

TRAITEMENT D'IMAGES

Partie Introductive

Frédéric Cointault
Institut Agro Dijon
Responsable Equipe ATIP
UMR Agroécologie
26 Bd Dr Petitjean
21000 Dijon
+33 3 80 77 27 54
frederic.cointault@agrosupdijon.fr

L'INSTITUT NATIONAL D'ENSEIGNEMENT SUPÉRIEUR POUR L'AGRICULTURE, L'ALIMENTATION ET L'ENVIRONNEMENT



0 - Préambule

I - Introduction

II - Définitions

III - Pré-traitement des images

IV - Segmentation image et contours

V - Hough et morphologie mathématique

VI – Analyse et Reconnaissance de formes

VII – Détection de mouvement

VIII – Introduction au Deep Learning

III – PRE-TRAITEMENT DES IMAGES

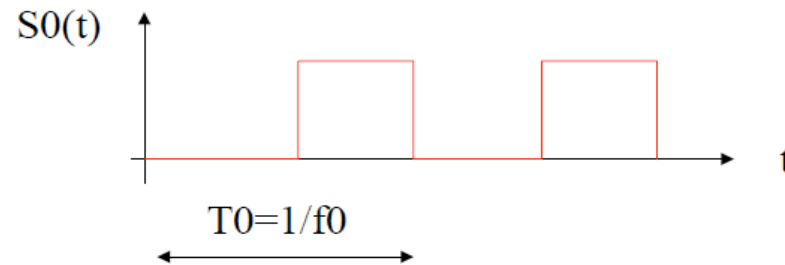
III – 4 Filtres numériques fréquentiels

*** Représentations temporelles, spatiales et fréquentielles:**

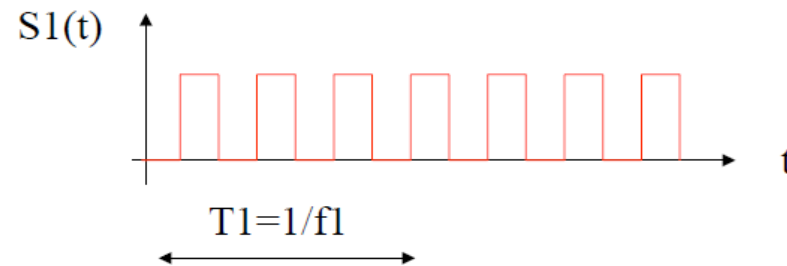
- Pour un signal : Amplitude des variations du signal
(tension, courant, pression, ...) en fonction du **Temps** (t)
- Pour une image: Amplitude des variations de l'Image (niveaux de gris)
en fonction des **variables Spatiales** (X,Y)
- Plusieurs Images: Amplitude des variations de l'Image (niveaux de gris)
en fonction des **variables Spatiales** et **Temps** (X,Y, t)

III – PRE-TRAITEMENT DES IMAGES

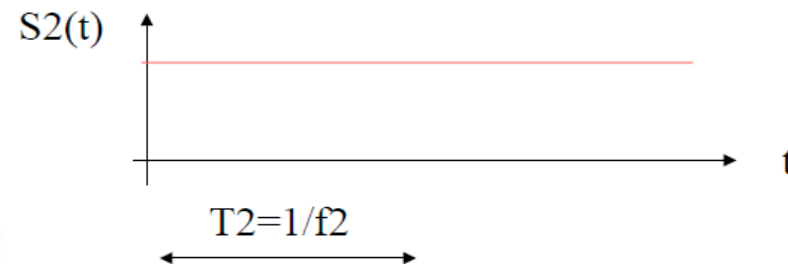
Trois cas de signaux:



avec:
 S_0 = Amplitude du signal
 T_0 = Période du signal
 f_0 = Fréquence du signal

