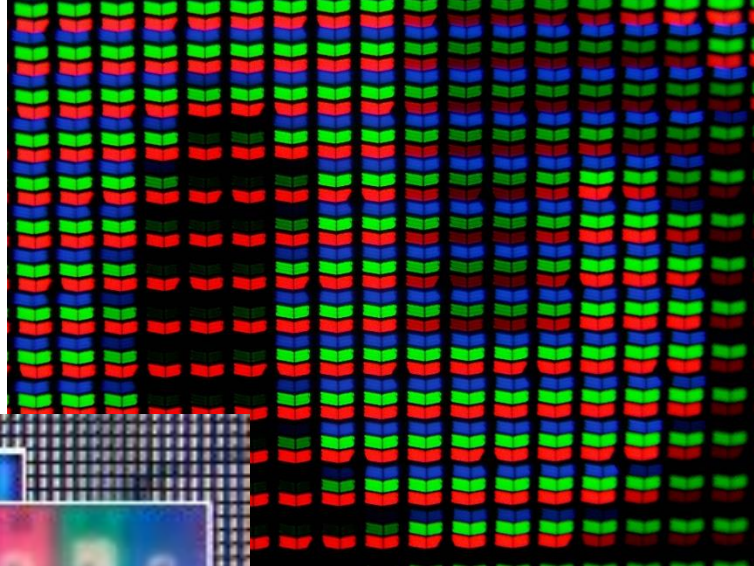
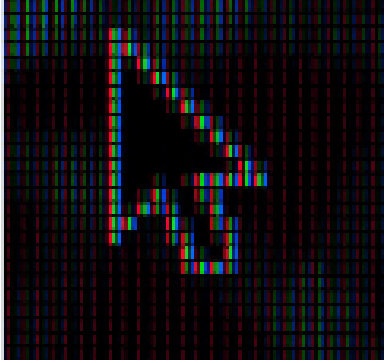


$$H \times W$$

$$0 \leq f(x,y) \leq 255 \text{ ?????!!!}$$

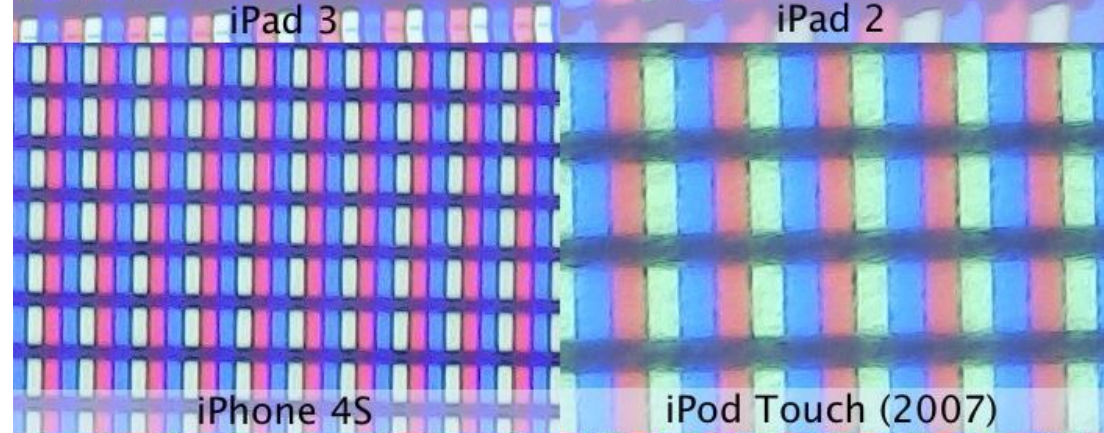
$$0.2989 * R + 0.5870 * G + 0.1140 * B$$

# Définition et Représentation de l'image



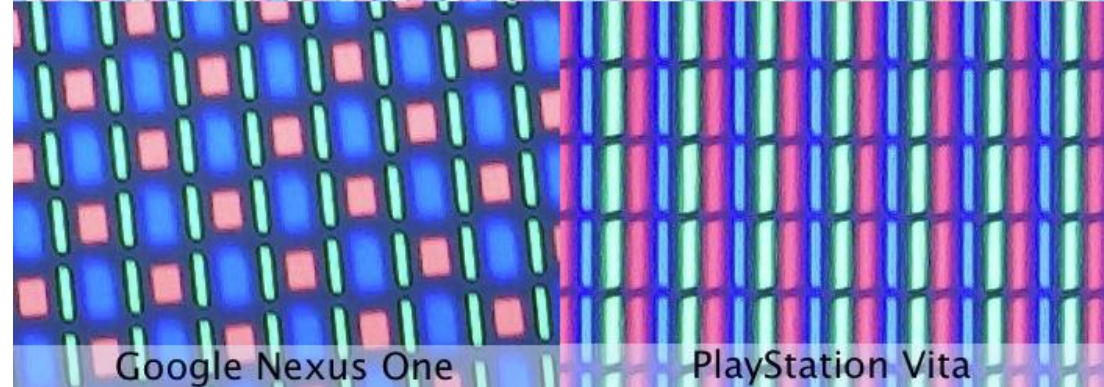
iPad 3

iPad 2



iPhone 4S

iPod Touch (2007)



Google Nexus One

PlayStation Vita



Les couleurs dans  
un écran digital

# Définition et Représentation de l'image

---

Pour l'acquisition d'une image on utilise des capteurs selon l'application désirée :

- application spatiale : Satellite
- application avec grande précision : caméra CCD (charge-coupled device)
- application (précision non exigée) : caméra TV

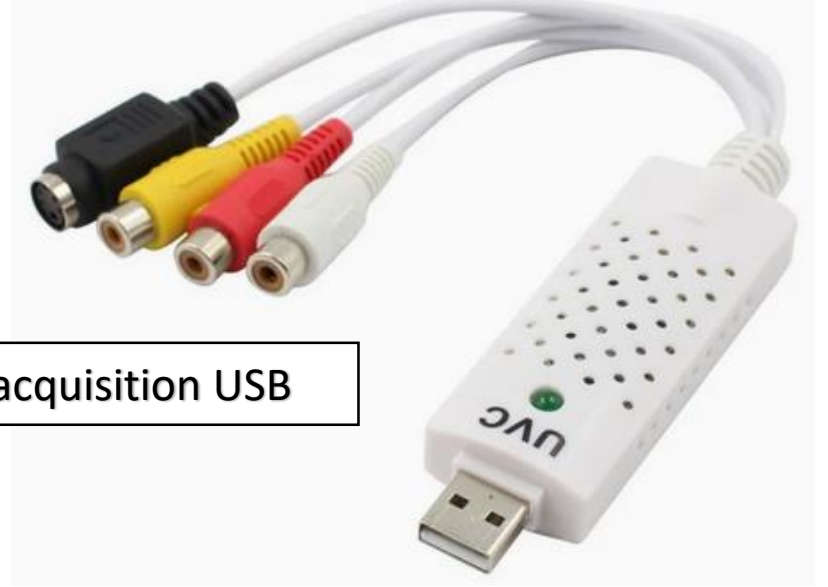
**Un capteur est constitué de:**

- Dispositif optique
- Système de transfert énergie lumineuse en énergie électrique
- Carte pour Echantillonnage du signal vidéo, numérisation, mémorisation

# Définition et Représentation de l'image



Carte  
d'acquisition  
spécialisée dans  
la numérisation  
d'images



Carte d'acquisition USB



Une caméra CCD

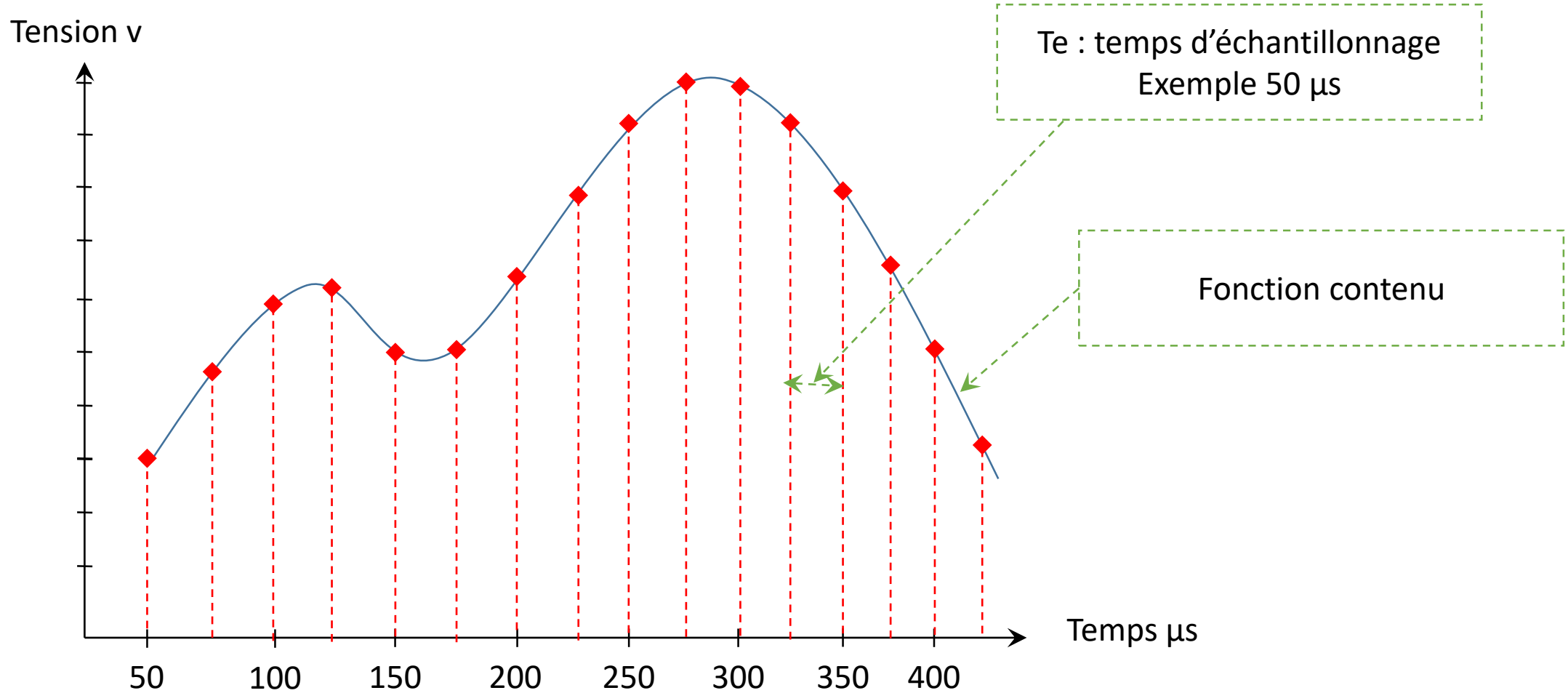


Une caméra  
numérique

# Rappel sur la codification en binaire

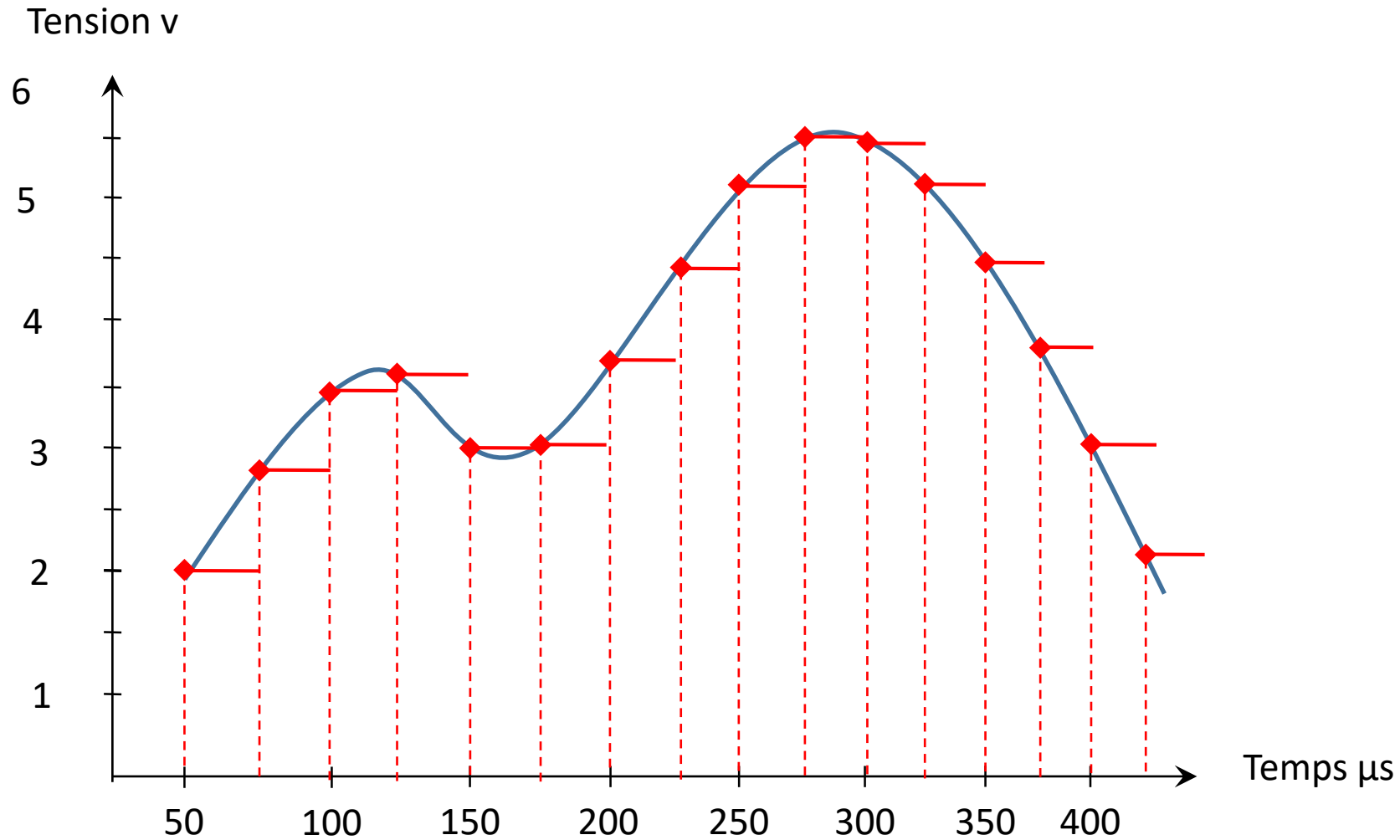
## Numérisation d'un signal analogique : Échantillonnage

Passage d'une fonction continue vers une fonction discrète (discrétisation)



