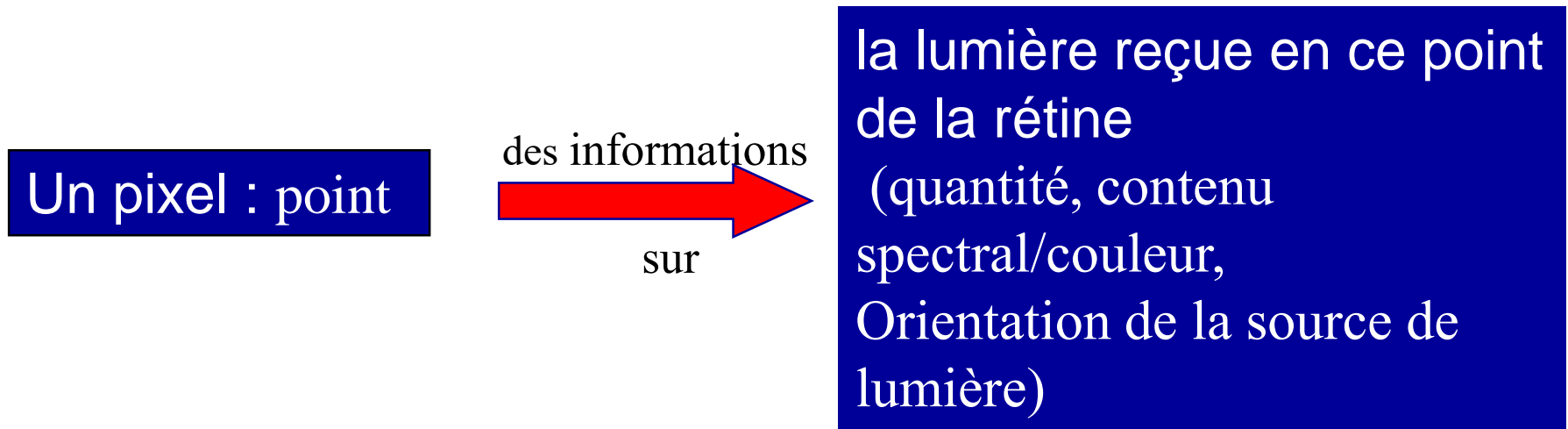
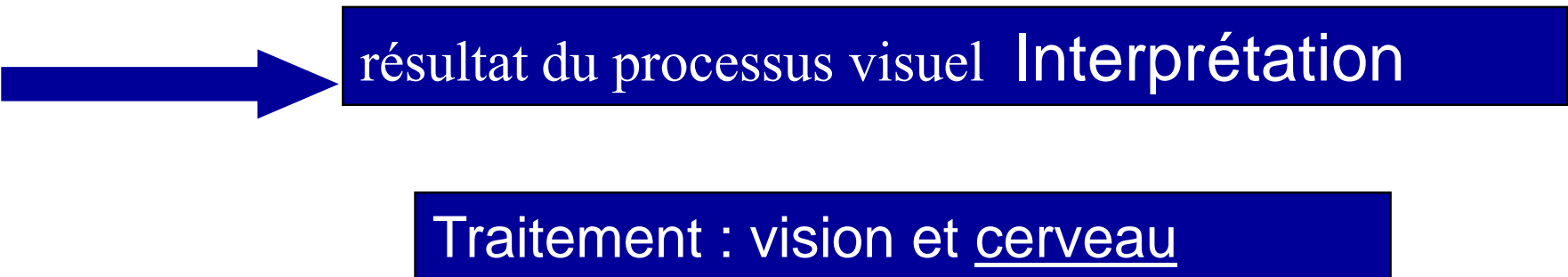


# Introduction

- Notre environnement spatial est composé d'objets structurés, dans un monde 3D
- L'humain sait décrire et interpréter cet environnement (presque parfaitement)
- Or l'information disponible sur la rétine n'est qu'un ensemble de points
- Pour un capteur on parle de pixels (picture elements)



- Les éléments « vus » (table, chaise, personnage, ...) n'existent pas sur la rétine !!



# Partie 1 : Rappels

- 1- Outils Fondamentaux
- 2- Introduction aux opérateurs de détection de contours

# 1- Outils Fondamentaux

# Quelques définitions

L'image peut être plongée dans le plan discret :  $\mathbb{Z}^2$

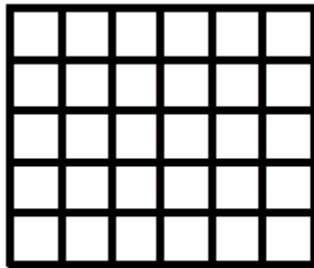
Point discret : point de  $\mathbb{Z}^2$ , correspond au pixel centré en ce point

2 points discrets P et Q sont dits **4-adjacents** ssi

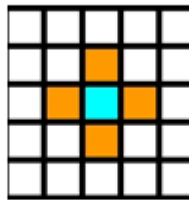
$$|x_P - x_Q| + |y_P - y_Q| = 1$$

2 points discrets P et Q sont dits **8-adjacents** ssi

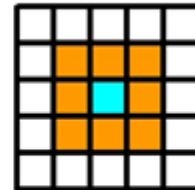
$$\text{Max}(|x_P - x_Q|, |y_P - y_Q|) = 1$$