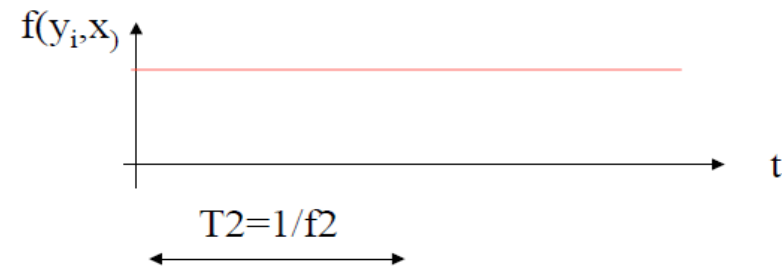
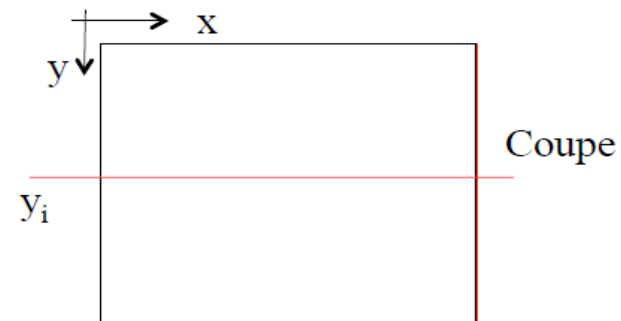
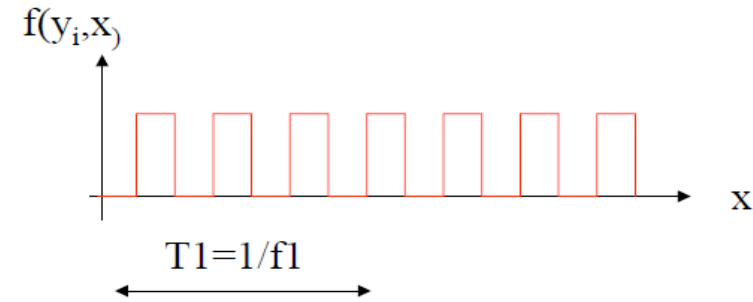
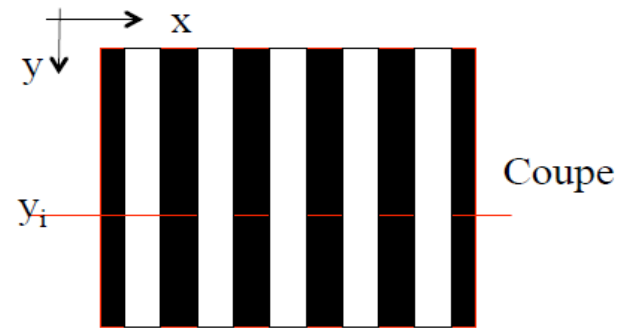
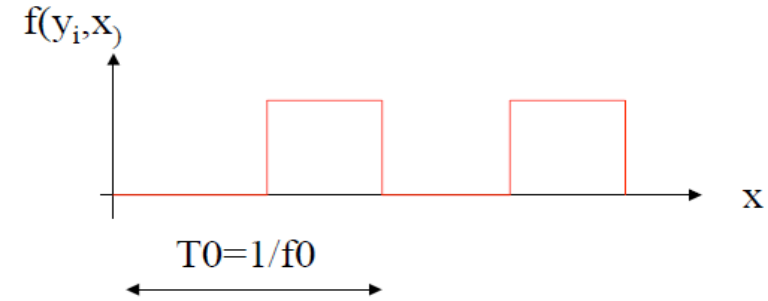
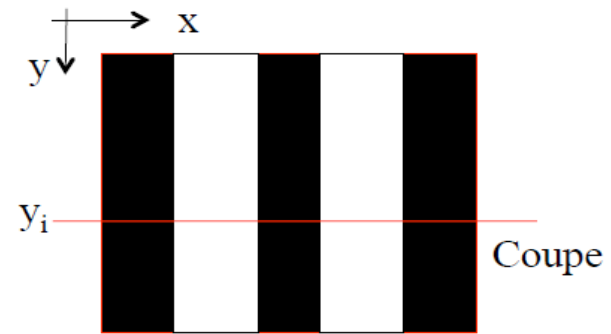


S_1 présente de nombreuses variations:
 $T_1 \ll T_0$ and $f_1 \gg f_0$
Donc f_1 correspond
aux hautes fréquences



S_2 ne présente aucune variation:
est infini and $f_2 = 0$
Donc f_2 correspond
aux très basses fréquences (ici 0!)

III – PRE-TRAITEMENT DES IMAGES



III – PRE-TRAITEMENT DES IMAGES

Fréquences et images:

- * Faibles variations en niveaux de gris (ex: Fond Image):

Basses Fréquences

- * Grandes variations en niveaux de gris (ex: Contours, Bruit):

Hautes Fréquences

- * Représentation fréquentielle obtenue avec la

TRANSFORMEE de FOURIER 2D

- * Avantage: Représentation dans domaine spatial:

Filtrage = Convolution: $g(x,y)=h(x,y)*f(x,y)$

- * Représentation dans domaine fréquentiel :

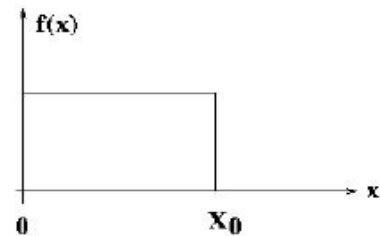
Filtrage = Simple Multiplication: $G(u,v)=H(u,v).F(u,v)$

III – PRE-TRAITEMENT DES IMAGES

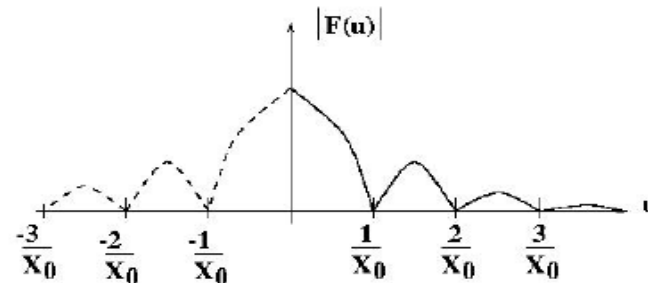
$$F(u) = \int_{-\infty}^{+\infty} f(x) \cdot \exp(-j2\pi ux) dx \quad \text{avec } u : \text{Fréquence}$$

TF 1D d'un signal
analogique

EXAMPLE:

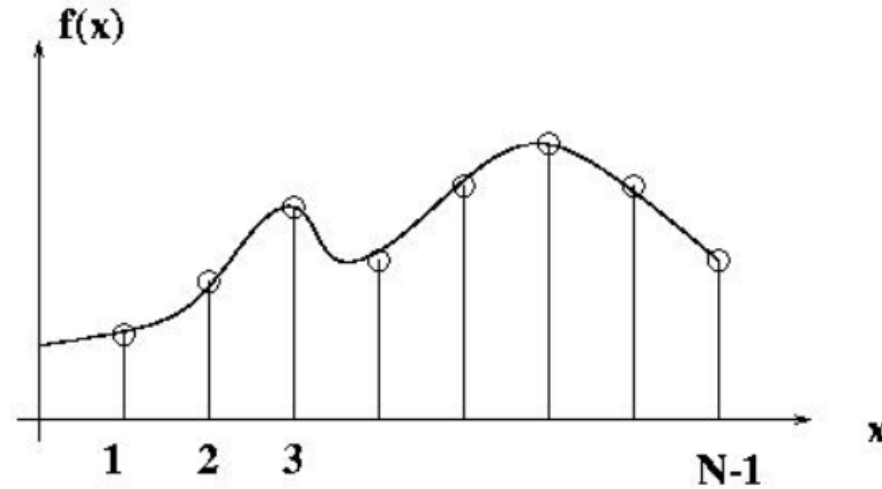


$$|F(u)| = A \cdot X_0 \left| \frac{\sin(\pi u X_0)}{\pi u X_0} \right|$$



III – PRE-TRAITEMENT DES IMAGES

Signal numérique
1D avec N
échantillons



$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \cdot \exp\left(\frac{-j2\pi ux}{N}\right) \quad \text{avec } u = 0, 1, 2, \dots, N-1$$

La transformée de Fourier 1D nécessite N^2 calculs

Alors que la Transformée de Fourier rapide (FFT-1D) nécessite $N \log_2 N$ calculs

III – PRE-TRAITEMENT DES IMAGES

→ **Correspond à un Signal Numérique 2D**

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \cdot \exp(-j2\pi(\frac{ux}{M} + \frac{vy}{N}))$$