# Rappel sur la codification en binaire

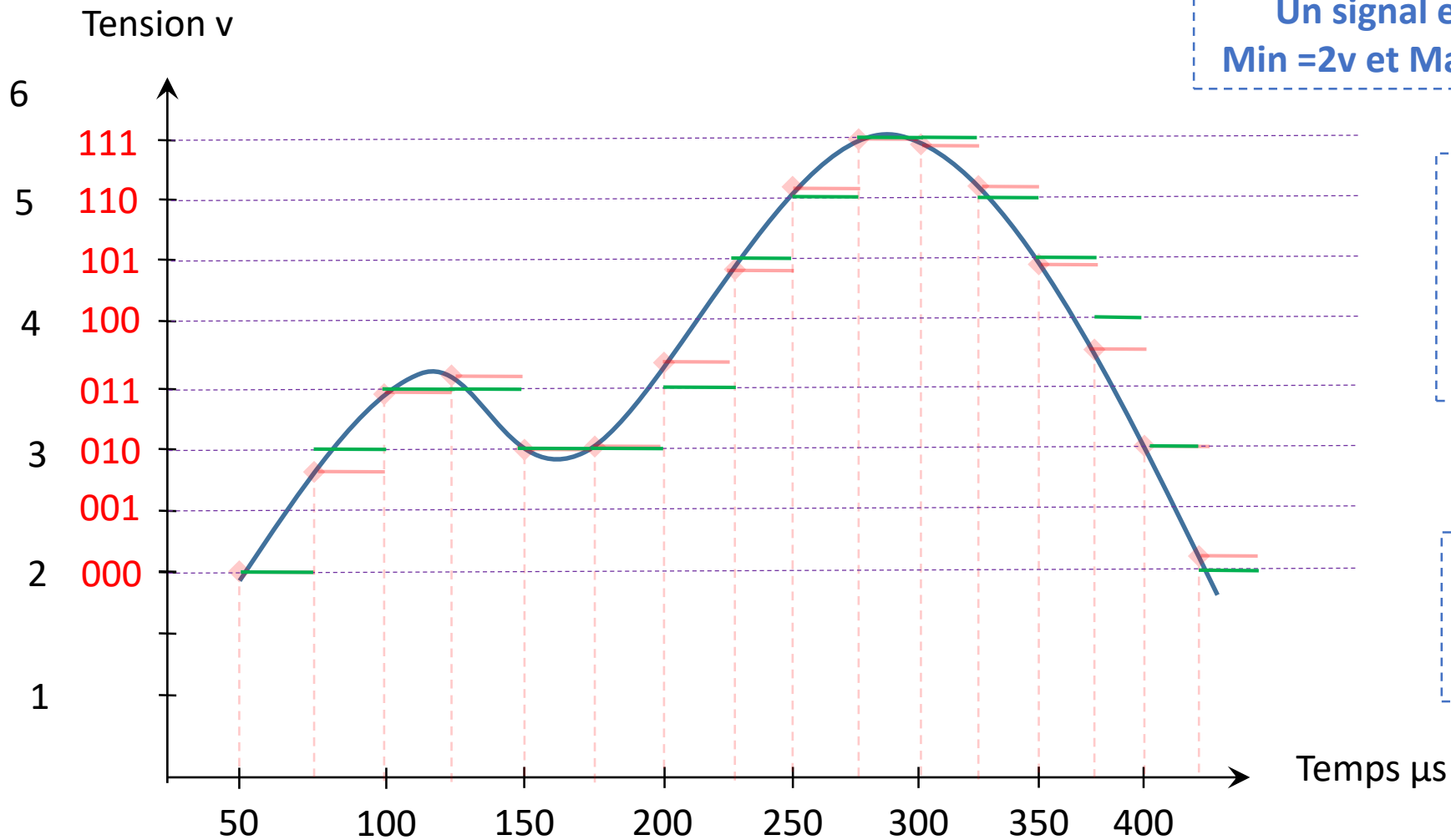
Numérisation d'un signal analogique : **Échantillonnage**



$$T_e = 25 \mu s$$
$$F_e = \frac{1}{25 * 10^{-6}} = 10 * 10^2 \text{ Hz}$$

# Rappel sur la codification en binaire

Numérisation d'un signal analogique : **Quantification**

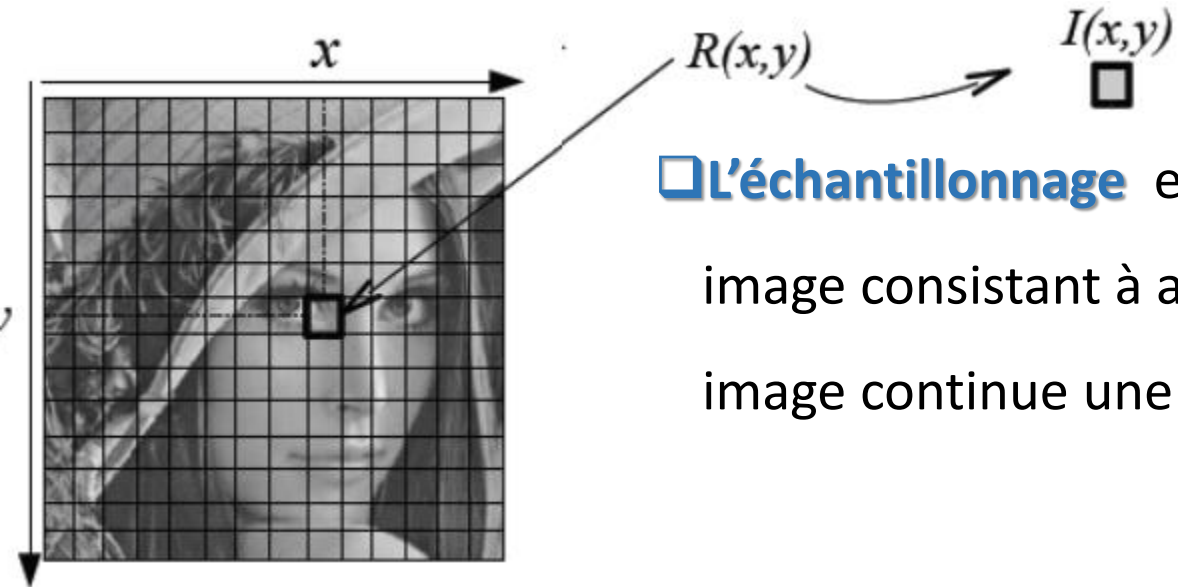


Un signal est compris entre  $2v$  et  $6v$   
Min =  $2v$  et Max =  $6v \rightarrow \Delta u = 6 - 2 = 4v$

Nombre de bits  $n = 3$   
Nombre de valeurs possible =  $2^3$   
 $\rightarrow$  Aller de  $[2,6]$  à  $[0,2^3-1]$

$$Val_{Num} = \frac{Val_v - min_v}{Max_v - Min_v} * 2^{nbit} - 1$$

# Définition et Représentation de l'image



□ **L'échantillonnage** est le procédé de discrétisation spatiale d'une image consistant à associer à chaque zone rectangulaire  $R(x,y)$  d'une image continue une unique valeur  $I(x,y)$ .

□ **La quantification** désigne la limitation du nombre de valeurs différentes que peut prendre  $I(x,y)$ .



□ **Une image numérique** est donc une image **échantillonnée et quantifiée**.

# Définition et Représentation de l'image

*Résolution...*

*...spatiale :*

*Échantillonnage*



256x256



128x128



64x64



32x32

*...tonale :*

*Quantification*



6 bits



4 bits



3 bits



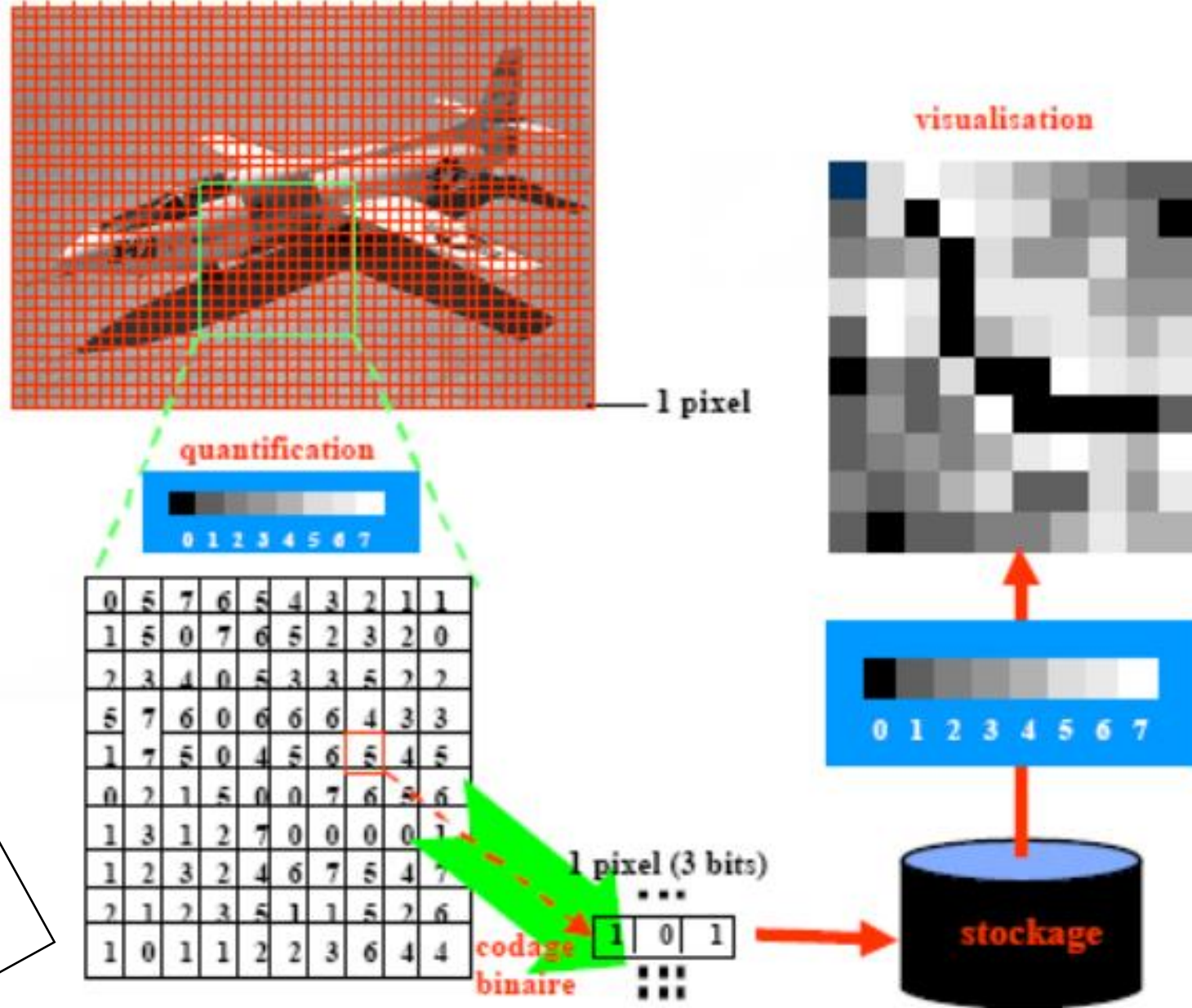
2 bits



1 bit

# Définition et Représentation de l'image

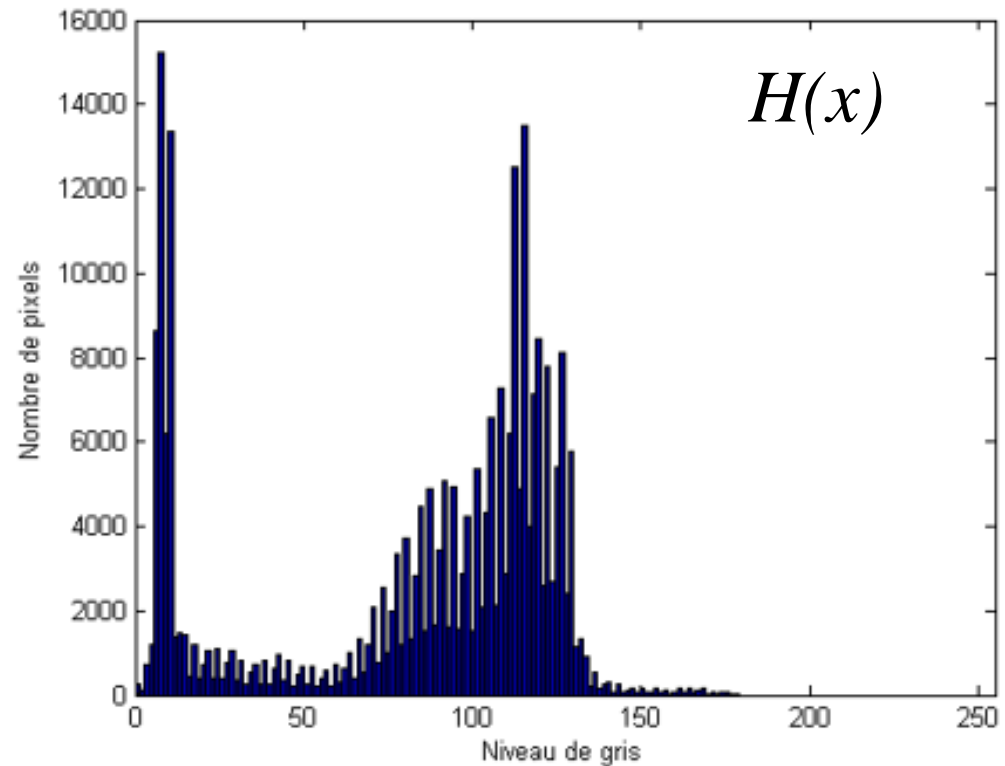
numérisation = échantillonnage + quantification



## Histogramme

L'histogramme représente la répartition des pixels en fonction de leur niveau de gris

$H(x)$  est le nombre de pixels dont le niveau de gris est égal à  $x$



# Définition et Représentation de l'image

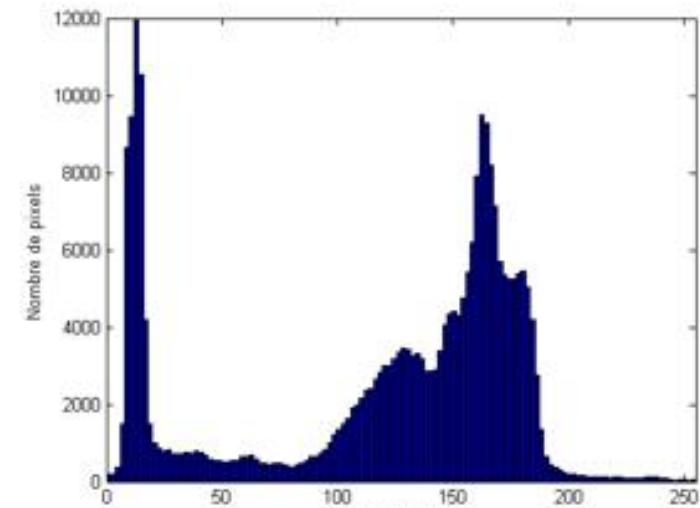
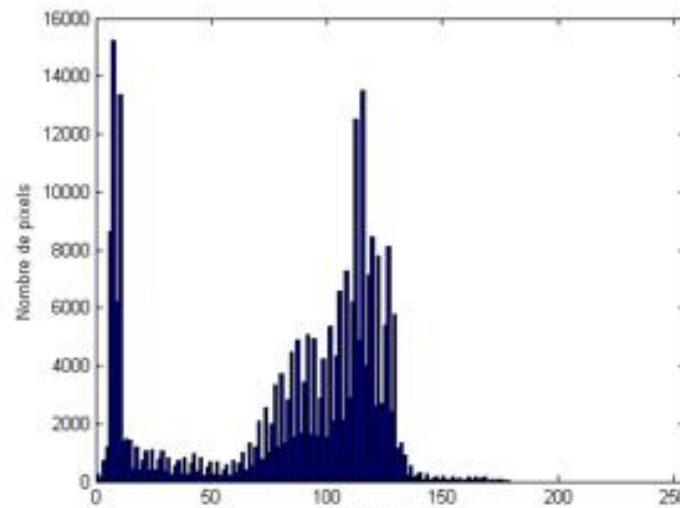
Étirement d'histogramme :

Ou la Normalisation de l'histogramme :



→ 8bit (max = 255)

$$\text{NewVal} = \frac{\text{Val} - \text{min}}{\text{Max} - \text{Min}} * 255$$



# Définition et Représentation de l'image

Calculer l'Histogramme de cette image.

$$0 \leq f(x,y) \leq 7$$

Normaliser cette l'Histogramme.