Points discrets



4-connectiv  



8-connectiv  

## Distances Classiques - Rappel

L'image peut être plongée dans le plan discret :  $\mathbb{Z}^2$

Point discret : point de  $\mathbb{Z}^2$ , correspond au pixel centré en ce point

Distance euclidienne : P et Q

$$D_2(P, Q) = ((x_P - x_Q)^2 + (y_P - y_Q)^2)^{\frac{1}{2}}$$

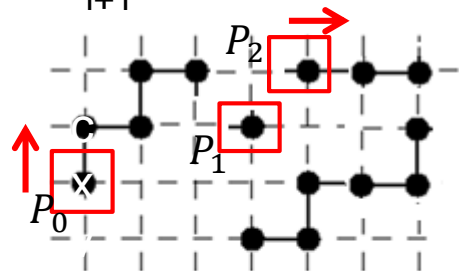
$$L_1(P, Q) = |x_P - x_Q| + |y_P - y_Q|$$

$$L_\infty(P, Q) = \text{Max}(|x_P - x_Q|, |y_P - y_Q|)$$

## Chemin discret et connexité

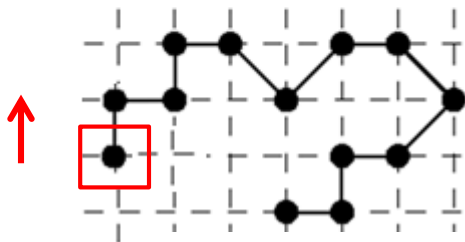
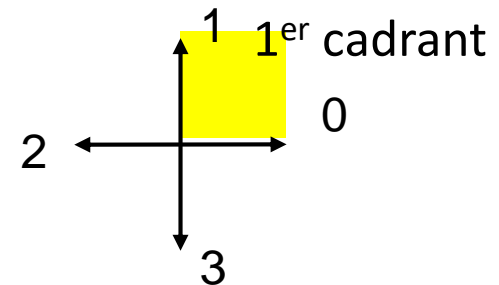
Un **chemin discret**  $k$ -connexe est une suite de point discrets  $(P_0, P_1, \dots, P_n)$  tels que  $\forall i \in \{1, \dots, n\}$ ,  $P_{i-1}$  et  $P_i$  sont  **$k$ -adjacents**.

**Code de Freeman** : le suite  $(P_0, P_1, \dots, P_n)$  est représentée par  $(P_0, d_0, \dots, d_{n-1})$ . La valeur  $d_i$  code le déplacement relatif de  $P_i$  à  $P_{i+1}$  avec les configurations :



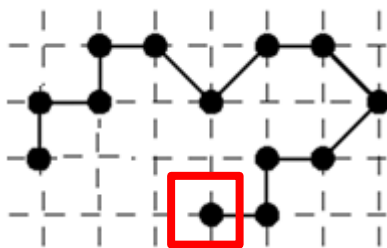
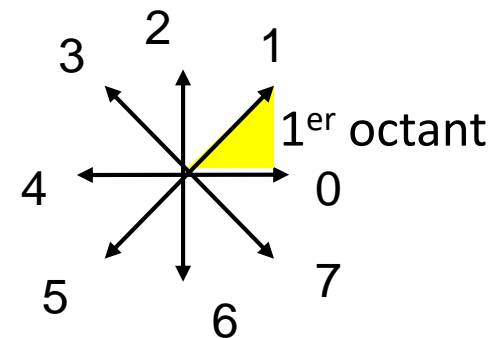
$d_1$  Codage du chemin en :

$\{P_0, 1010\}; \{P_1\}, \{P_2, 0033223\}$



$d_\infty$  Codage du chemin en :

$\{P_0, 2020710664464\}$



Codage du chemin en :

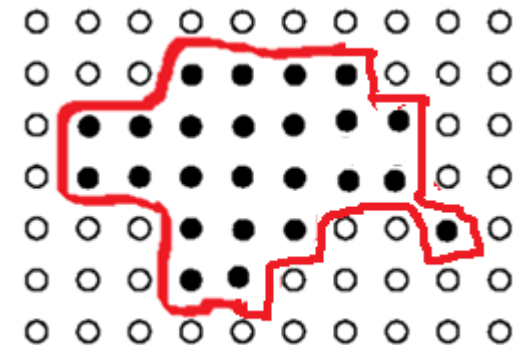
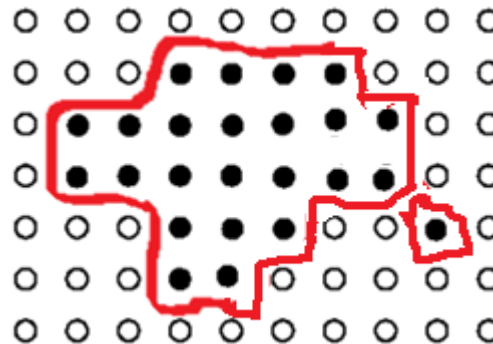
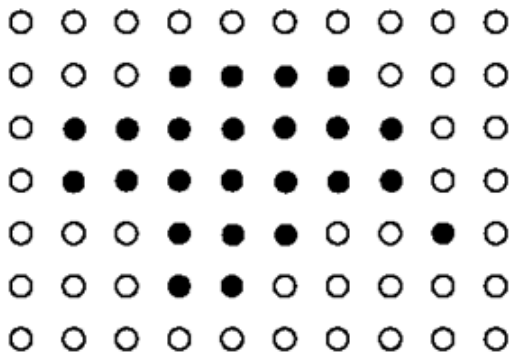
$\{P, 020134534646\}$