avec $u = 0,1,2,\dots,M-1$ et $v = 0,1,2,\dots,N-1$

**Application à une
image de M*N pixels**

→ *Correspond à $M^2 \cdot N^2$ calculs*

avec FFT 2D : $M \log_2 M \cdot N \log_2 N$ calculs

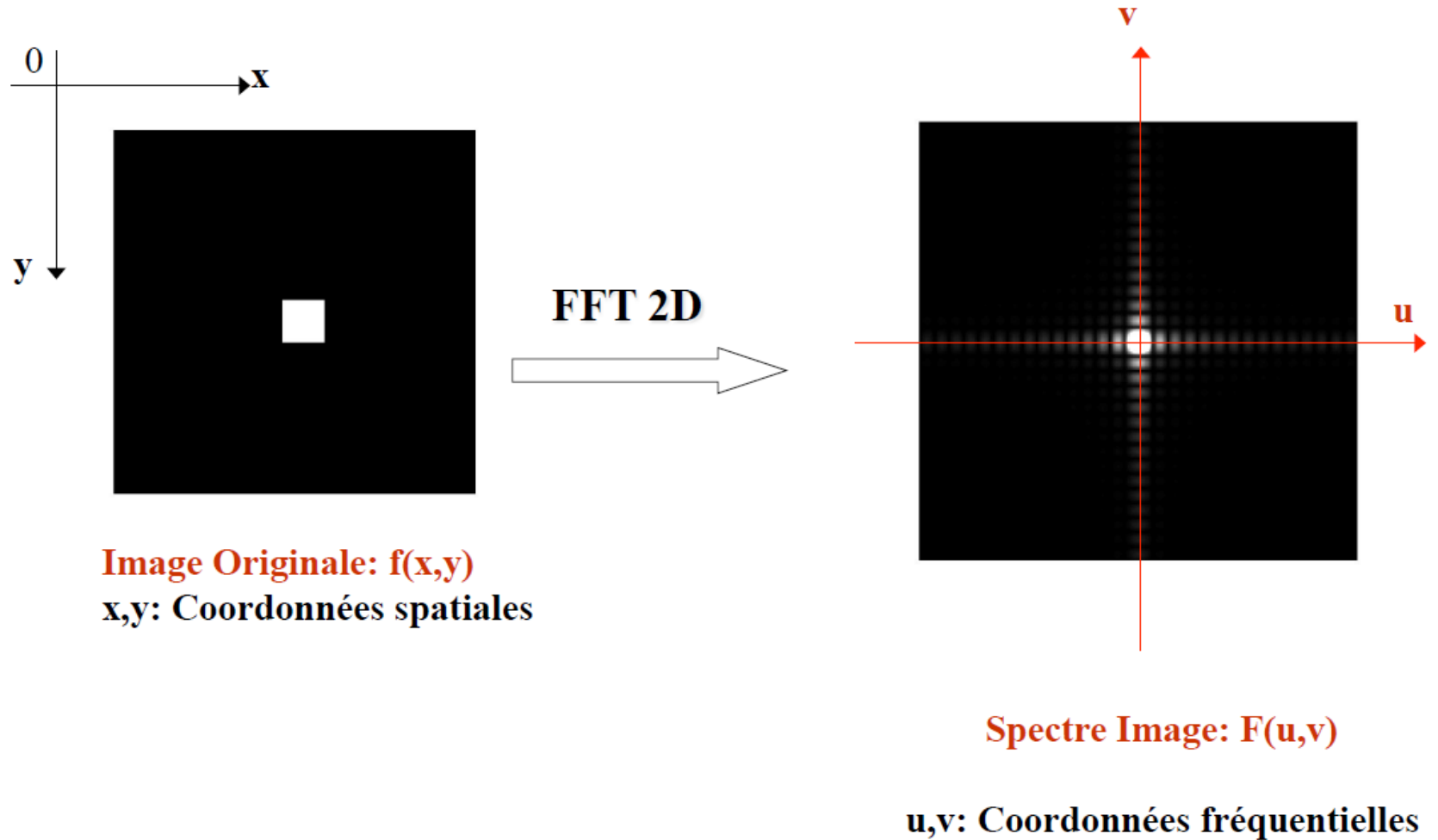
→ *Exemple : $M = N = 512$*

Transformée de Fourier Standard : $\approx 64 \cdot 10^9$ calculs

Transformée de Fourier Rapide (FFT) : $\approx 16 \cdot 10^6$ calculs

III – PRE-TRAITEMENT DES IMAGES

Exemple FFT 2D

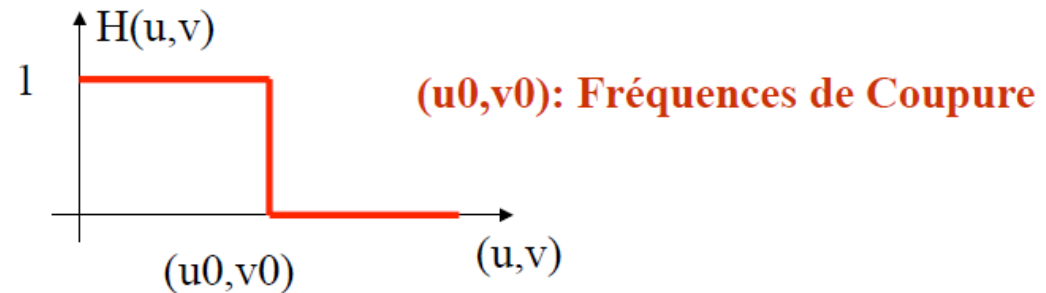


III – PRE-TRAITEMENT DES IMAGES

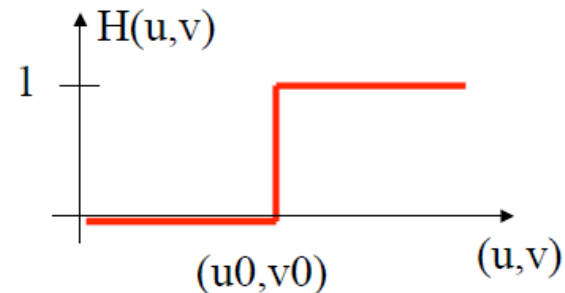
$$G(u,v)=H(u,v).F(u,v)$$

avec : $G(u,v)$: Image Filtrée $F(u,v)$: Image Originale $H(u,v)$: Filtre

Filtre Passe-Bas Idéal: $G(u,v)=F(u,v)$ si $(u,v)<(u_0,v_0)$



Filtre Passe-Haut Idéal: $G(u,v)=F(u,v)$ si $(u,v)>(u_0,v_0)$



Filtre idéal dans le
domaine
fréquentiel

III – PRE-TRAITEMENT DES IMAGES

Exemple de filtre
passe-bas

Image Bruitée Originale

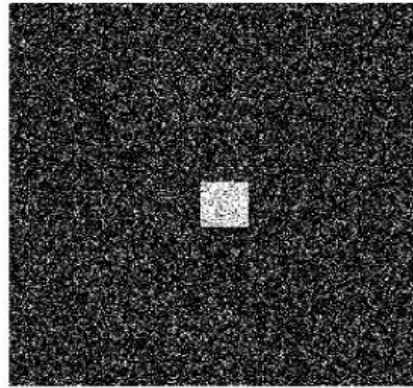
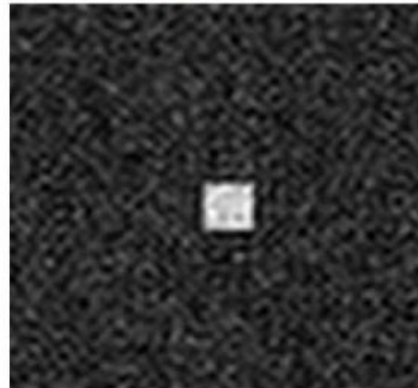
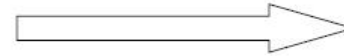


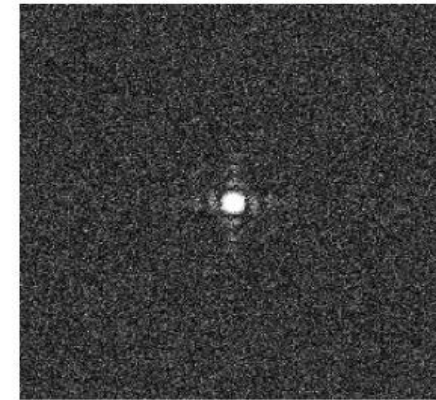
Image Filtrée



FFT-2D



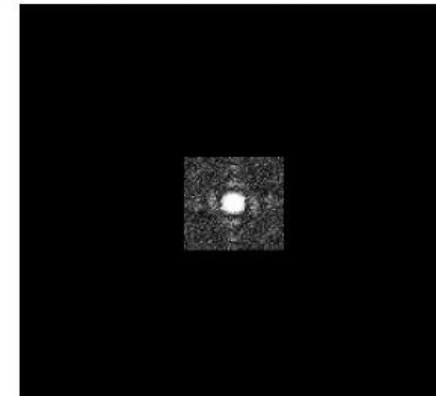
SPECTRE



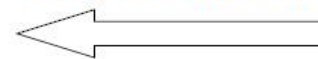
Hautes
Fréquences=0



SPECTRE APRES FILTRAGE



FFT-2D Inverse



III – PRE-TRAITEMENT DES IMAGES

Masque de Sobel

$$h_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \text{ et } h_y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$



Original



Noyau [-1 1]



Noyau [-1 0 1]