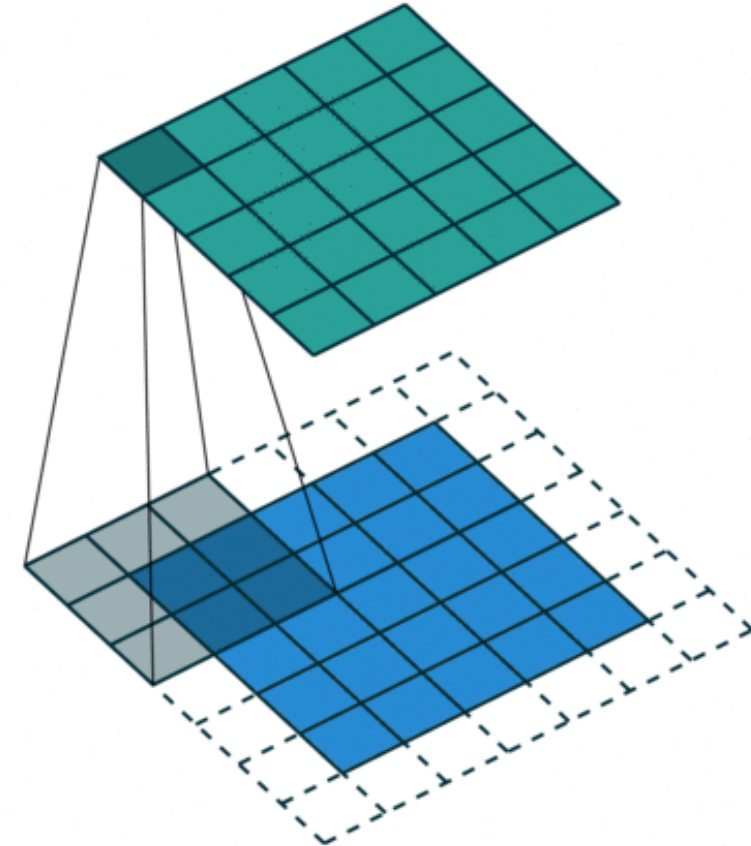
	0	1	2	3	4	5
0	0	2	3	0	4	2
1	1	1	5	0	2	5
2	0	5	2	0	5	0
3	2	2	2	3	2	0
4	0	1	0	4	3	2
5	1	1	5	0	2	4

# Opérations sur les images

- ❑ **La convolution** C'est l'opérateur de base du traitement linéaire des images.
- ❑ Elle permet de calculer l'approximation des dérivées horizontales et verticales
- ❑ Pour calculer une convolution, on remplace la valeur de chaque pixel par la valeur du produit scalaire entre les valeurs du noyau de convolution et les valeurs du voisinage du pixel considéré (par rapport à l'origine (0,0) du noyau de convolution).

**Attention : implémentation « parallèle ».**

- ❑ Remarque: la convolution est appliquée (entre autre) à filtrer l'image pour atténuer le bruit présent.



# Opérations sur les images

Exemple de convolution :

$a_{00}$	$a_{01}$	$a_{02}$	$a_{03}$	$a_{04}$	$a_{05}$	$a_{06}$	$a_{07}$
$a_{10}$	$a_{11}$	$a_{12}$	$a_{13}$	$a_{14}$	$a_{15}$	$a_{16}$	$a_{17}$
$a_{20}$	$a_{21}$	$a_{22}$	$a_{23}$	$a_{24}$	$a_{25}$	$a_{26}$	$a_{27}$
$a_{30}$	$a_{31}$	$a_{32}$	$a_{33}$	$a_{34}$	$a_{35}$	$a_{36}$	$a_{37}$
$a_{40}$	$a_{41}$	$a_{42}$	$a_{43}$	$a_{44}$	$a_{45}$	$a_{46}$	$a_{47}$
$a_{50}$	$a_{51}$	$a_{52}$	$a_{53}$	$a_{54}$	$a_{55}$	$a_{56}$	$a_{57}$
$a_{60}$	$a_{61}$	$a_{62}$	$a_{63}$	$a_{64}$	$a_{65}$	$a_{66}$	$a_{67}$
$a_{70}$	$a_{71}$	$a_{72}$	$a_{73}$	$a_{74}$	$a_{75}$	$a_{76}$	$a_{77}$

|

$M_{00}$	$M_{01}$	$M_{02}$
$M_{10}$	$M_{11}$	$M_{12}$
$M_{20}$	$M_{21}$	$M_{22}$

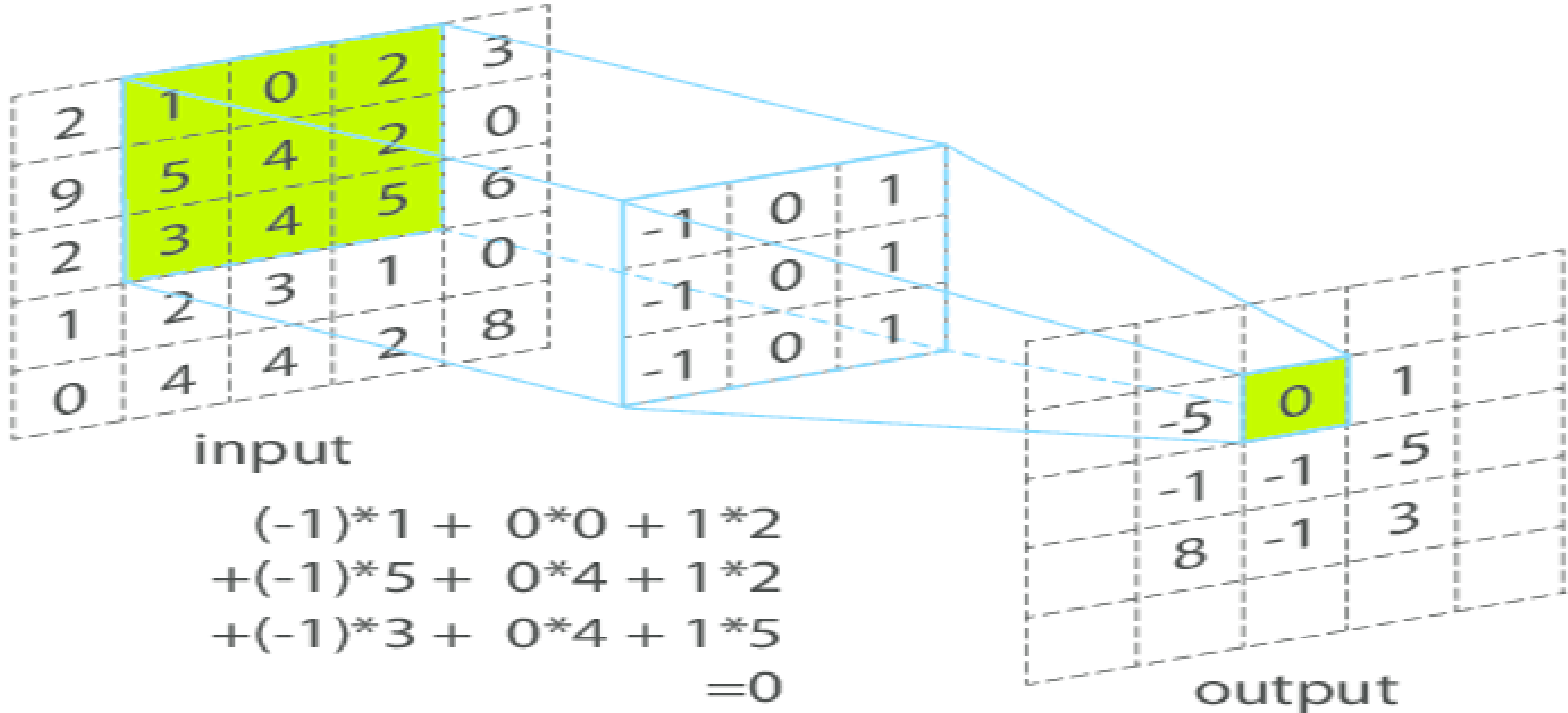
h

$$a_{33} =$$

$$\begin{aligned}
 & a_{22}M_{00} + a_{23}M_{01} + a_{24}M_{02} + \\
 & a_{32}M_{10} + a_{33}M_{11} + a_{34}M_{12} + \\
 & a_{42}M_{20} + a_{43}M_{21} + a_{44}M_{22}
 \end{aligned}$$

# Opérations sur les images

Exemple de convolution :



# Opérations sur les images

❑ Soit  $I$  une image numérique et  $h$  une fonction de  $[x_1, x_2] \times [y_1, y_2]$  à valeurs réelles.

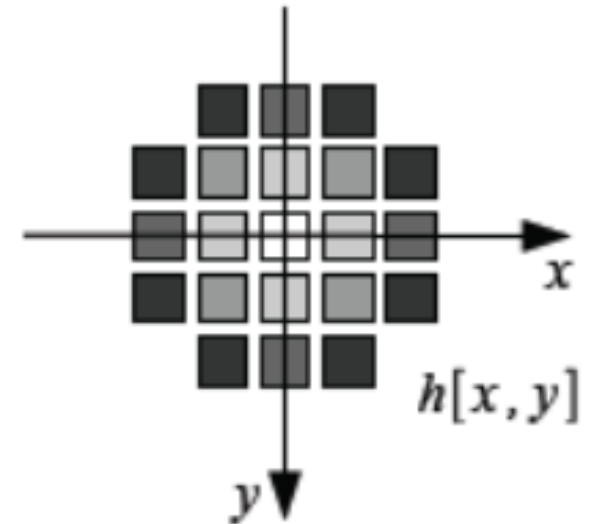
❑ La **convolution** discrète de  $I$  par  $h$  est définie par :

$$\square (I * h)[x, y] = \sum_{i=x_1}^{x_2} \sum_{j=y_1}^{y_2} h(i, j) I(x - i, y - j)$$

❑ Les nouvelles valeurs du pixel sont calculées par **produit scalaire**

entre le noyau de convolution et le voisinage correspondant du pixel.

❑ La fonction  $h$  est dite **noyau de convolution** ou **masque de convolution**



La convolution :

Le masque de convolution est le plus souvent **carré** de taille **3x3** ou **5x5** ou plus,  
(mais **impair**)

Propriétés de la convolution :

Commutativité :  $h * g = g * h$

Associativité :  $(h * g) * k = h * (g * k)$

Distributivité :  $h * (g + k) = (h * g) + (h * k)$

Appliquer deux masque h1 et h2 ou deux fois le masque h sur une image I !!!???

Appliquer les deux masques h1 et h2 sur l'image initiation et faire la somme des résultats

Calculer la somme de h1 et h2 et appliquer le noyau sur l'image initiation.



Appliquer un masque h1 sur l'image initiation et faire la somme du résultat avec l'image initiale.

→ exemple rehaussement des contours (Contours + Image) ,

La convolution :

Problème : Que faire avec les bords de l'image ?

- Mettre à zéro (0)** → les pixels en dehors de l'image sont de valeur nulle
- Convolution partielle** → Sur une portion du noyau
- Miroir de l'image** →  $f(x-1,y) = f(x+1,y)$
- Duplication** → on duplique les premières et dernières lignes et colonnes au delà des bords ;

## Le bruit dans une image

Le bruit est l'ensemble des pixels de l'image qui ont des valeurs aberrantes (un point blanc au milieu des points noirs ou vice versa.)

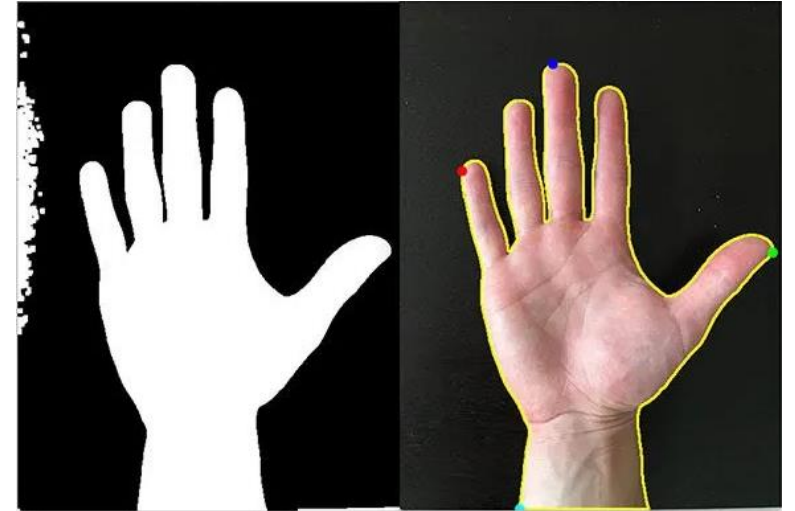
Le bruit représente les parasites ou les interférences d'un signal, c'est-à-dire les parties du signal déformées localement, il est causé par:

- Une instabilité de la source du signal.
- Un signal parasite provenant d'ailleurs.
- Une instabilité du récepteur.
- Etc ..

**Donc nécessité de filtrer l'image**

## Le filtrage d'une image

Le filtrage consiste à appliquer une transformation (appelée filtre) à tout ou une partie d'une image numérique en appliquant un opérateur .