## Composante connexe / région 2D

**Ensemble k-connexe** : ensemble de points discrets  $E$  tels que

$\forall P, Q \in E$ , il existe un chemin discret k-connexe dans  $E$   
d'extrémités  $P$  et  $Q$

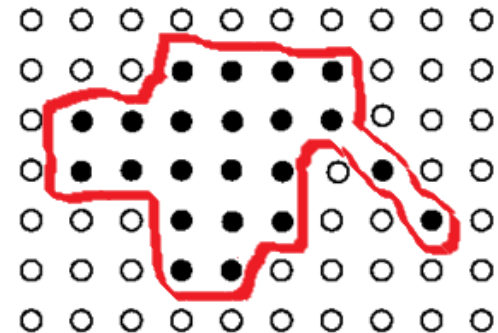
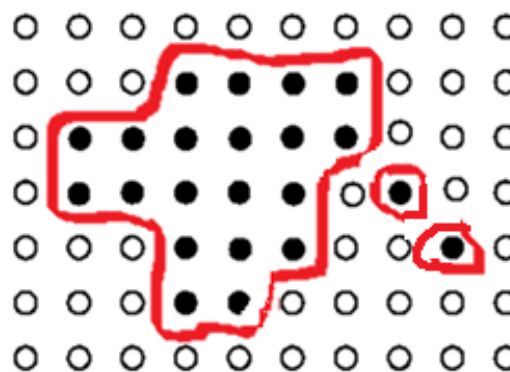
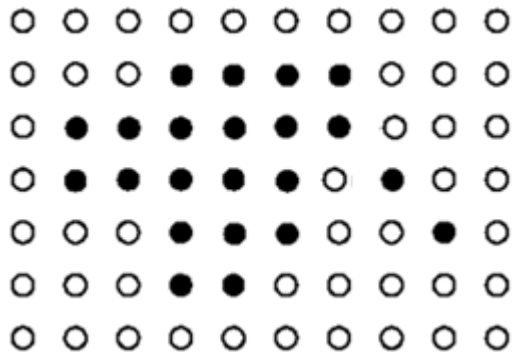
**Composante connexe** d'un ensemble de points discrets :  
ensemble connexe maximal (ou classe d'équivalence pour la  
relation d'adjacence)  $E$  tels que  $\forall P, Q \in E$ , il existe un chemin  
discret k-connexe dans  $E$  d'extrémités  $P$  et  $Q$





## Composante connexe / région 2D

**Exemple composante connexe** ensemble de points discrets :  
ensemble connexe maximal

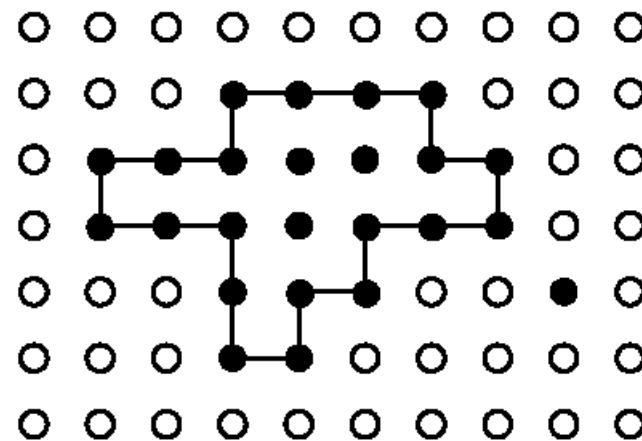
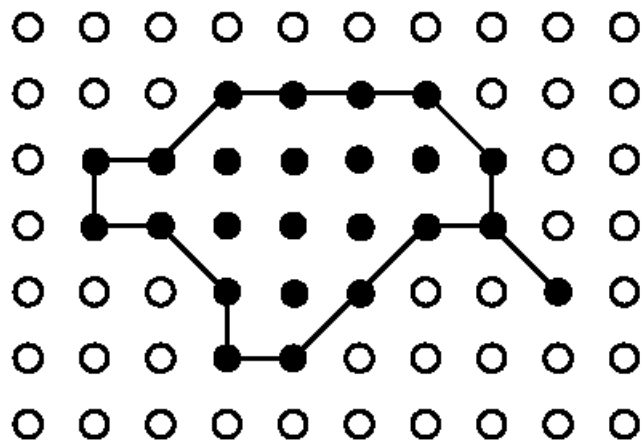


## Frontière d'une région 2D

### Définition classique

La frontière d'une région 8-connexes (respectivement 4-connexes)  $R$  est l'ensemble des points de  $R$  dont au moins un des 4-voisins (resp. 8-voisins) n'est pas élément de  $R$ .

⇒ La frontière est composée de chemins 8-connexes (resp 4-connexes).