Gradient horizontal (Sobel)



Gradient vertical (Sobel)



Module du gradient de Sobel

III – PRE-TRAITEMENT DES IMAGES

TYPES DE MASQUE	GRADIENTS PARTIELS	AMPLITUDE	DIRECTION
<p><i>Masques de Roberts</i></p> <div> <div> -1 0 0 1 </div> <div> 0 -1 1 0 </div> </div>	G_1, G_2 Substitution du pixel supérieur gauche	$A = \sqrt{G_1^2 + G_2^2}$	$\theta = \frac{\pi}{4}$ + $\arctan\left(\frac{G_2}{G_1}\right)$
<p><i>Masques de Sobel</i></p> <div> <div> 1 0 -1 2 0 -2 1 0 -1 </div> <div> 1 2 1 0 0 0 -1 -2 -1 </div> </div>	G_x, G_y	$A = \sqrt{G_x^2 + G_y^2}$	$\theta = \arctan\left(\frac{G_y}{G_x}\right)$
<p><i>Masques de Prewitt</i></p> <div> <div> 1 0 -1 1 0 -1 1 0 -1 </div> <div> 1 1 1 0 0 0 -1 -1 -1 </div> </div>	G_x, G_y	$A = \sqrt{G_x^2 + G_y^2}$	$\theta = \arctan\left(\frac{G_y}{G_x}\right)$

<p><i>Masques de Kirsh</i></p> <div> 5 5 5 -3 0 -3 -3 -3 -3 </div> <p>+ les 7 autres masques obtenus par permutation circulaire des coefficients</p>	G_i pour i de 1 à 8	maximum des $ G_i $	Direction correspondant au G_i sélectionné
<p><i>Masques de Robinson</i></p> <div> 1 1 1 1 -2 1 -1 -1 -1 </div> <p>+ les 7 autres masques obtenus par permutation circulaire des coefficients</p>	G_i pour i de 1 à 8	maximum des $ G_i $	Idem
<p><i>Laplacien discret</i></p> <div> 1 1 1 1 -8 1 1 1 1 </div>	<p><i>Laplacien de Robinson</i></p> <div> 1 -2 1 -2 4 -2 1 -2 1 </div>		



0 - Préambule

I - Introduction

II - Définitions

III - Pré-traitement des images

IV - Segmentation image et contours

V - Hough et morphologie mathématique

VI – Analyse et Reconnaissance de formes

VII – Détection de mouvement

VIII – Introduction au Deep Learning

IV – Segmentation des images

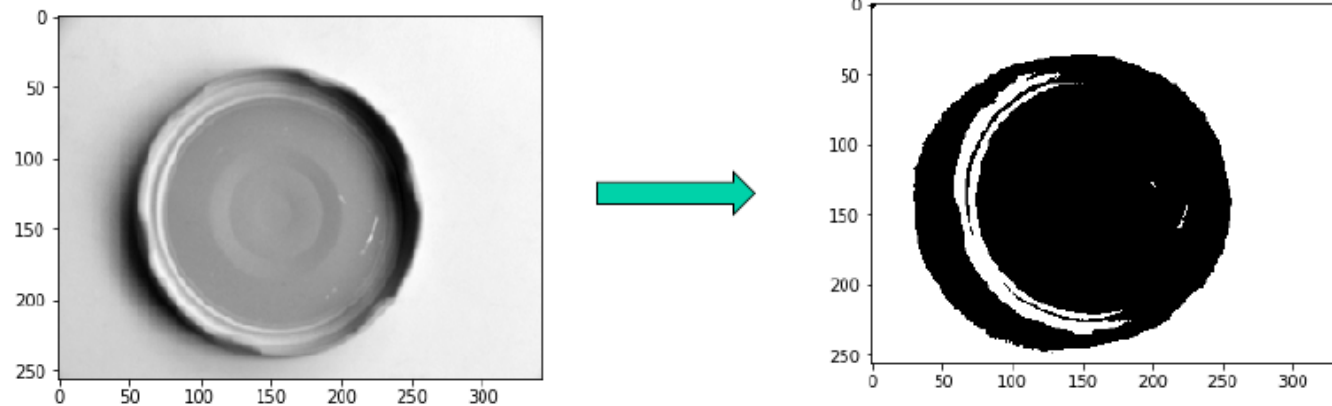
IV-I Segmentation par les régions

IV-2 Segmentation par les contours

IV – Segmentation des images

Segmentation par les régions

- Binarisation (cf chap III):

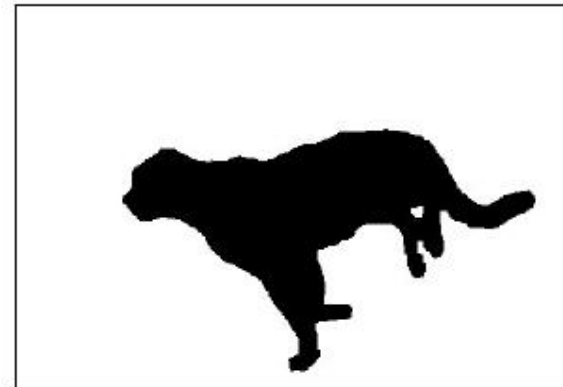


- Morphologie mathématique (cf chap VI)
- Clustering (algorithme K-Means)
- Mask-RCNN (Deep Learning)

IV – Segmentation des images

- Clustering (algorithme K-Means):