❑ **Les filtres passe-bas:** consistent à **atténuer** les composantes de l'image ayant une fréquence haute. Appelé habituellement **le lissage**

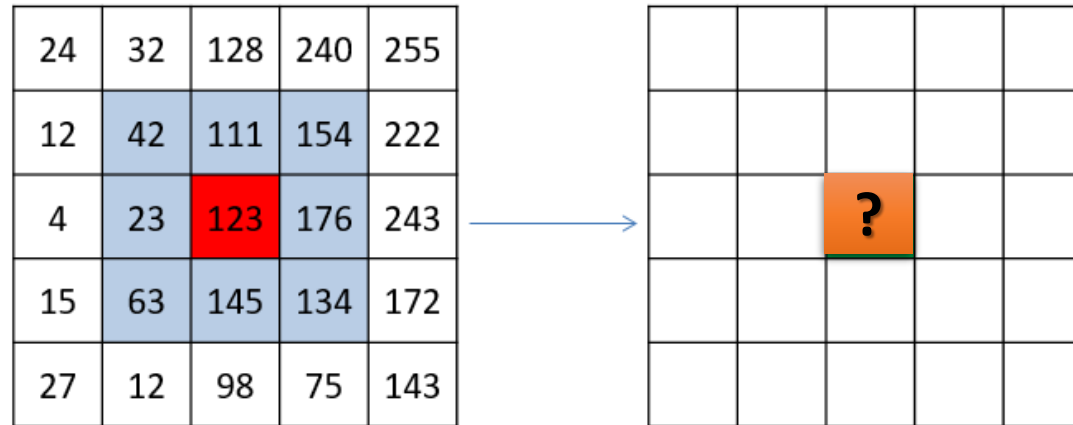
❑ **Les filtres passe-haut:** à l'inverse des filtres passe-bas, ils permettent d'accentuer les composantes de basse fréquence de l'image et permettent notamment **d'accentuer** les détails.

## Les deux catégories de filtres

### a) Les filtres linaires

Ils utilisent une combinaison linéaire des valeurs du voisinage pour déterminer le niveau de gris d'un pixel.

□ Le filtre **moyenneur**: remplacent le pixel courant par la valeur moyenne des pixels sur une fenêtre.



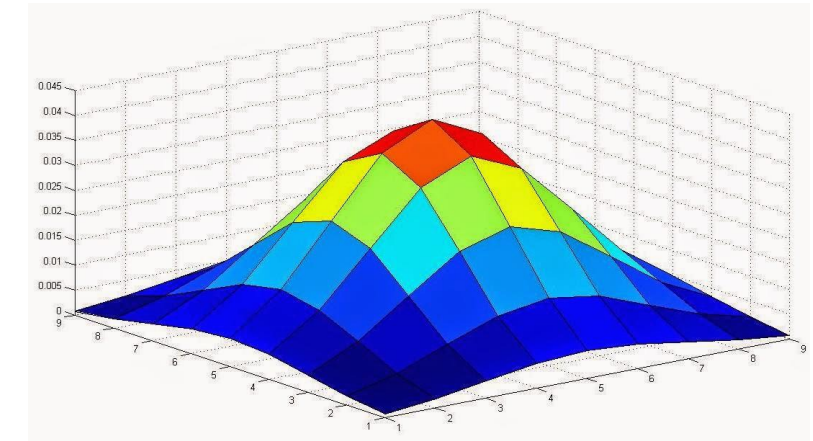
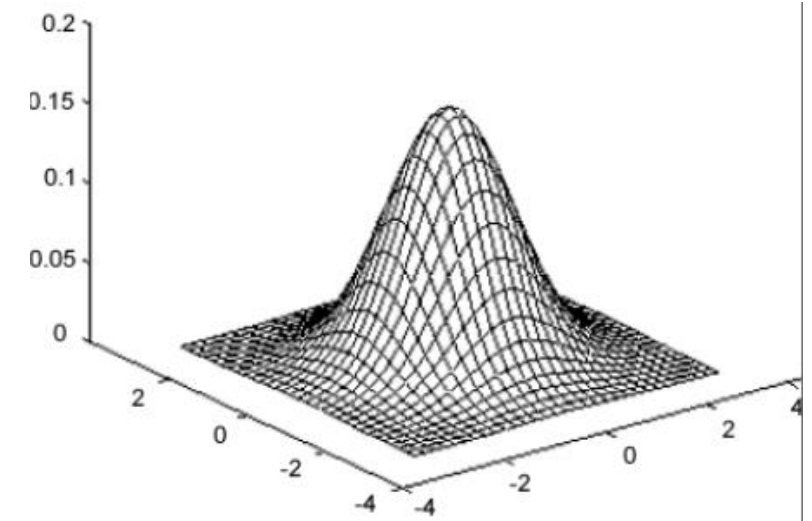
Le masque (noyau) de convolution du filtre moyenneur????

# Opérations sur les images

## ❑ Le filtre de Gauss:

- Il effectue une convolution de l'image avec une gaussienne
- filtre passe-bas, il a pour effet d'adoucir (lisser) l'image, il effectue une convolution de l'image avec une gaussienne.

$$G_{\sigma}(x, y) = \frac{1}{2\pi\sigma^2} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)}$$



# Opérations sur les images

## Exemple de convolution gaussienne

Masque 5\*5

Avec  $\sigma = 1.4$

$$G_{\sigma}(x, y) = \frac{1}{2\pi\sigma^2} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)}$$



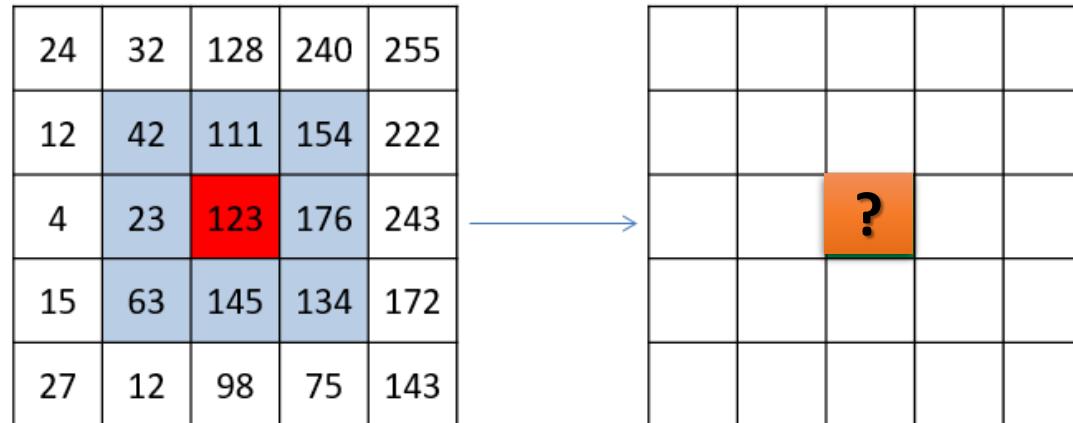
0.011	0.023	0.029	0.023	0.011
0.023	0.049	0.063	0.049	0.023
0.029	0.063	0.081	0.063	0.029
0.023	0.049	0.063	0.049	0.023
0.011	0.023	0.029	0.023	0.011

$$\approx \frac{1}{185} *$$

2	4	5	4	2
4	9	12	9	4
5	12	15	12	5
4	9	12	9	4
2	4	5	4	2

## b) Les filtres non linaires

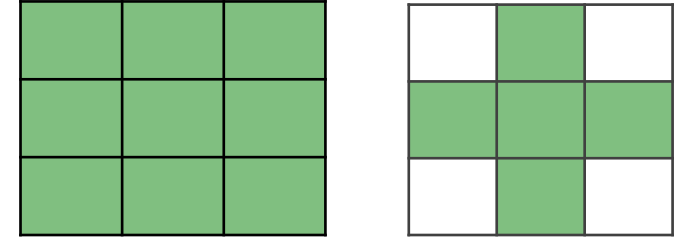
- **filtre médian**: classer les pixels voisins par ordre croissant des valeurs de niveaux de gris, puis affecter la valeur médiane au pixel central



## b) Les filtres non linaires

☐ filtres morphologiques: basés sur deux opérations:

Utilisent **un élément structurant** →  
définition du voisinage (ou de la connexité)



- ❖ **Erosion**: effectue un « et » logique entre les voisins d'un pixel (diminue le contour de l'ordre d'un pixel)
- ❖ **Dilatation**: effectue un « ou » logique entre les voisins d'un pixel (augmente l'épaisseur d'un contour)

# Opérations sur les images

## b) Les filtres non linaires

☐ filtres morphologiques: basés sur deux opérations:

Utilisent  $u$

dé

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

❖ Eros

d'un

connex

entre

0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	1
0	0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

le contour de l'ordre

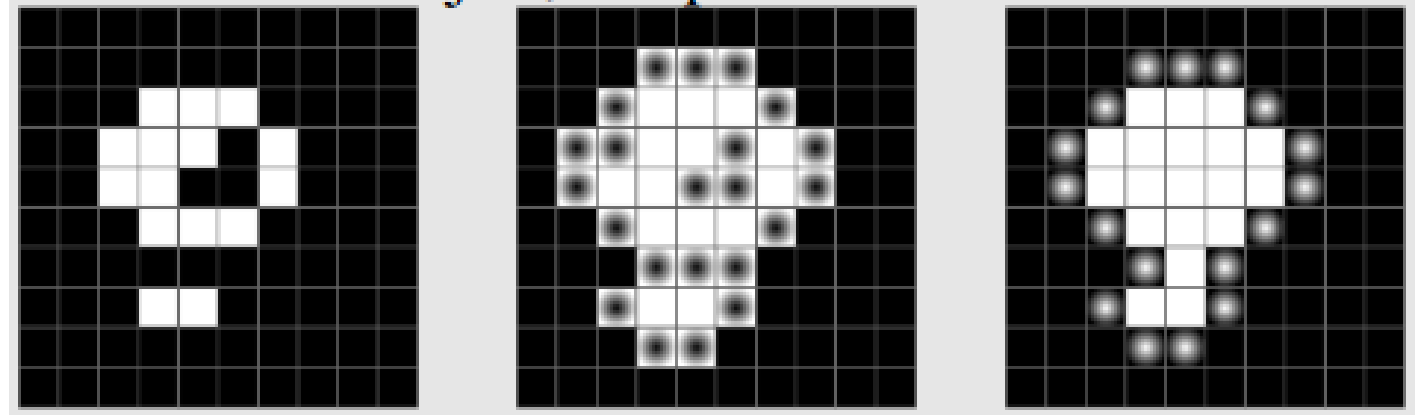
❖ Dilatation: effectue un « ou » logique entre les voisins d'un pixel (augmente l'épaisseur d'un contour)

# Opérations sur les images

☐ filtres morphologiques (suite):

☐ **Fermeture (dilatation puis érosion)**

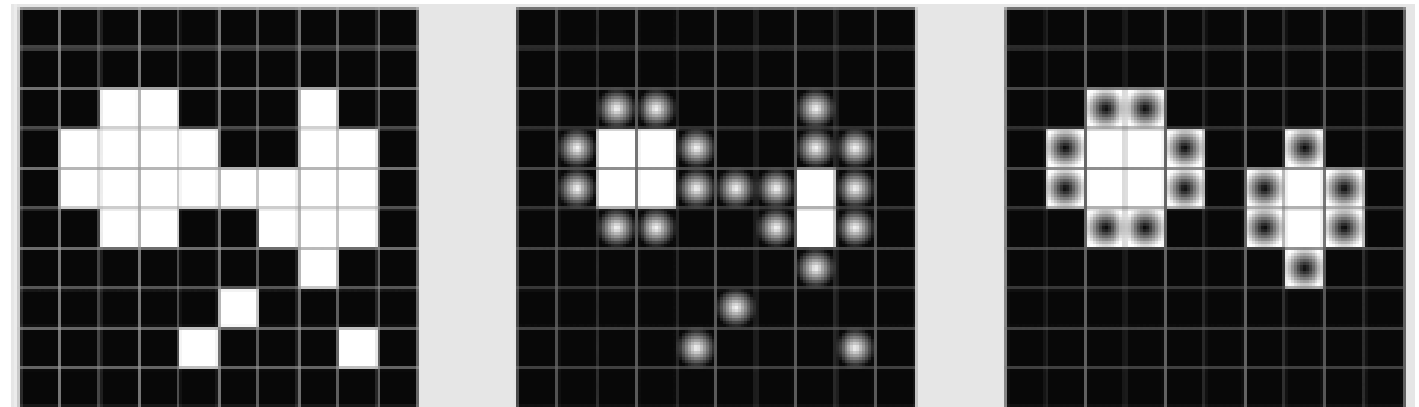
➔ Fermer les objets, remplir les trous



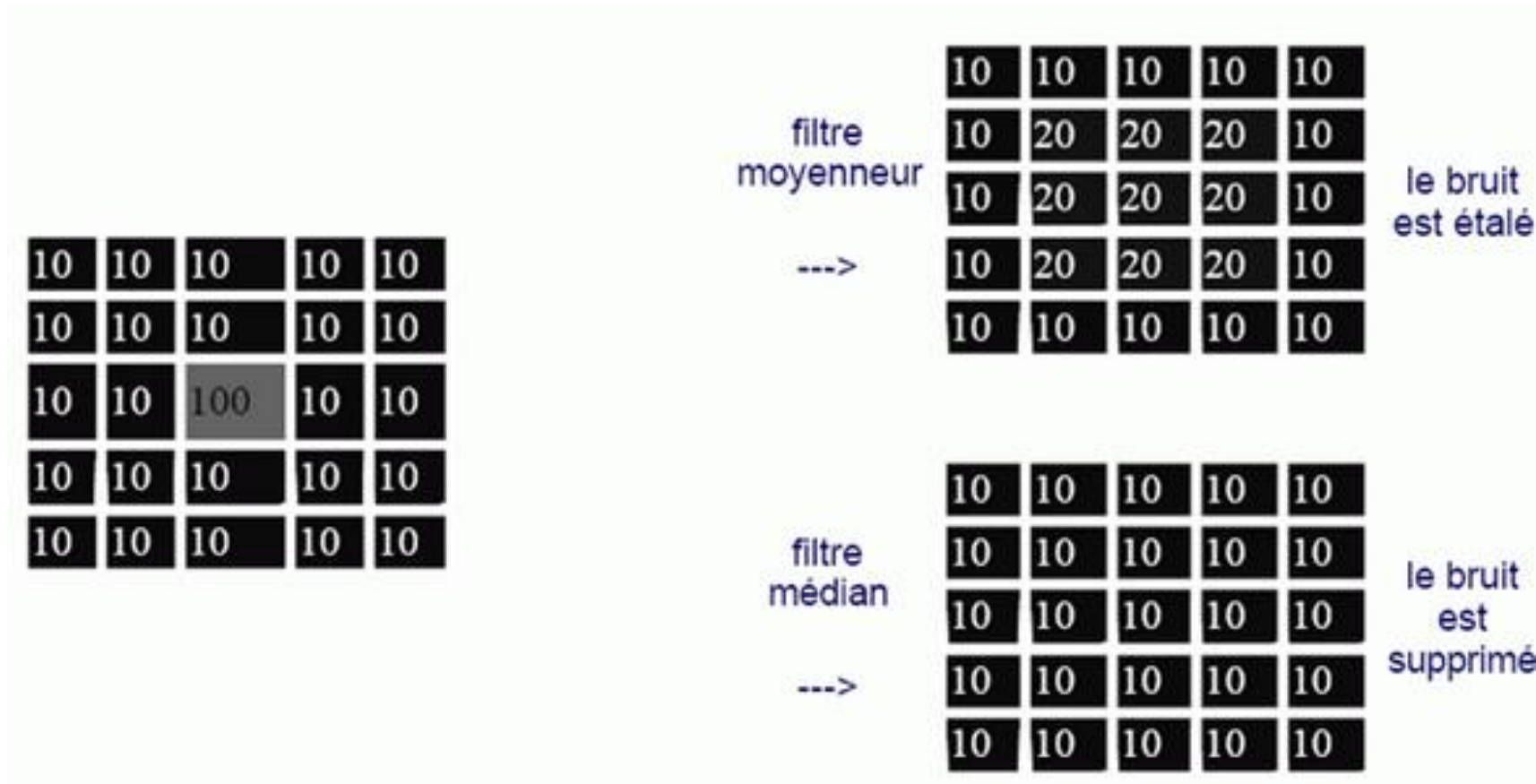
☐ **Ouverture (érosion puis dilatation)**