## La convolution

Soit  $f$  une fonction donnée ;  $f : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$

La convolution de  $f$  par  $h$  est définie par :

$$(f * h)(x, y) = \iint_{\mathbb{R}^2} f(u, v) \cdot h(x - u, y - v) du dv$$

On pose

$$U = x - u \Leftrightarrow u = x - U ; du = -dU \quad \because \quad V = y - v \Leftrightarrow v = y - V ; dv = -dV$$

D'où on obtient

$$(f * h)(x, y) = \iint_{\mathbb{R}^2} h(x - U, y - V) \cdot f(U, V) dU dV = (h * f)(x, y)$$

$$f * h(x, y) = h * f(x, y)$$

## La convolution

Soit  $f$  une fonction définie  $R \times R \rightarrow R$   $f * h(x, y) = h * f(x, y)$   
 La convolution de  $f$  par une fonction  $h$  (filtre  $h$ ) par :

$$f * h(x, y) = (f * h)(x, y) = \iint_{R^2} f(u, v) \cdot h(x - u, y - v) du dv$$

Si on pose :  $U = x - u, V = y - v$  On a :  $dU = -du, dV = -dv$

$$(f * h)(x, y) = \iint_{R^2} f(x - U, y - V) \cdot h(U, V) dU dV = (h * f)(x, y)$$

$$\text{D'où } (f * h)(x, y) = (h * f)(x, y)$$

## Quelques propriétés de la convolution

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

Commutativité :  $f * h = h * f$

Distribut de  $+$  /  $*$  :  $h * (f + g) = h * f + h * g$

Dérivabilité :  $(f * h)' = f * h'$

Quelques propriétés :

associativité :  $f * h * g = (f * h) * g$

commutativité :  $f * h = h * f$

dist. de  $+$  /  $*$  :

dérivabilité :  $(f * h)' = f * h'$

## Dans le cas discret la convolution se traduit par

Soit  $I$  le support d'une image et le filtre  $h$  défini par

$$h : [m_1, m_2] \times [n_1, n_2] \rightarrow R$$

la convolution de  $I$  par  $h$  est définie par :

$$(I * h)(x, y) = \sum_{u=m_1}^{u=m_2} \sum_{v=n_1}^{v=n_2} h(u, v) f(x - u, y - v)$$

$h$  est appelé noyau de la convolution (on parle aussi de filtre, de masque)

$[m_1, m_2] \times [n_1, n_2]$  représente le support de la fonction  $h$  : l'ensemble en dehors duquel la fonction est identiquement nulle

## 2: Rappel sur les opérateurs différentiels pour la détection de contours

## Segmentation à base de frontières

### Détection de contours : Edge detection

But :

Présentation des opérateurs de base pour la détection de contour

Avant propos : Nous avons déjà visité des méthodes de segmentation basées région où on s'attache à partitionner l'image en plages de pixels semblables

→ on s'appuie sur des descripteurs et des critères de similarité.

Dans ce chapitre on s'intéresse aux méthodes qui identifient un objet par ses contours : on s'appuie sur des propriétés de différenciation ou de changements de la « valeur image »



## Détection de contours : Edge detection

\* **Outils** : opérateurs de convolution voir chapitre : outils fondamentaux

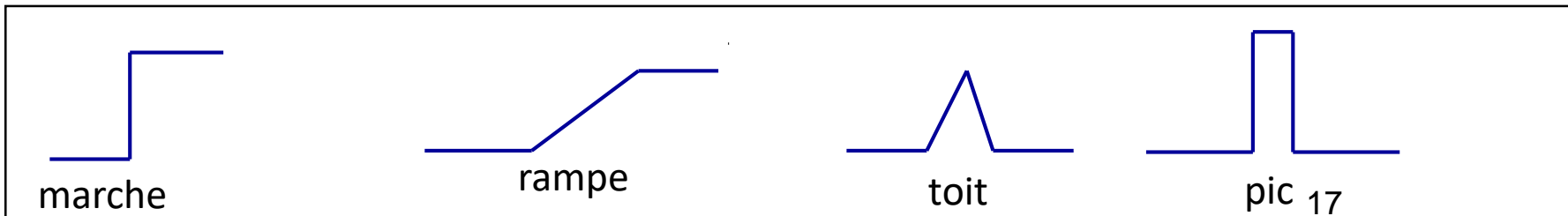
- **Qu'est-ce qu'un contour ?**

- zone de forte transition de l'intensité lumineuse
- orientée
- point de variation maximum ?