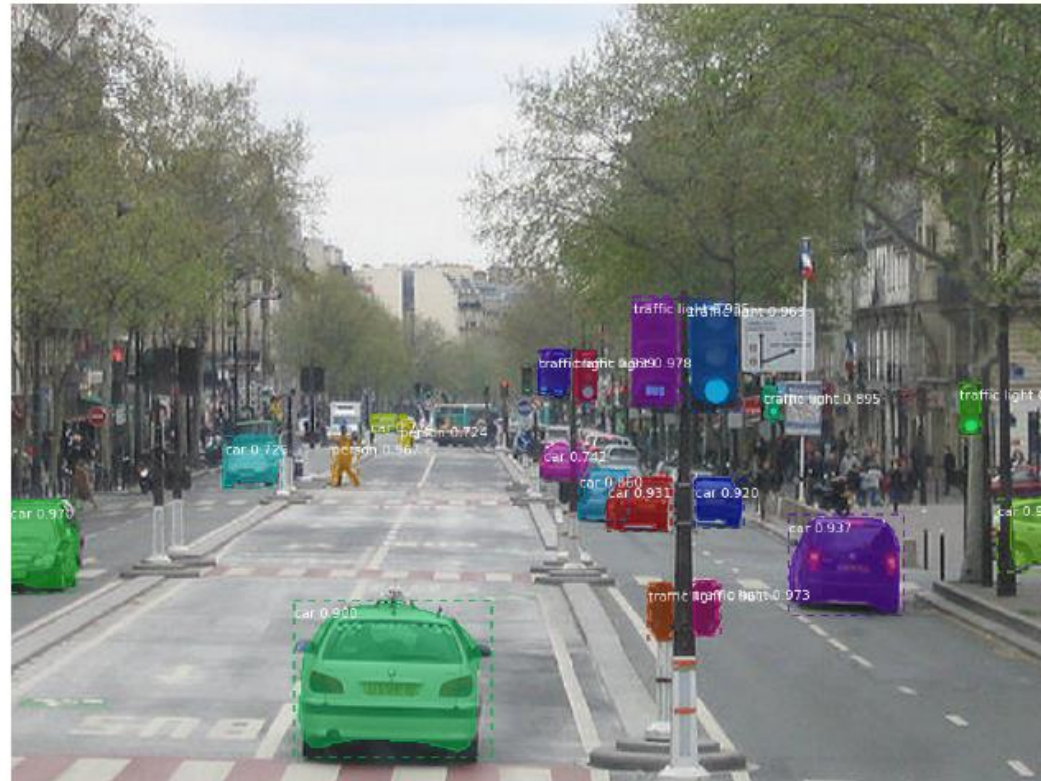
<https://github.com/suhas-nithyanand/Image-Segmentation-using-K-Means>



IV – Segmentation des images

- **Mask-RCNN (Deep Learning):**

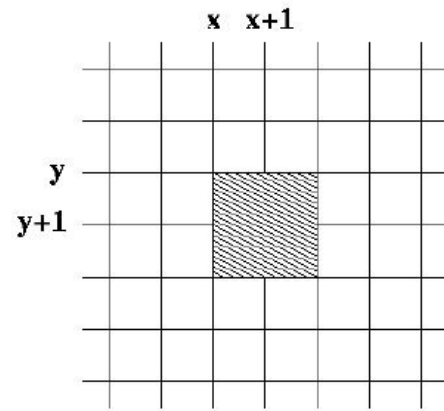
https://github.com/matterport/Mask_RCNN



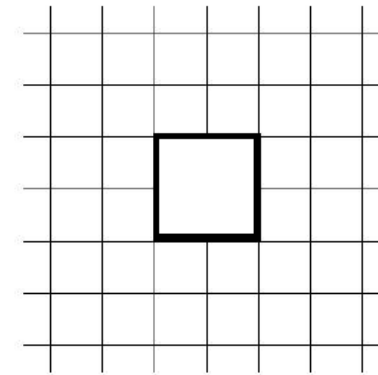
IV – Segmentation des images

Segmentation par les contours

→ Approche « Dérivée »

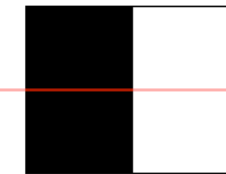


Original Image



Edge Detection

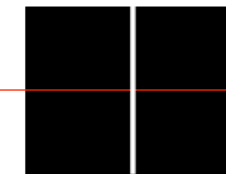
(Contours of the Image)



0 0 0 1 1 1 1

$f(x) - f(x-1)$

0 0 0 1 0 0 0



Exemple Gradient de Roberts :

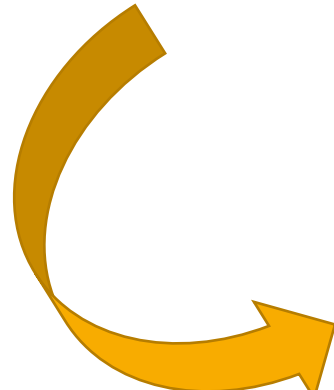
$$g(x, y) = |f(x, y) - f(x+1, y+1)| + |f(x+1, y) - f(x, y+1)|$$

IV – Segmentation des images

Exemple du Gradient de Roberts:

$$g(x, y) = |f(x, y) - f(x+1, y+1)| + |f(x+1, y) - f(x, y+1)|$$

Image d'un carré de 8x8 Pixels



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

Image « Contours »

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

IV – Segmentation des images

Généralisation des opérateurs de contours:

- Opérateur de Roberts: 2 Masques de Convolution

$$\text{hr1} = \begin{vmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{vmatrix} \qquad \text{hr2} = \begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{vmatrix}$$

$$\begin{aligned} g(x, y) &= |f(x, y) - f(x+1, y+1)| + |f(x+1, y) - f(x, y+1)| = \\ &= |Hr1 * f(x, y)| + |Hr2 * f(x, y)| \end{aligned}$$

- Opérateur de Prewitt: 2 Masques de Convolution

$$\text{hp1} = \begin{vmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{vmatrix} \qquad \text{hp2} = \begin{vmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{vmatrix}$$

$$g(x, y) = |Hp1 * f(x, y)| + |Hp2 * f(x, y)|$$

IV – Segmentation des images

- Opérateur de Sobel: 2 Masques de Convolution

$$Hs1 = \begin{vmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{vmatrix} \qquad Hs2 = \begin{vmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{vmatrix}$$

$$g(x, y) = |Hs1 * f(x, y)| + |Hs2 * f(x, y)|$$

Filtres de Roberts, Prewitt, Sobel efficaces pour des images peu bruitées

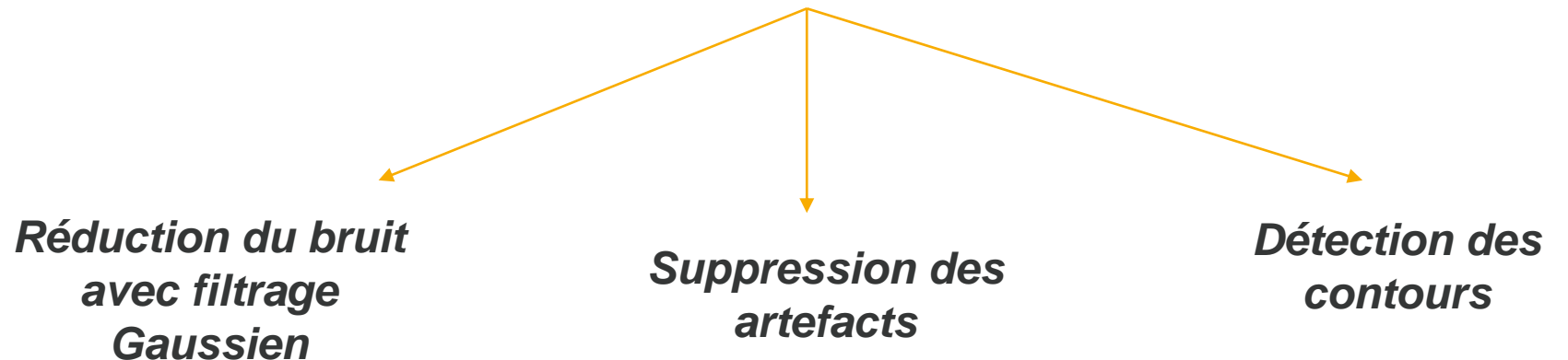
Pour des images bruitées : filtre de Canny ou filtrage passe-bas suivi de Sobel

IV – Segmentation des images

Trois critères à prendre en considération :

1. *bonne détection* : faible taux d'erreur dans la signalisation des contours,
2. *bonne localisation* : minimisation des distances entre les contours détectés et les contours réels,
3. *clarté de la réponse* : une seule réponse par contour et pas de faux positifs

Filtre de Canny





0 - Préambule

I - Introduction

II - Définitions

III - Pré-traitement des images

IV - Segmentation image et contours

V - Hough et morphologie mathématique

VI – Analyse et Reconnaissance de formes

VII – Détection de mouvement

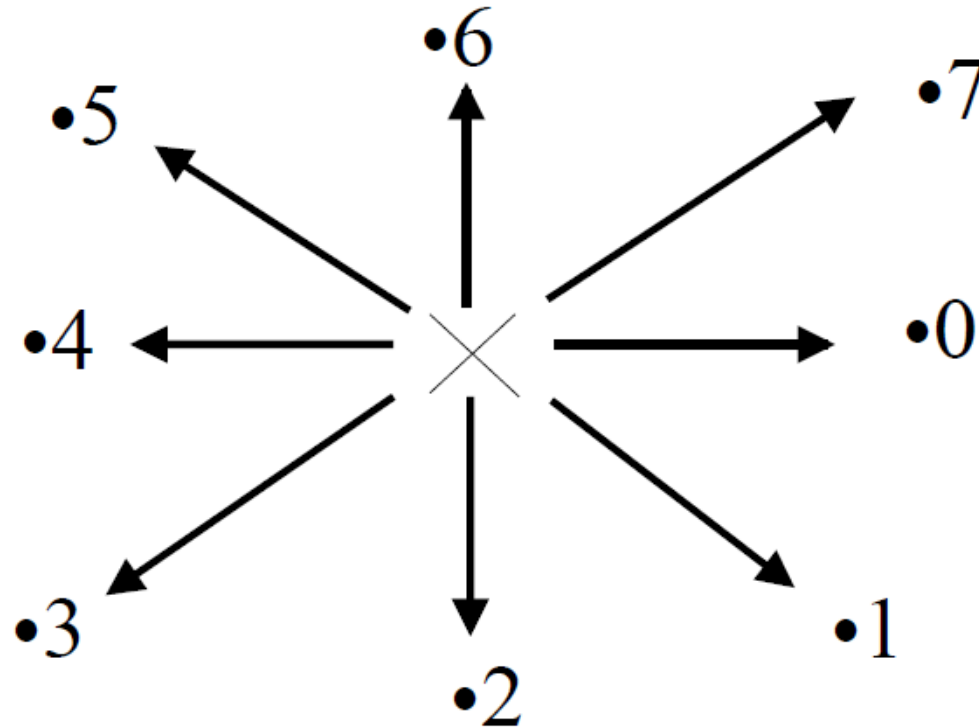
VIII – Introduction au Deep Learning

IV – Codage de contours

- **Code de Freeman**
 - Reconnaissance de formes
 - Calculs géométriques
- **Transformée de Hough**
 - Détections de formes géométriques: droites, cercles,...