➔ Supprimer les objets de largeur inférieure à celle de l'élément structurant

➔ Séparer les objets fusionnés à tort



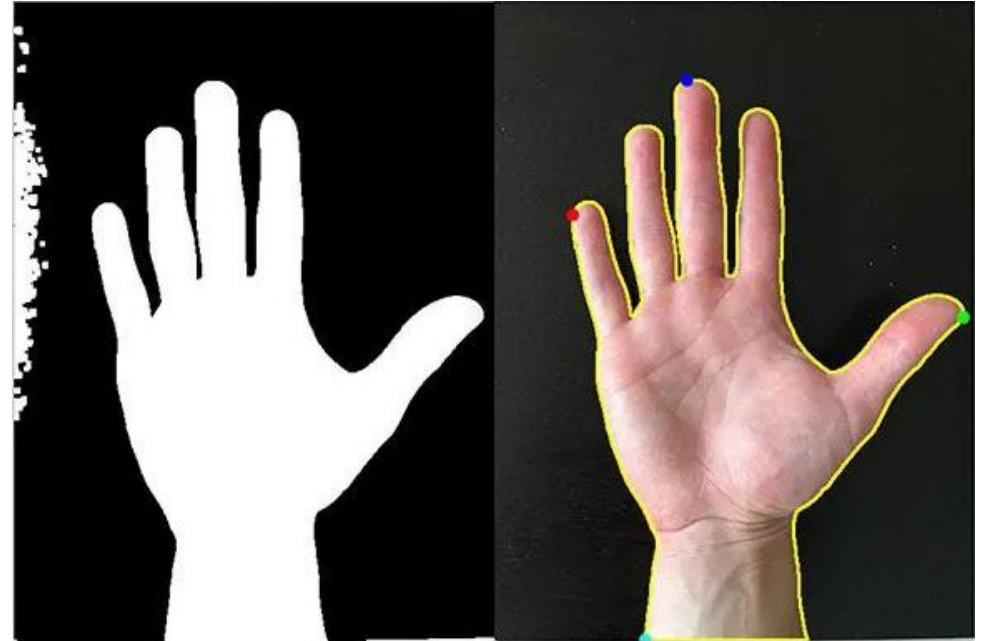
## Applications



## La détection de contours

la détection de contours se ramène à la recherche des discontinuités locales de la fonction des niveaux de gris de l'image.

On peut utiliser (pour la détection de contours) des méthodes différentielles:



- ❑ **Le Gradient:** c'est la dérivée première de la fonction de niveau de gris, le contour correspond à la norme du gradient supérieure à un seuil donné.
- ❑ **Le Laplacien:** c'est la dérivée seconde de la fonction de niveau de gris, le contour correspond au passage par zéro du laplacien.

# Opérations sur les images

## La première dérivée (le Gradient)

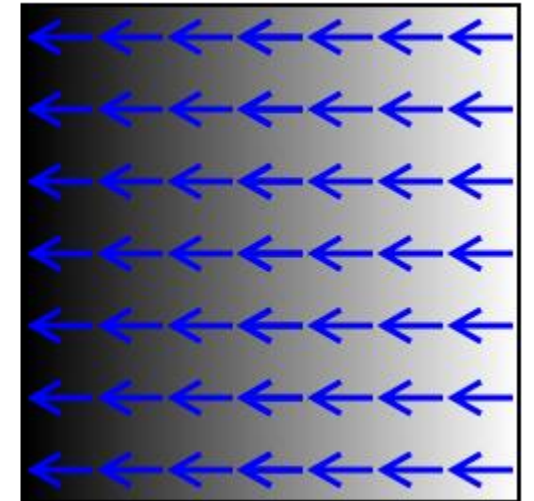
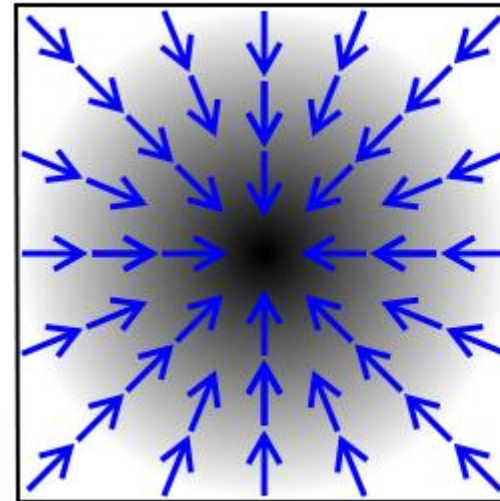
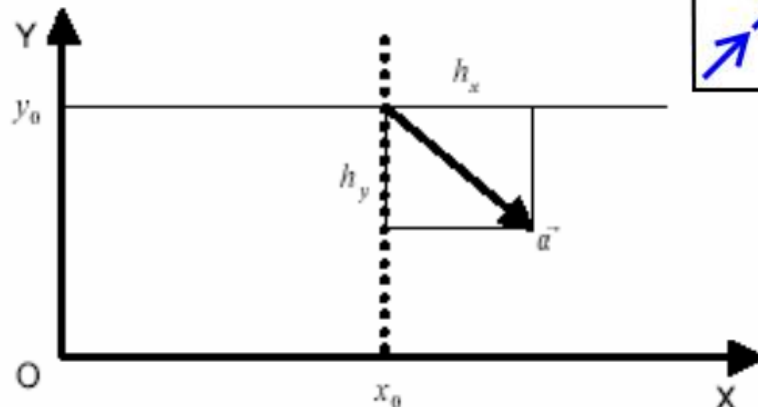
La fonction image  $f(x,y)$  étant définie dans un espace bidimensionnel, nous pouvons définir des dérivées partielles par rapport aux variables de définition de  $f$  :

$$\frac{\partial f(x, y)}{\partial x} = \lim_{hx \rightarrow 0} \frac{f(x + h_x, y) - f(x, y)}{h_x}$$

$$\frac{\partial f(x, y)}{\partial y} = \lim_{hy \rightarrow 0} \frac{f(x, y + h_y) - f(x, y)}{h_y}$$

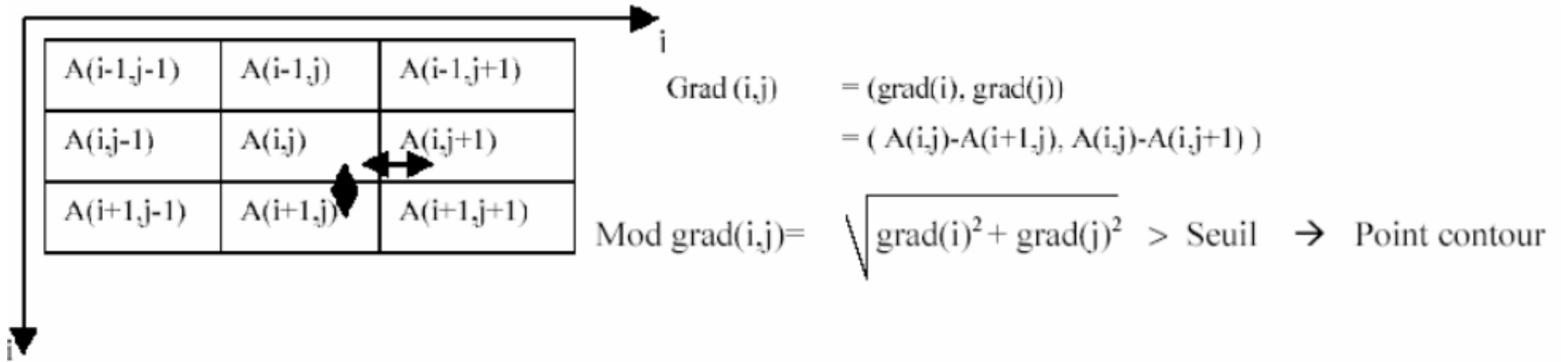
Le vecteur gradient est donné par:

$$\vec{\nabla} f \begin{cases} \frac{\partial f(x, y)}{\partial x} \\ \frac{\partial f(x, y)}{\partial y} \end{cases}$$



# Opérations sur les images

## Exemple de calcul du gradient



200 200 0 0 0  
 200 200 0 0 0  
 200 200 0 0 0  
 200 200 0 0 0  
 200 200 0 0 0

Image source

0 200 0 0 0  
 0 200 0 0 0  
 0 200 0 0 0  
 0 200 0 0 0  
 0 200 0 0 0

Module du gradient

# Opérations sur les images

## Exemple de calcul du gradient



Image initiale

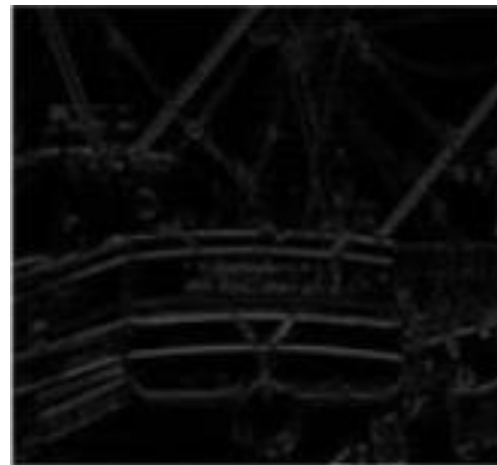
Gradient  $h_x$



Contours verticaux



Gradient  $h_y$



Contours horizontaux



Contours

# Opérations sur les images

## Filtres et calcul du gradient

a) **Le filtre de Prewitt:** Il calcule le gradient de l'intensité de chaque pixel en utilisant la convolution avec l'image pour calculer l'approximation des dérivées horizontales et verticales

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} * A \quad \text{et} \quad G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix} * A \quad \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} [-1 \quad 0 \quad 1]$$

b) **Le filtre de Sobel:** Similaire au filtre de Prewitt, mais avec un lissage pondéré

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} \otimes A \quad \text{et} \quad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \otimes A$$

## La deuxième dérivée (le Laplacien)

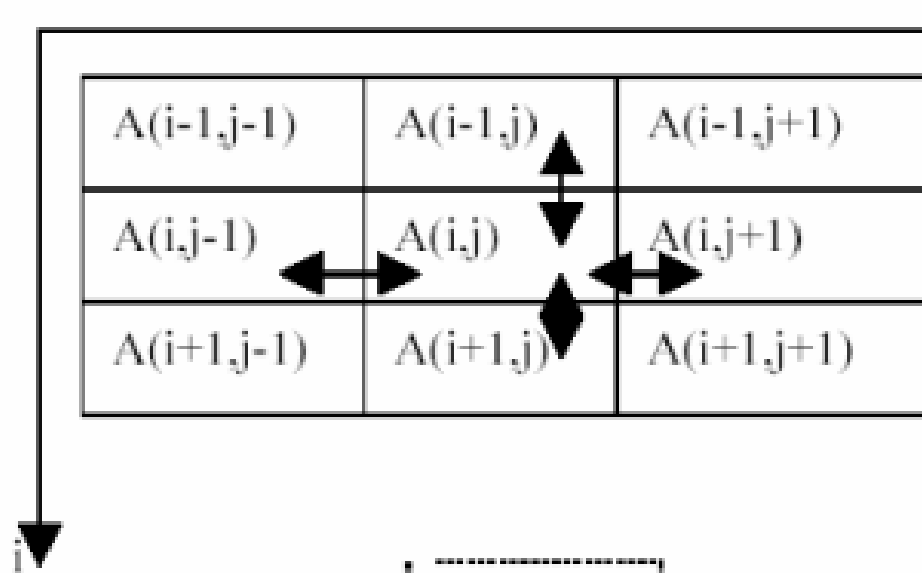
Le Laplacien est une grandeur signée, traduisant de façon sommaire la concavité.

Les points de contours correspondent au passage par zéro.

$$\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

# Opérations sur les images

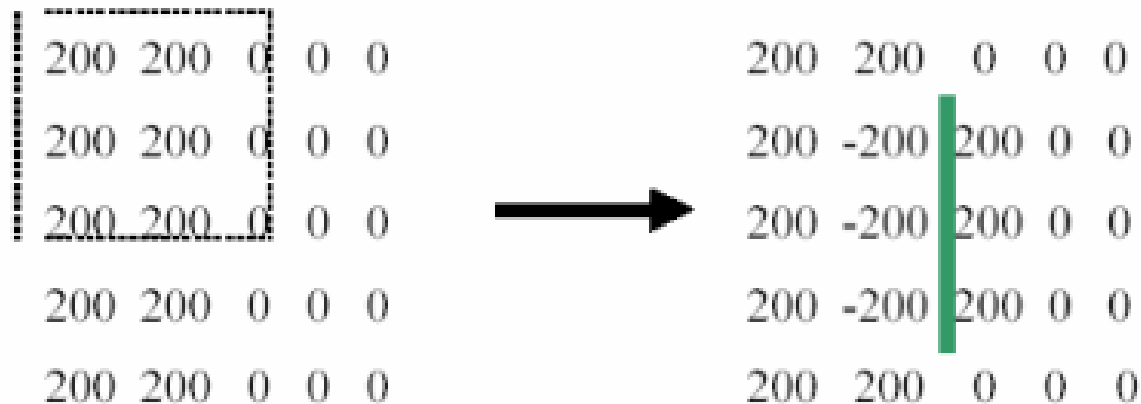
## Exemple de calcul du laplacien



Dans le cas discret :

$$\nabla f(i,j) = -4 A(i,j) + A(i,j+1) + A(i,j-1) + A(i-1,j) + A(i+1,j)$$

$$f(x,y) \otimes \text{Masque} \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



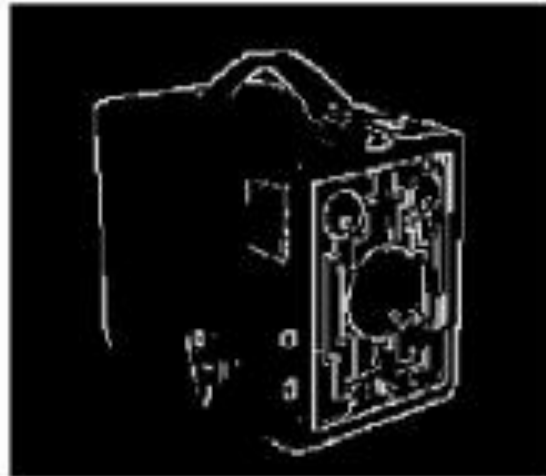
# Opérations sur les images

Exemple de calcul du laplacien



*Image*

*Gradient*



*Laplacien*

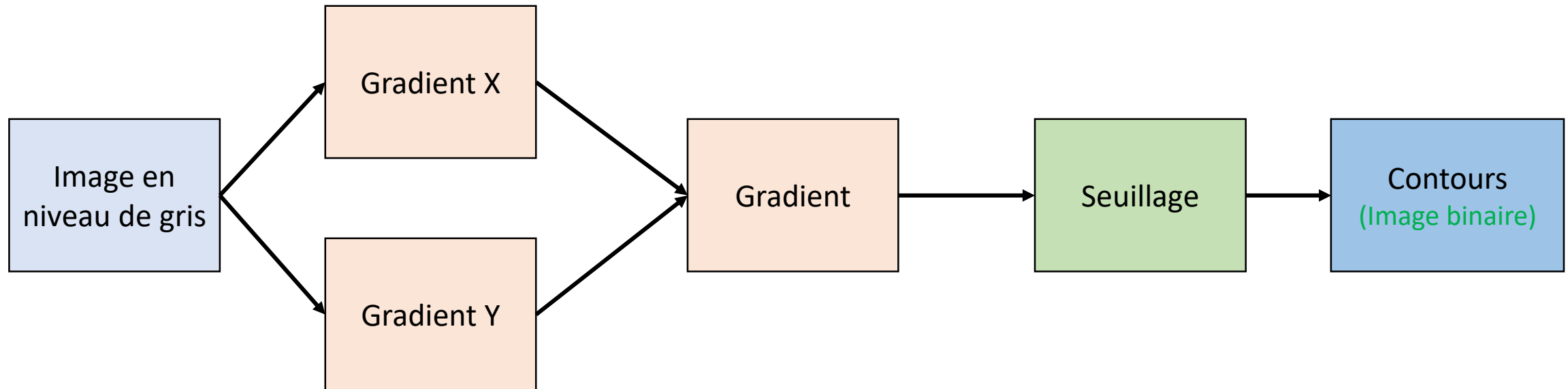


**b) Le filtre de Canny:** Il s'agit tout d'abord de

- lisser l'image, afin d'éliminer le bruit,
- et de calculer dans un second temps le gradient et l'angle de la normale au gradient.