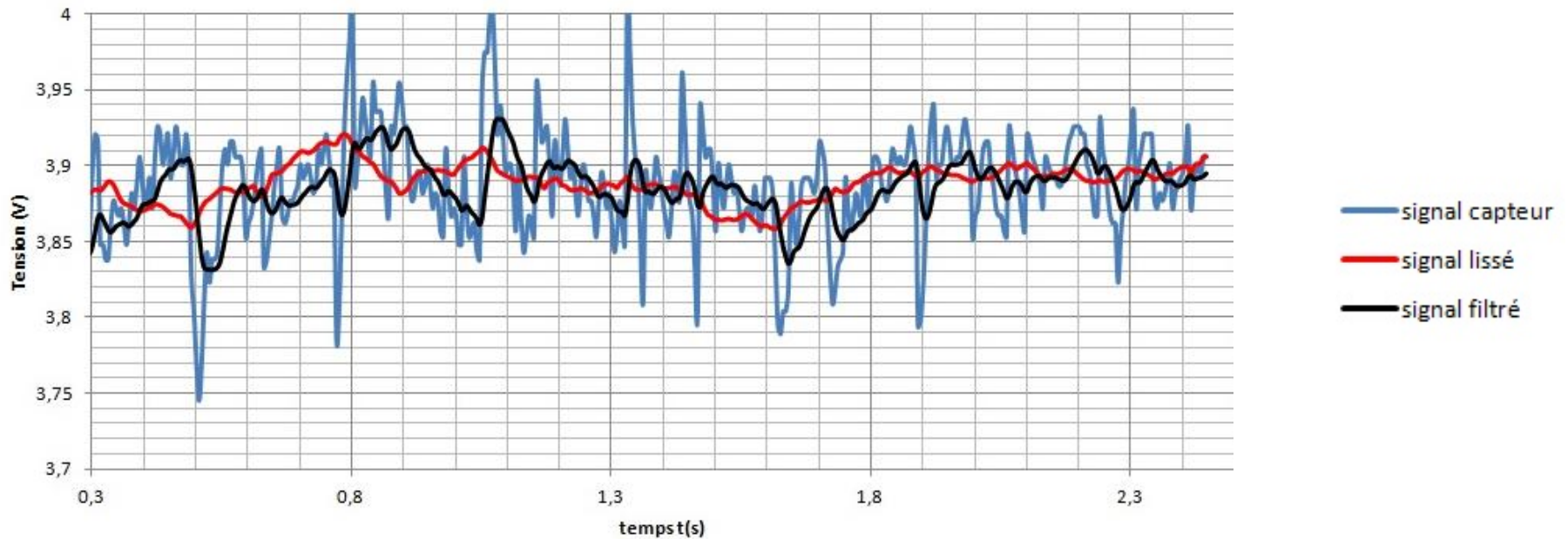
- **Principe de base**

- Un contour se matérialise par une rupture d'intensité dans l'image suivant une direction donnée.