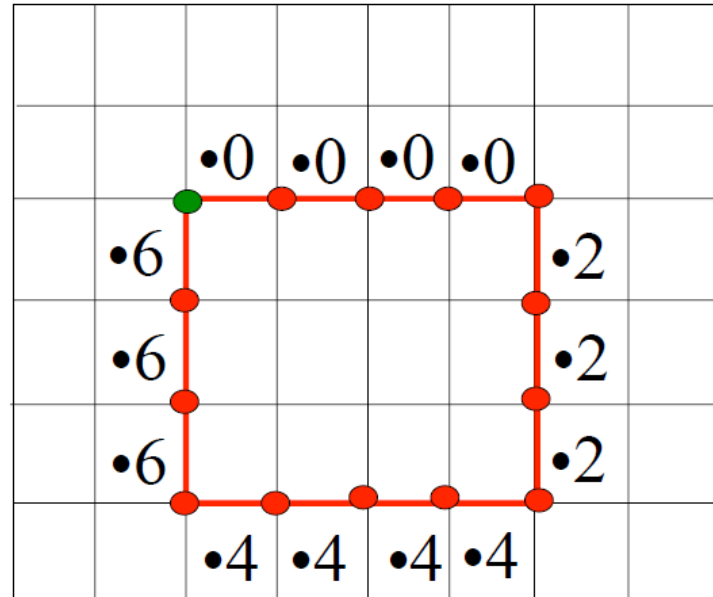
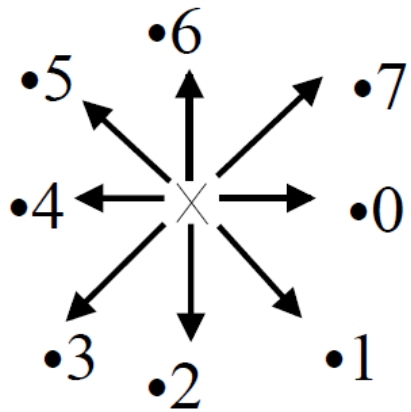
IV – Codage de contours

- Principe: Codage de l'orientation des pixels de contours



IV – Codage de contours

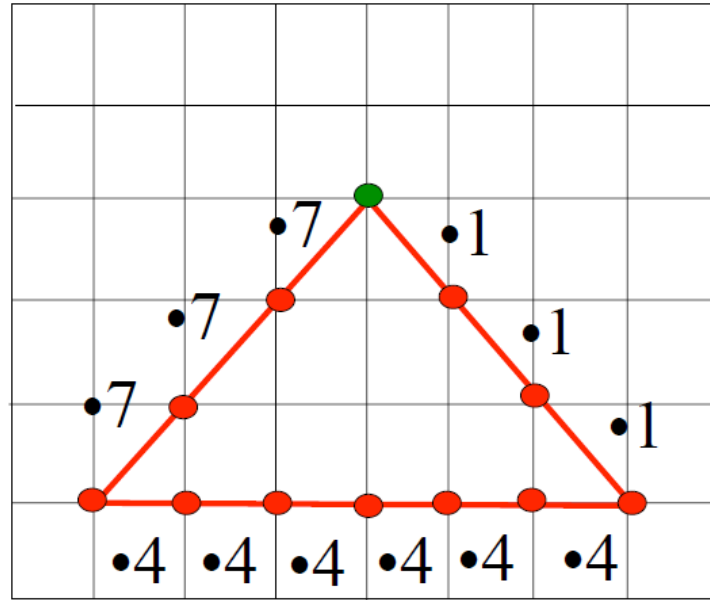
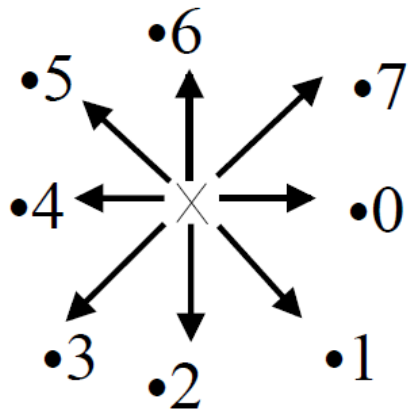
Codage contours d'un rectangle



Code obtenu: 00002224444666

IV – Codage de contours

Codage contours d'un triangle



Code obtenu: 111444444777

IV – Codage de contours

- Première étape: Compression du code

- Code original transformé en vecteurs (ou Segments).

- Chaque vecteur contient deux composantes: Orientation et Longueur.

- Seconde Etape: Reconnaissance formes et/ou Calculs géométriques

Analyse du code

- Example 1: Rectangle →

Code Obtenu: 00002224444666

Code Comprimé: 0:4 ; 2:3 ; 4:4 ; 6:3

4 vecteurs détectés : Quadrilatère

Périmètre= $4 + 3 + 4 + 3 = 14$

- Example 2: Triangle →

Code Obtenu: 111444444777

Code Comprimé: 1:3 ; 4:6 ; 7:3

3 vecteurs détectés : Triangle

Périmètre= $3 \times 1.4 + 6 + 3 \times 1.4 = 14.4$