## Seuillage des contours

- Un contour est une zone de transition filiforme entre deux zones homogènes de l'image, appelées Régions,
- Contour est une discontinuité dans l'intensité de l'image,



## Seuillage des contours

Permet de sélectionner les informations significatives dans une images en niveau de gris,

Souvent les filtres utilisés ne suffisent pas pour éliminer le bruit

Image Résultant de ce traitement est **Binaire (0 ou 1)**,

Si la valeur du pixel  $>$  **Seuil**  $\rightarrow$  La valeur résultant  $= 1$

Sinon  $\rightarrow$  La valeur résultant  $= 0$

## Réglage d'un paramètre : **Seuil** :

**Seuil fixe** : une valeur fixe pour le seuil,

**Seuil Global** : la moyenne des pixels de l'image

**Seuil Local** : La moyenne pondérée des pixels voisinages (ex : en appliquant un masque de convolution)

# Opérations sur les images

Seuillage des contours

Problème : Comment calculer le seuil ?

Seuil faible (bas) → détection de vrais point du contours

+ point dus au bruit,

Seuil élevé → supprimer les point dus au bruit +

supprimer les vrais point du contours,



Seuil = 80 (Seuil bas)



Seuil = 100



Seuil = 140



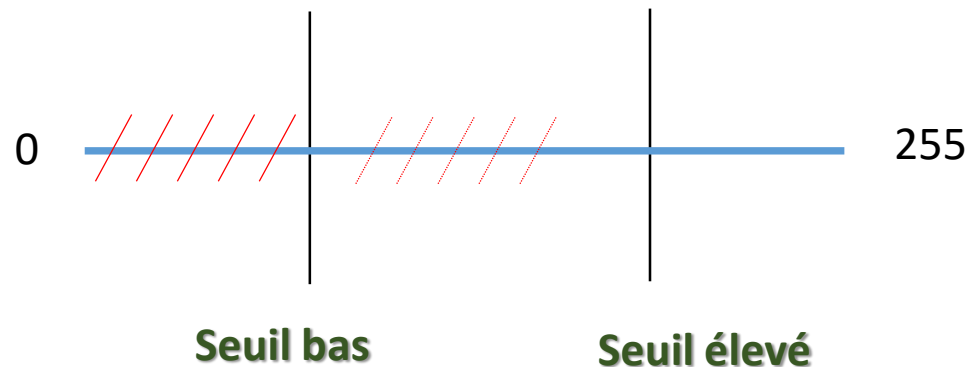
Seuil = 190 (Seuil élevé)

## Seuillage par hystérésis

Consiste à deux seuils à comparer à l'intensité du gradient:

Pour chaque point si l'intensité de son gradient est:

- Inférieur au seuil bas, le point est rejeté
- Supérieur au seuil élevé, le point est accepté comme formant un contour
- Entre les deux seuils, le point est accepté s'il est connecté à un point déjà accepté.



## Suivi de contours

Recherche de chemin optimal dans un graphe

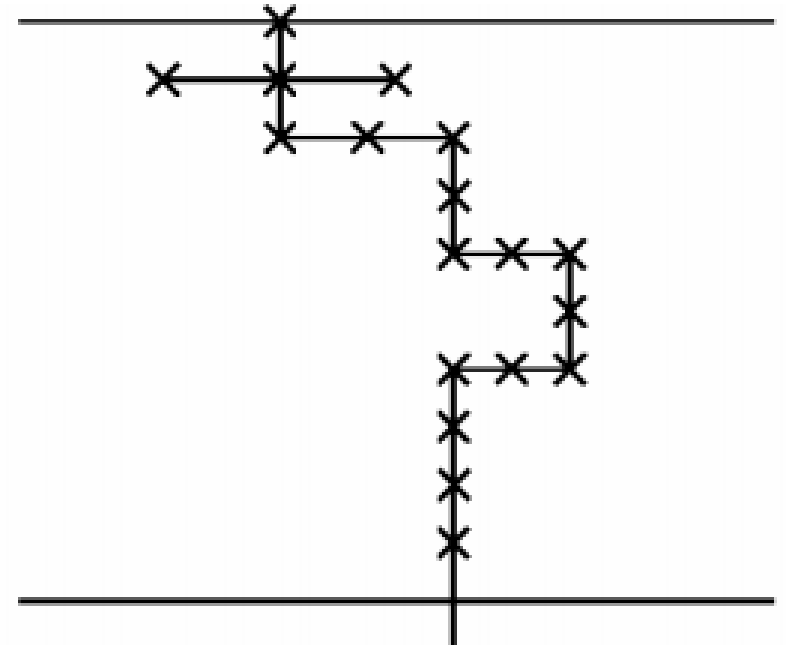
- prendre un bon point
- choix du point suivant
- critère d'arrêt

## Chaînage de contours

Transformer la description matricielle des contours sous forme de listes chaînées.

La structure de données permet la fusion ou suppression de chaînes selon un critère donné (simple jeux de pointeurs).

Possibilité d'une approximation polygonale à partir de la chaîne de contours.



## Segmentation de contours

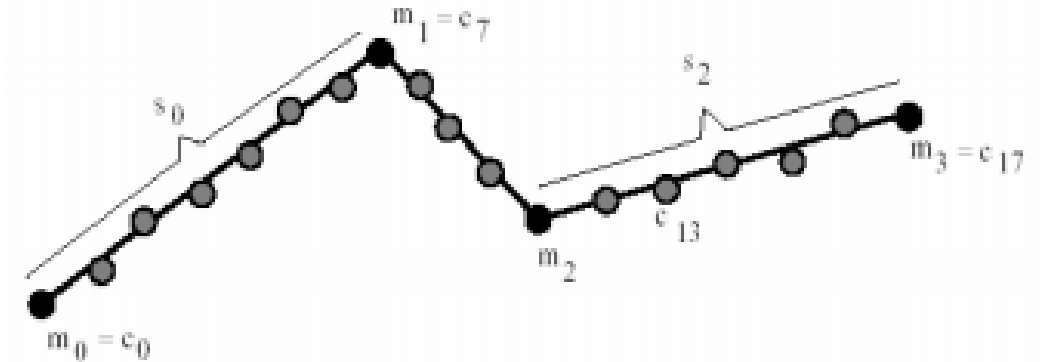
Les chaînes de contours peuvent être partitionnés dans des segments de courbes qui ont une description analytique connue telles **des lignes droites et des coniques**

### a) Segmentation d'une chaîne

$$C = \{c_i, i = 0, 1, \dots, n\}$$

S partition de C,  $S = \{S_0, S_1, S_2, \dots, S_{N-1}\}$

$M = \{m_0, m_1, m_2, \dots, m_N\}$  points de cassure correspondant à la partition S



### Deux problèmes à résoudre:

- Trouver un partitionnement dans la chaîne en segments  $\{S_0, S_1, \dots, S_k, \dots\}$
- Trouver pour chaque segment la meilleure approximation analytique

## Segmentation de contours

### b) Approximation d'un segment par une droite

Segmentation de contours

L'équation de la droite est la distance d'un point  $(X_i, Y_i)$  sont données par:

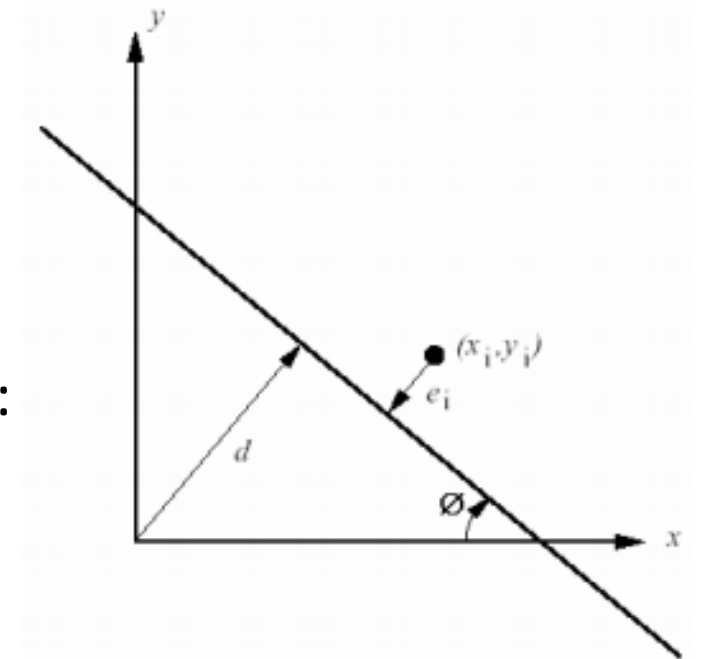
$$x \sin \varnothing + y \cos \varnothing = d$$

La distance d'un point  $(x_i, y_i)$  à cette droite :

$$e_i = |x_i \sin \varnothing + y_i \cos \varnothing - d|$$

trouver la droite (les paramètres  $\varnothing$  et  $d$ ) qui minimise la grandeur:

$$E = \sum_{i=1}^N e_i^2$$



## Segmentation de contours

### c) Approximation par un cercle

Il s'agit de trouver les paramètres du cercle pour représenter la chaîne de contours

$$(x - a)^2 + (y - b)^2 = r^2$$

## Segmentation de contours

### d) Algorithme de découpage récursif (Approximation avec des segments)

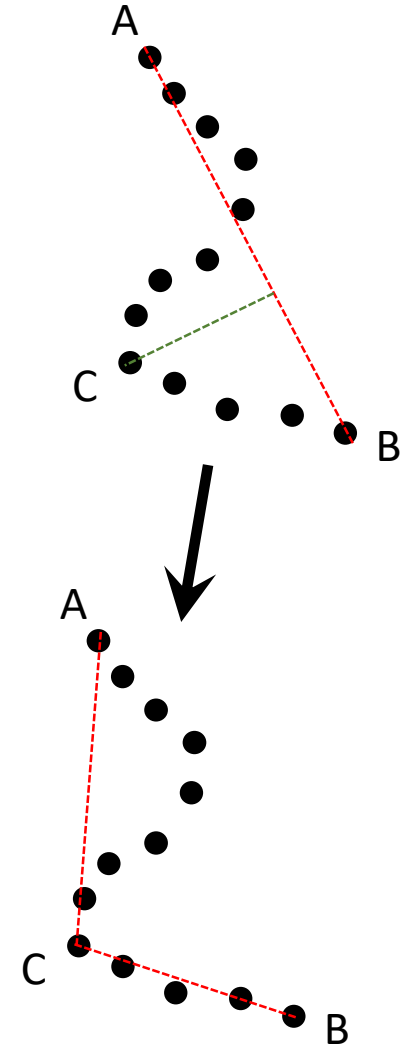
1- pour une chaîne de points: est ce qu'un segment de droite?

2- Si oui aller à FIN

5- sinon diviser la chaîne en deux sous chaînes et répéter pour chaque chaîne

4- FIN

Si mauvaise approximation par un segment, on peut localiser sur le point le plus loin un nouveau point de cassure



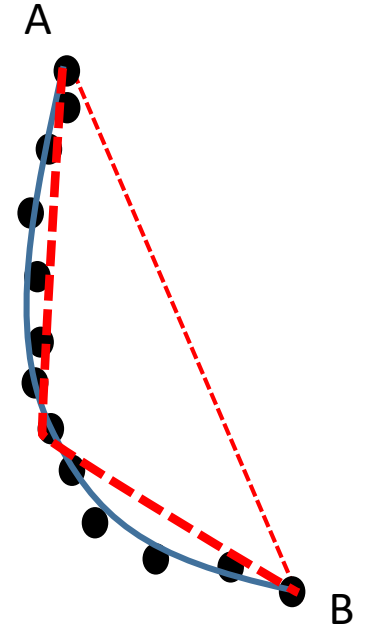
## Segmentation de contours

### e) Algorithme de découpage et union (Segments et des courbes)

Deux problèmes pour cet algorithme:

1- Si la chaîne est approximée correctement par un segment et par une courbe, alors que choisir? (Nécessité de facteur d'évaluation).