Quelques profils de contours



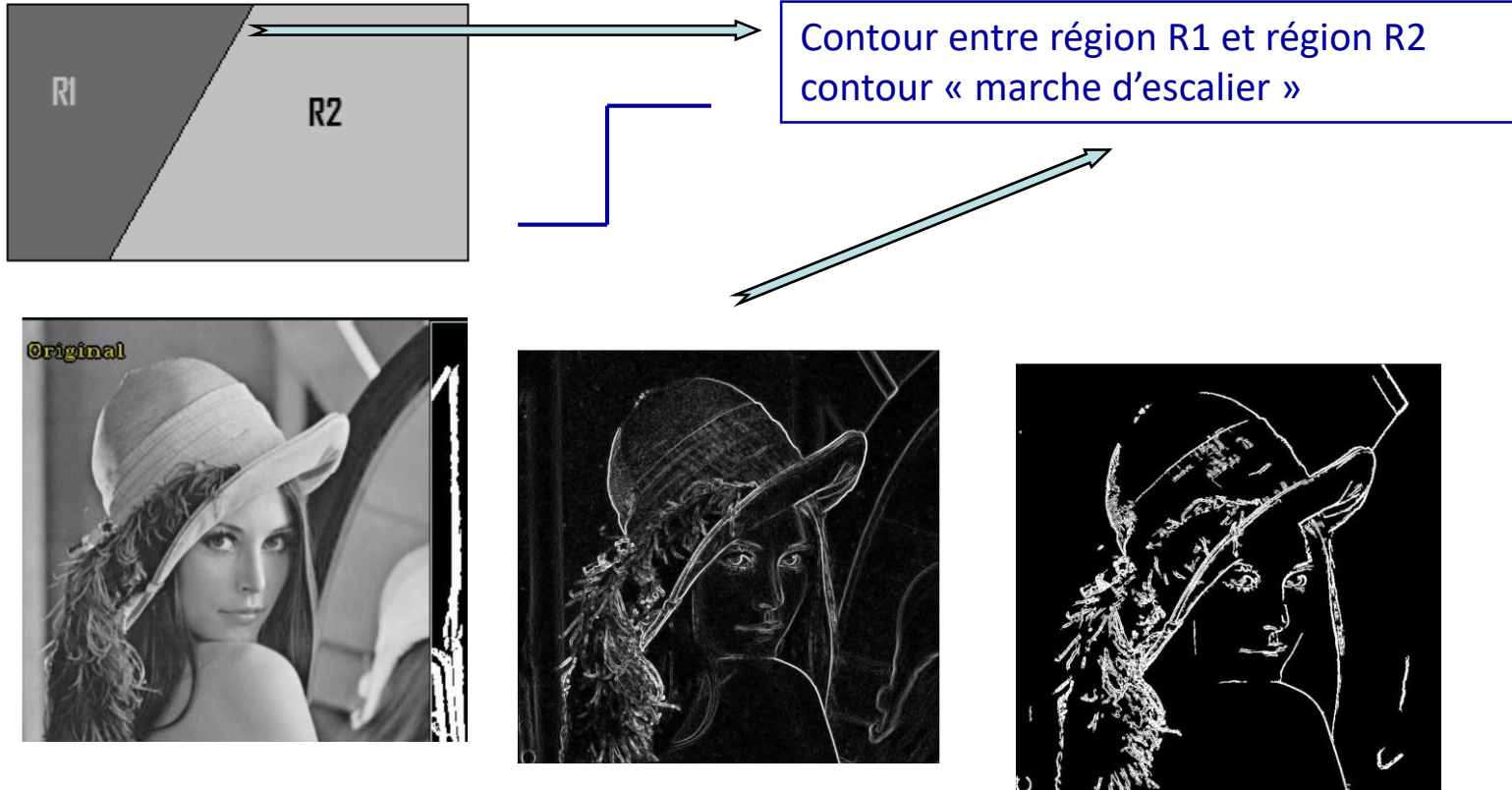
## Qu'est qu'un contour



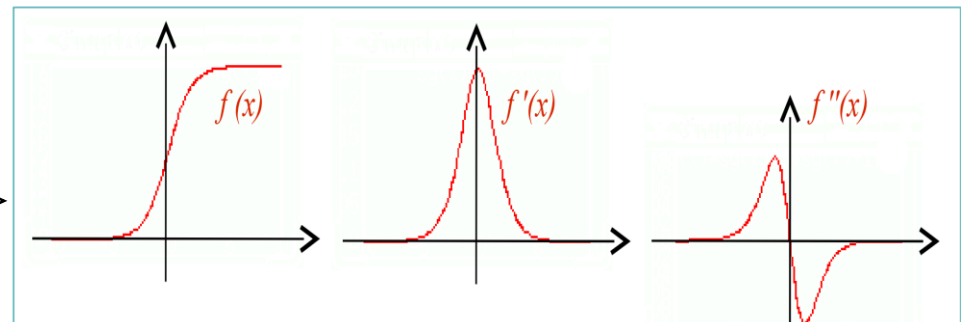
Contour théorique en 2D

Contour: changement brutal de l'intensité lumineuse.

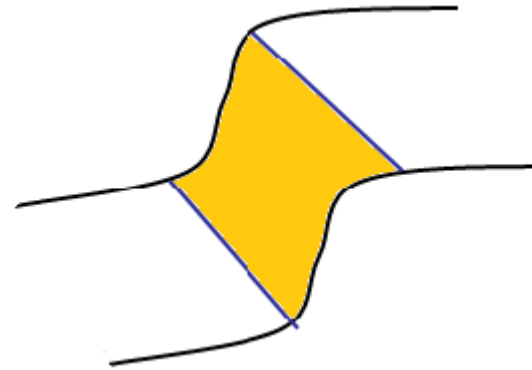
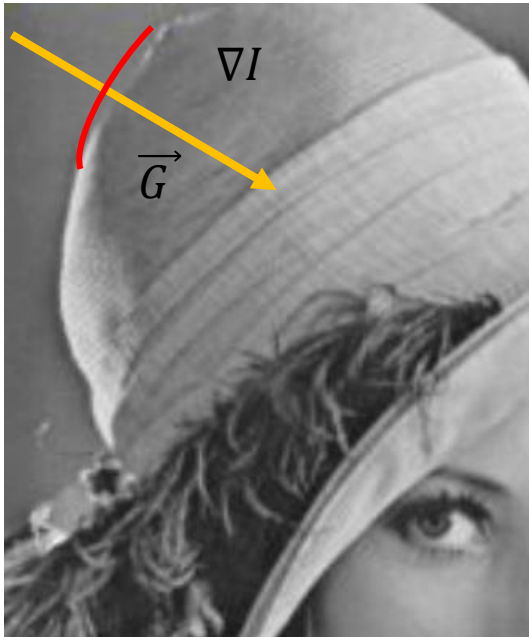
## Détection de contours : Edge detection



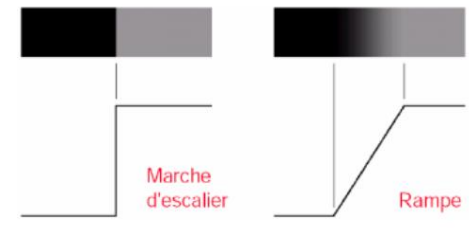
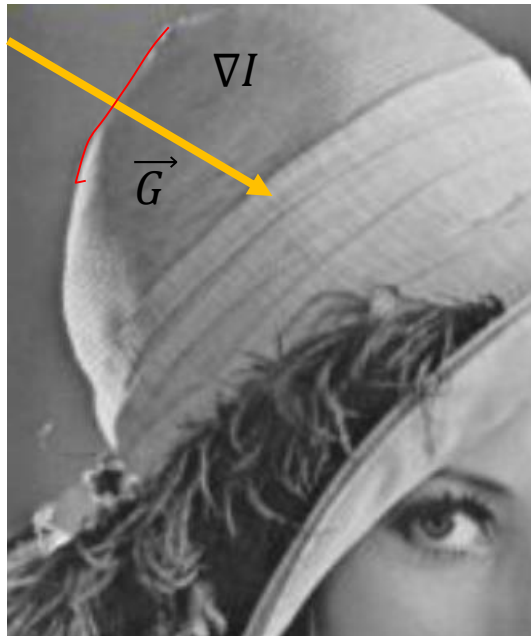
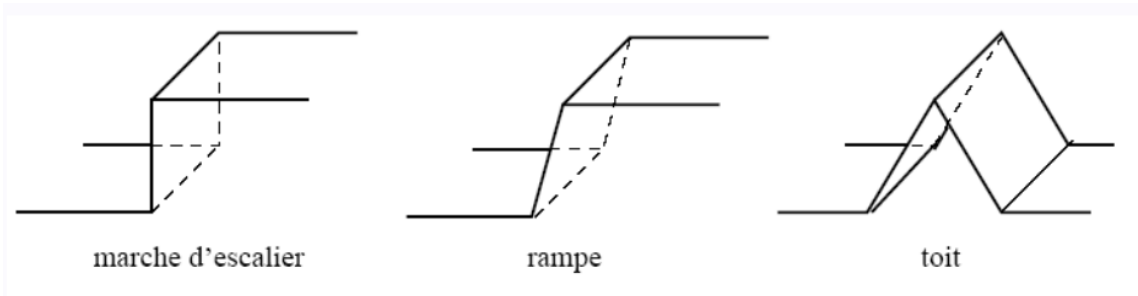
2 principes de détection de contours :  
maximum dérivée première  
Passage par zéro dérivée seconde



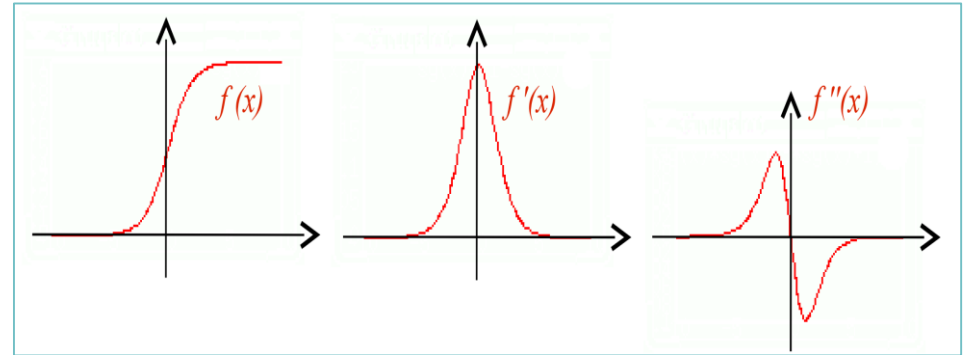
Pour retrouver cette propriété en 2d, il faut se placer dans la direction du gradient.



Pour la dérivée seconde, on passe par le laplacien



## En 1D



En se plaçant dans le domaine du continu:

On représente l'image  $I$  par la fonction  $f$ ,

Pixel  $(x, y) \rightarrow$  on a  $f(x, y)$

Si on suppose que  $f$  est une fonction continue, pour quantifier les variations de  $f$  en calculant sa différentielle :  $df(x, y)$

$$df(x, y) = \frac{\partial f(x, y)}{\partial x} dx + \frac{\partial f(x, y)}{\partial y} dy$$

En prenant la fonction  $f$ , soit le point  $((x, y), f(x, y))$

On considère le vecteur :  $\nabla. f(x, y) = \begin{cases} \frac{\partial f(x, y)}{\partial x} \\ \frac{\partial f(x, y)}{\partial y} \end{cases}$

$\nabla. f(x, y)$  Vecteur gradient

Le gradient ( $\nabla$ .) est un opérateur **directionnel**  $\rightarrow$  vecteur

L'**amplitude** du gradient de l'image en 1 pixel donne une indication sur l'amplitude de la variation de la valeur image (intensité lumineuse) en ce point.

$$\|\nabla.f(x,y)\| = \sqrt{\left(\frac{\partial f(x,y)}{\partial x}\right)^2 + \left(\frac{\partial f(x,y)}{\partial y}\right)^2}$$

**exemple : calcul de l'amplitude par la norme euclidienne**

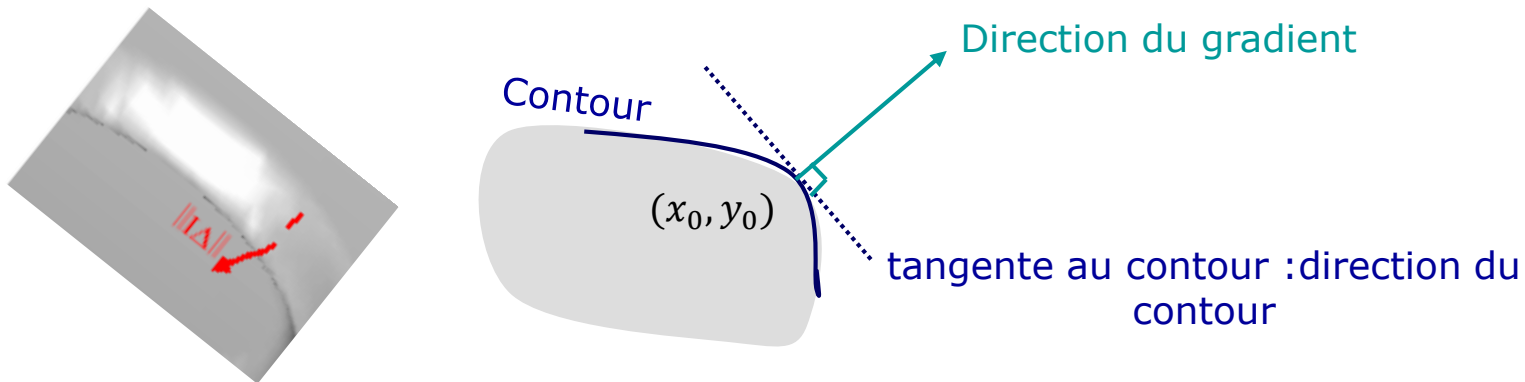
Sa direction est donnée par :

$$\theta = \text{Arc tan} \left( \frac{\frac{\partial f(x,y)}{\partial y}}{\frac{\partial f(x,y)}{\partial x}} \right)$$



Au niveau de l'interprétation, le gradient donne des courbes de niveaux ou des courbes d'altitude.

- La composante gradient  $g$  correspond à la direction principale de variation.
- La direction du vecteur gradient est par définition normale au contour.
- La direction du contour (la tangente au contour) correspond à la direction de l'isophote ou ligne de niveau.