2- Les points de cassure choisis par l'algorithme de découpage récursif peuvent couper un arc de cercle en deux parties, il faudrait donc accompagner la procédure de découpage par une procédure d'union de deux chaînes adjacentes



## Segmentation de contours

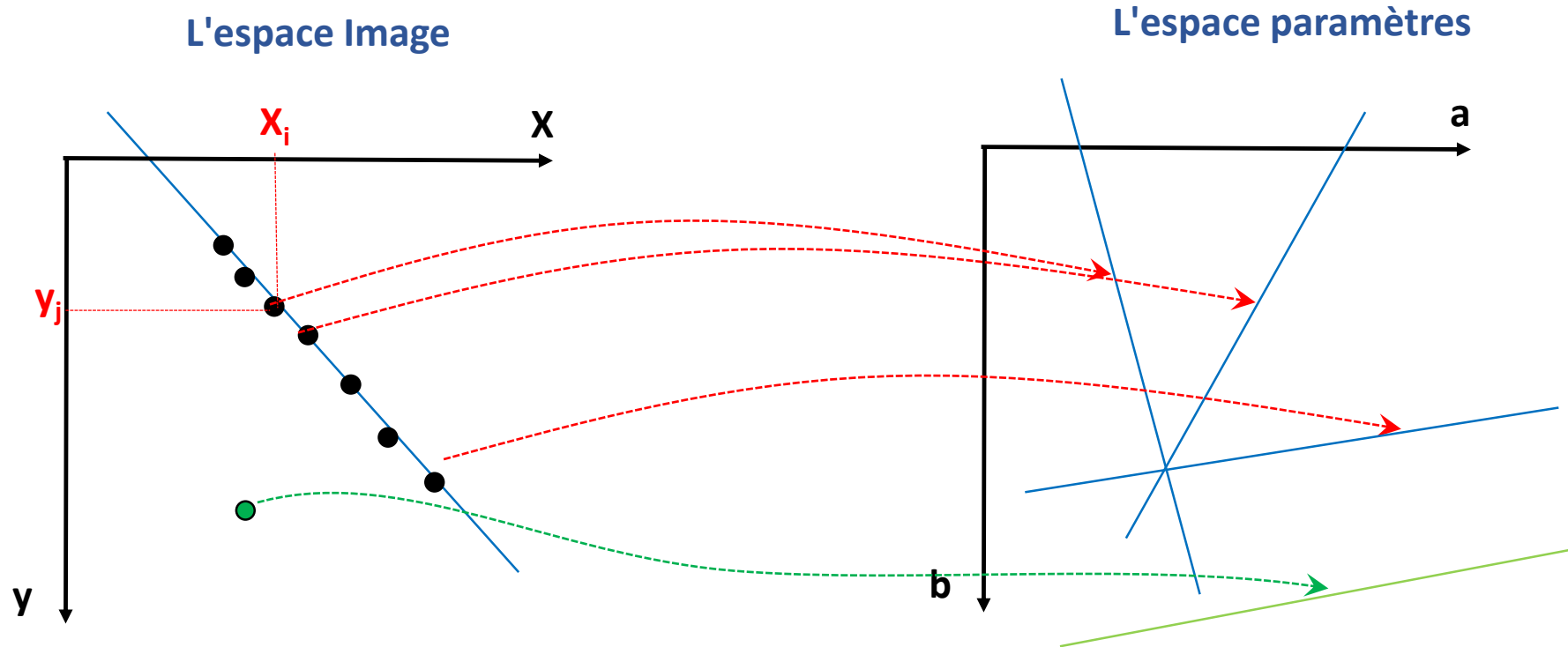
### f) La transformée de Hough

une méthode qui consiste à faire correspondre à des points de contour du plan image des points d'un espace de paramètres.

- A chaque point du contour du plan image correspond un point dans le plan  $(\theta, d)$
- Il est possible de "grouper des points de contour colinéaires sans passer par L'étape de chaînage.

## Segmentation de contours

### f) La transformée de Hough



Donné : ensemble de points  
Tache : calculer la droite  $y = a x + b$

$$b = -a x_i + y_i$$

## Segmentation de contours

L'algorithme de détection des lignes :

1- Quantification de l'espace de paramètres  $(a, b) \rightarrow$  continu vers discret

2- Création d'une matrice d'accumulation  $M(a,b)$

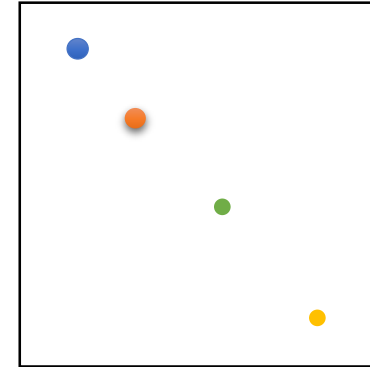
3- Initialisation  $M(a,b) = 0$  pour toutes les valeurs  $a,b$

4- Pour chaque point de la ligne droite  $(x_i, y_j)$  :

$$M(a,b) = M(a,b) + 1$$

5- Rechercher les maximums locaux dans la matrice  $M(a,b)$  :

L'espace Image

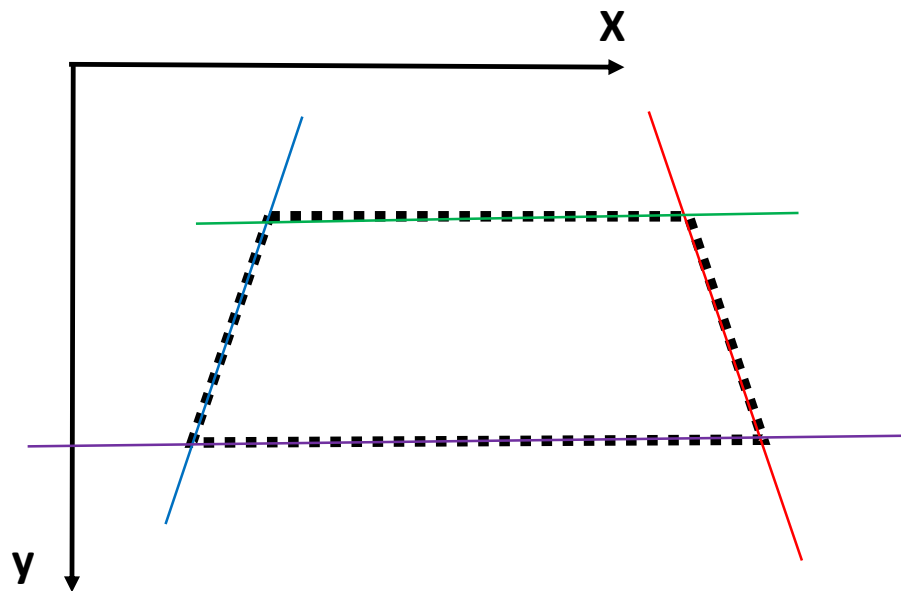


0	0	1	0	0	1
1	0	1	0	1	0
0	1	1	1	0	0
1	1	4	1	1	1
0	1	1	1	0	0
1	0	1	0	1	0

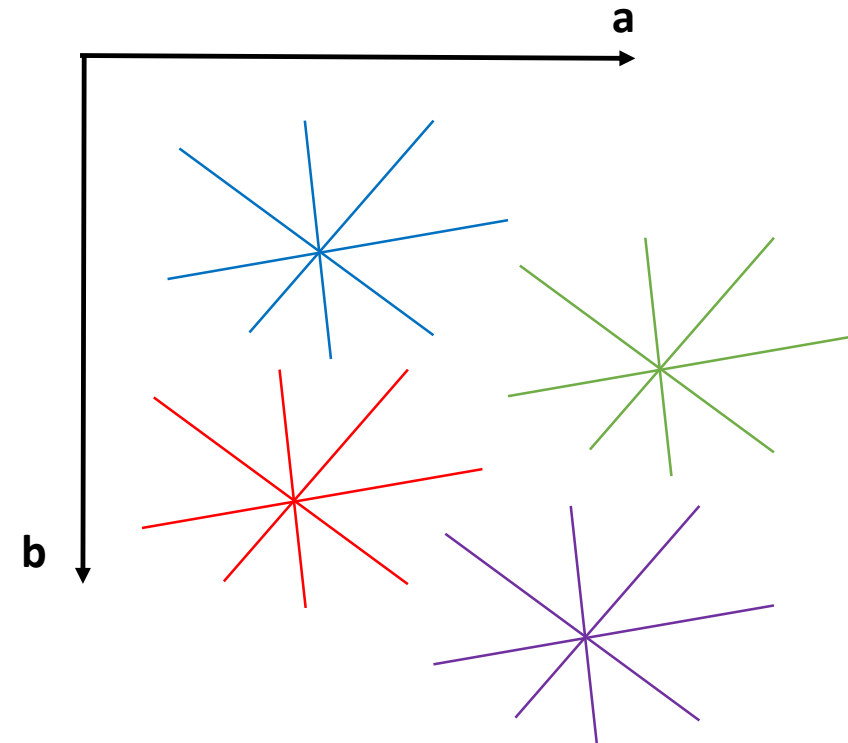
L'espace paramètres  $M(a,b)$

## Segmentation de contours

L'algorithme de détection des lignes :



L'espace Image



L'espace paramètres

## Segmentation de contours

### Le problème de cette modélisation :

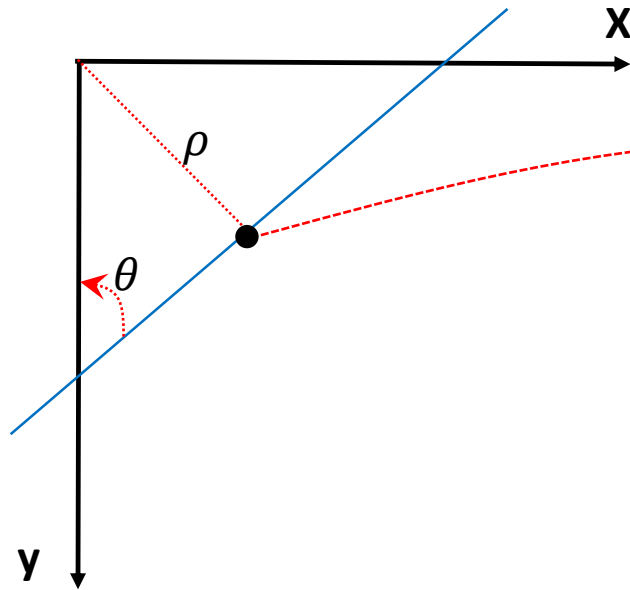
- $-\infty < b < +\infty$
- Une matrice d'accumulation très grande
- Plus complexe en terme du calcul et espace mémoire

### Solution :

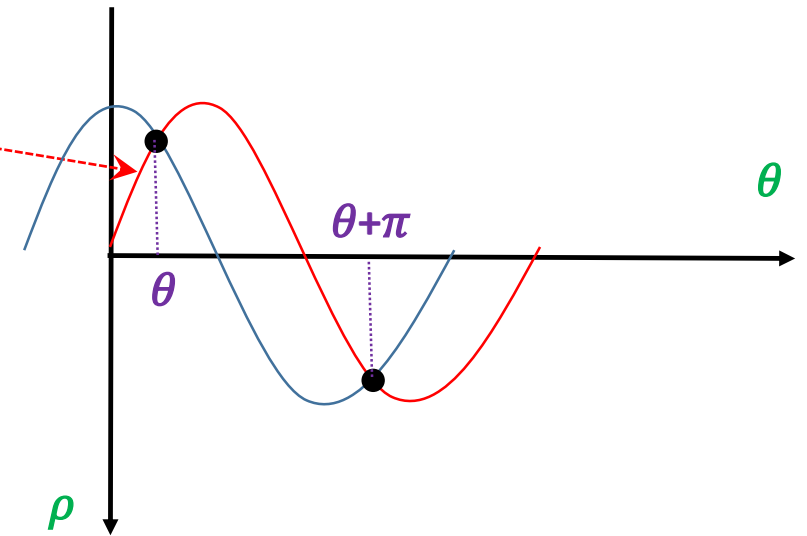
- L'utilisation de l'équation de la droite :  $x \sin \theta - y \cos \theta + \rho = 0$
- L'orientation  $\theta$  est fini :  $0 < \theta < \pi$ .
- Distance  $\rho$  est fini

## Segmentation de contours

Meilleure paramètres de la droite :



$$x \sin \theta - y \cos \theta + \rho = 0$$



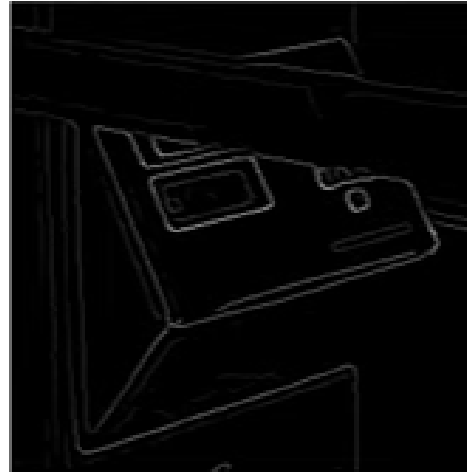
$$x \sin \theta - y \cos \theta + \rho = 0$$

## Segmentation de contours

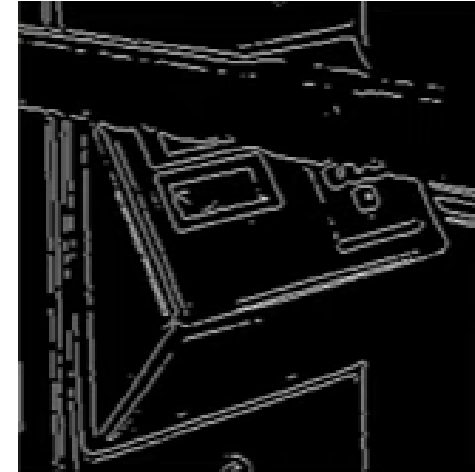
Image originale



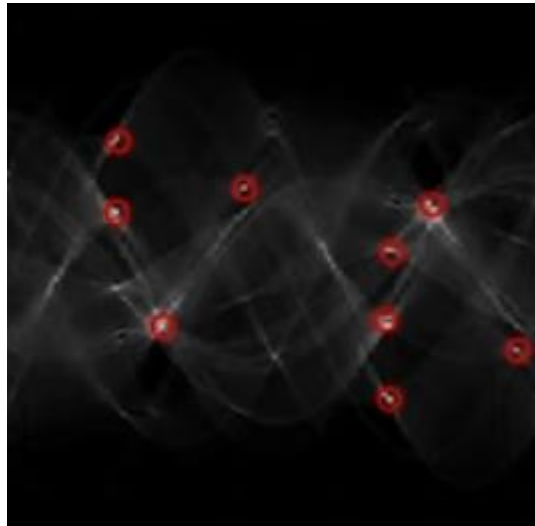
Gradient



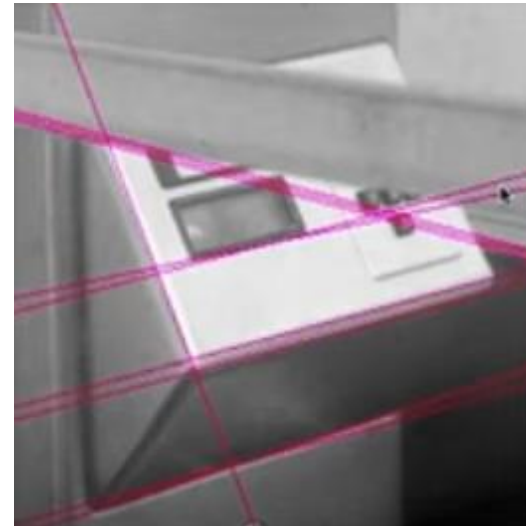
Seuillage



Transformée du  
Hough  $M(\rho, \theta)$



Lignes détectées



# Segmentation d'images

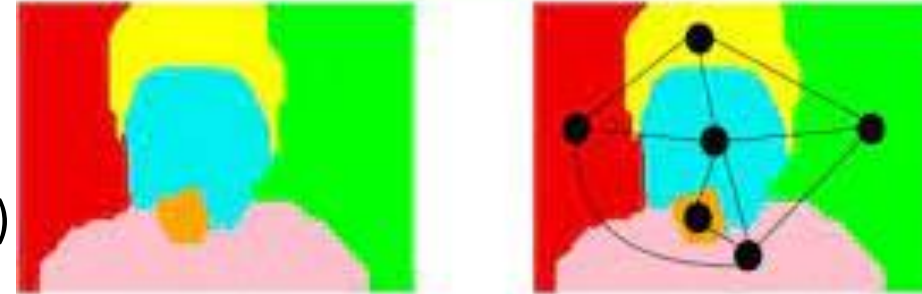
## Segmentation d'images en régions

C'est un problème fondamental en vision par ordinateur,



### On dispose de:

- Ensemble d'entités (point image)
- Ensemble d'attributs caractérisant ces entités (position, luminance,..)



### On cherche:

- Une (ou des) partition(s) de ces données ayant des propriétés intéressantes par rapport aux attributs et aux relations topologiques.

### Les problèmes:

- Définir les propriétés des partitions que l'on cherche
- Concevoir des algorithmes permettant l'obtention de partitions optimisant ces propriétés.

# Segmentation d'images

## Segmentation d'images en régions

- La segmentation consiste alors à regrouper les pixels ayant les mêmes caractéristiques ou les mêmes attributs



Image originale

Seuillage

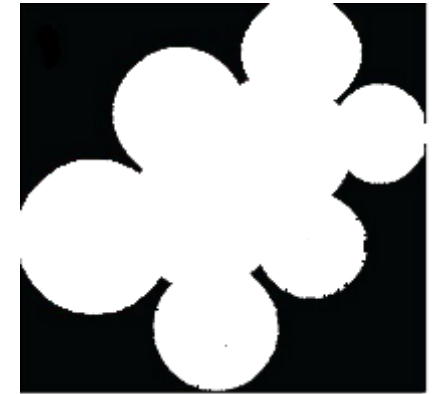


Image binaire

Segmentation



Image

## Segmentation d'images en régions

- La segmentation peut être utilisée comme un problème de clustering (ex : K-Means, Mean-shift, )

La similarité basée sur :

- l'intensité
- La couleur
- La position

À partir de ces caractéristiques nous pouvons calculer d'autres comme :

- La profondeur des points
- La texture
- Type des surfaces
- ...

## Segmentation d'images en régions

### - K-Means :

#### Algorithme K-means :

- K le nombre de cluster à former
- M matrice de données

#### DEBUT

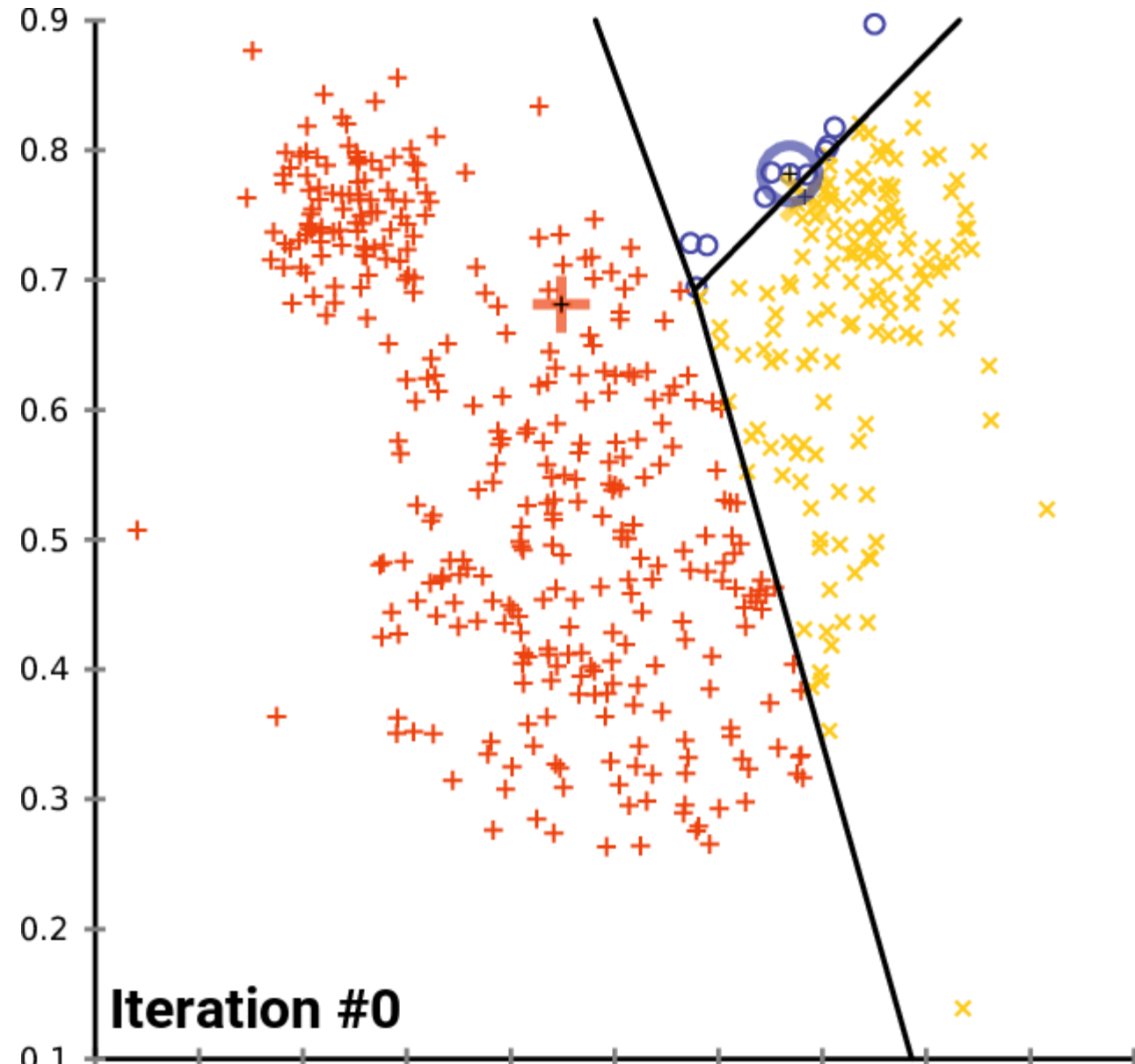
- Choisir aléatoirement K points à partir de M. Ces points sont les centres des clusters,

#### Répéter :

- Affecter chaque point de M au groupe dont il est le plus proche au son centre
- Recalculer le centre de chaque cluster

Jusqu'à la convergence (stabilisation des centres)

Fin.



## Segmentation d'images en régions

### - Mean-shift

**Algorithme mean shift :**

- M matrice de données

**DEBUT**

Pour chaque point de la matrice M

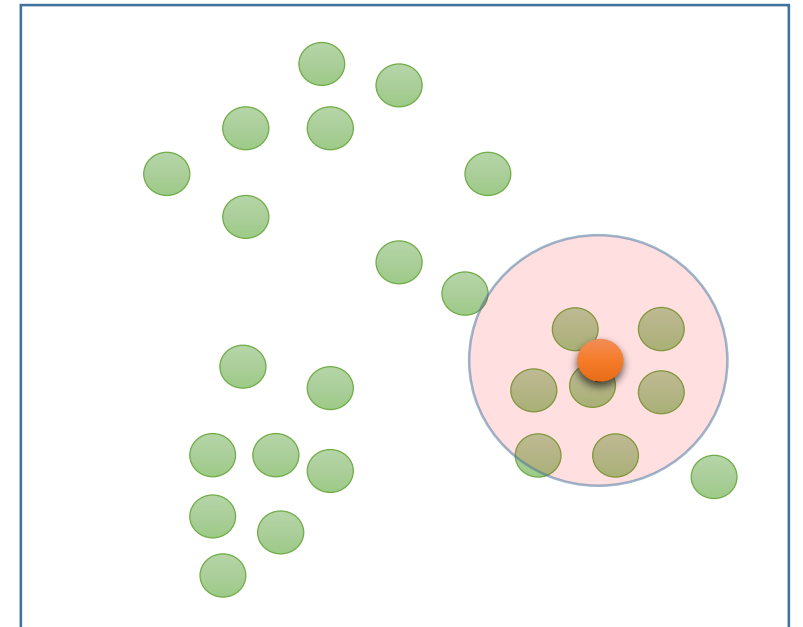
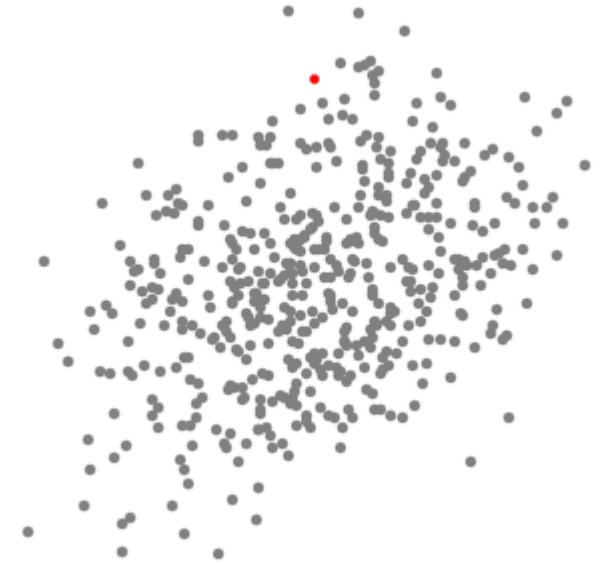
**Répéter :**

- On cherche l'ensemble E des points qui sont dans le voisinage de P.
- On déplace P vers l'isobarycentre de E.

//On réitère depuis l'étape 2 jusqu'à convergence.

**Jusqu'à** la convergence (stabilisation des centres)

**Fin.**



# Segmentation d'images

## Segmentation d'images en régions

- K-Means, Mean-s

Image originale



Image segmentée  
K=16 l'espace (R,V,B)



Image segmentée  
K=16 l'espace (R,V,B,x,y)

# Segmentation d'images

## La segmentation sémantique :

La segmentation sémantique consiste à diviser une image en régions sémantiquement significatives, en attribuant des étiquettes de classe à chaque pixel.

Essentiellement, elle fournit des informations détaillées sur les contours des objets et les régions d'intérêt.



# Segmentation d'images

## La segmentation sémantique :

Exemple : U-Net: Advancing Image Segmentation with Convolutional Neural Networks

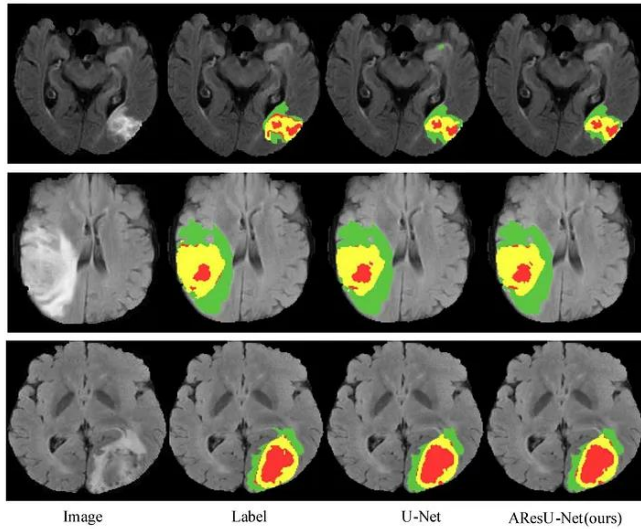
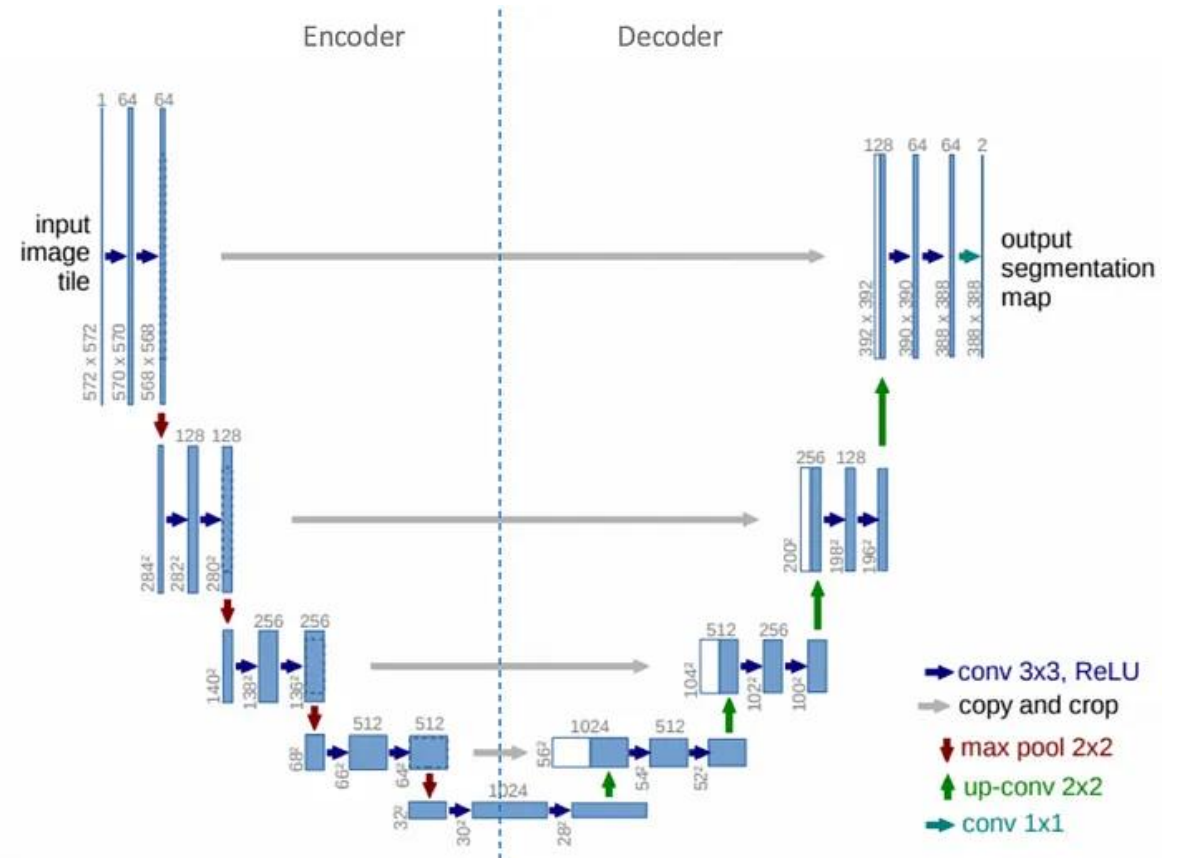


Image Label U-Net AResU-Net(ours)



## Segmentation d'images en régions

Segmentation d'images en régions

On peut aussi utiliser la texture