Dans le domaine discret (image), le calcul des dérivées partielles font l'objet d'une approximation par des différences finies.

$$\frac{\partial f^+(x, y)}{\partial x} \quad \text{est évaluée par} \quad \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x} = \Delta_x^+ f(x, y)$$

$$\frac{\partial f^+(x, y)}{\partial y} \quad \text{est évaluée par} \quad \frac{f(x, y + \Delta y) - f(x, y)}{\Delta y} = \Delta_y^+ f(x, y)$$

En passant à la dérivée au point  $(x, y)$  on obtient :

$$\begin{aligned} \frac{\partial f(x, y)}{\partial x} &\approx \frac{\Delta f(x, y)}{2 \Delta x} = \frac{\Delta_x^+ f(x, y) + \Delta_x^- f(x, y)}{2} \quad (\text{moyenne des 2 valeurs}) \\ &= \frac{f(x+1, y) - f(x-1, y)}{2} \end{aligned}$$

Dans le domaine discret (image), le calcul des dérivées partielles font l'objet d'une approximation par des différences finies.

En passant à la dérivée par rapport  $x$  au point  $(x, y)$  on obtient :

$$\frac{\partial f(x, y)}{\partial x} \approx \frac{\Delta f(x, y)}{2 \Delta x} = \frac{\Delta_x^+ f(x, y) + \Delta_x^- f(x, y)}{2}$$

$$\frac{\partial f(x, y)}{\partial x} \approx \frac{\Delta f(x, y)}{2 \Delta x} = \frac{f(x + 1, y) - f(x - 1, y)}{2} ; \Delta x = 1$$

Et pour la dérivée par rapport  $y$  à au point  $(x, y)$  on obtient :

$$\frac{\partial f(x, y)}{\partial y} \approx \frac{\Delta f(x, y)}{2 \Delta y} = \frac{f(x, y + 1) - f(x, y - 1)}{2} ; \Delta y = 1$$

Dans le domaine discret (image), le calcul des dérivées partielles peuvent être vues comme la convolution de  $f$  par une fonction discrète à support fini.

Ceci peut être écrit sous la forme :

$$\frac{\partial f(x, y)}{\partial x} \approx \frac{\Delta_x^+ f(x, y) + \Delta_x^- f(x, y)}{2}$$

$$\left. \frac{\partial f(x, y)}{\partial x} \right|_{(i,j)} \approx \frac{1}{2} f * K(i, j) \quad \text{avec} \quad K_x = \frac{1}{2} [\dots 0 \quad -1 \quad 0 \quad 1 \quad 0 \dots]$$

La dérivation peut être considérée comme un filtre dont le support est fini. Un filtre « passe haut » car il favorise les hautes fréquences spatiales, comme les détails. Un filtre « passe haut » est caractérisé par un noyau comportant des valeurs négatives autour du pixel central

En supposant que l'opérateur est séparable, en procédant de la même manière on peut écrire :

$$\frac{\partial f(x, y)}{\partial y} \approx \frac{\Delta_y^+ f(x, y) + \Delta_y^- f(x, y)}{2}$$

$$\left. \frac{\partial f(x, y)}{\partial y} \right|_{(i, j)} \approx f * K_y(i, j)$$

avec  $K_y = \frac{1}{2} [\dots 0 \quad -1 \quad 0 \quad 1 \quad 0 \dots]^t$

- Filtrage suivi de dérivation

Soit  $h$  un filtre donné, on rappelle la propriété :  $(f * h)' = f * h'$

Ainsi on peut appliquer le gradient (la dérivé) d'un filtre pour atténuer le bruit et calculer le gradient de l'image.

En supposant que l'opérateur est séparable, en procédant de la même manière on peut écrire :

- Filtrage suivi de dérivation  $\sim$  dérivation suivie de filtrage

Soit  $h$  un filtre donné, on rappelle la propriété :  $(f * h)' = f * h' = h' * f$

$$\frac{1}{3} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} = \frac{1}{3} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \text{Filtre Moyenneur}$$
  

$$\frac{1}{4} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \quad \sim \text{Filtre gaussien}$$

Opérateur *Kirsh*

$$\begin{array}{cc} K_1 & K_2 \\ \frac{1}{15} \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} & \frac{1}{15} \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} \\ K_3 & K_4 \\ \frac{1}{15} \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} & \frac{1}{15} \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix} \end{array}$$

## Récapitulatif : application à la détection de contours

Pour une image en intensité lumineuse au point  $P(x, y) \rightarrow I(x, y)$

**Gradient** :  $\nabla I(x, y) = (G_x, G_y) = \left( \frac{\partial I(x, y)}{\partial x}, \frac{\partial I(x, y)}{\partial y} \right)$

**Amplitude/module** gradient:  $\nabla I(x, y) = \|G(x, y)\| = \sqrt{G_x^2 + G_y^2}$

Pente/direction du gradient :  $\theta(x, y)$

$$\theta(x, y) = \text{Arctan} \left( \frac{\frac{\partial I(x, y)}{\partial y}}{\frac{\partial I(x, y)}{\partial x}} \right) = \text{Arctan} \left( \frac{G_y}{G_x} \right)$$

Exemple de calcul : filtre différentiel discret

$y \uparrow \rightarrow x$	-1	0	1
	-1	0	1
	-1	0	1

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

## Récapitulatif :

- Un détecteur de contour est un opérateur différentiel
- Peut être réalisée par une convolution (discret :masque) :  $H(i, j)$

Le masque H vérifie :

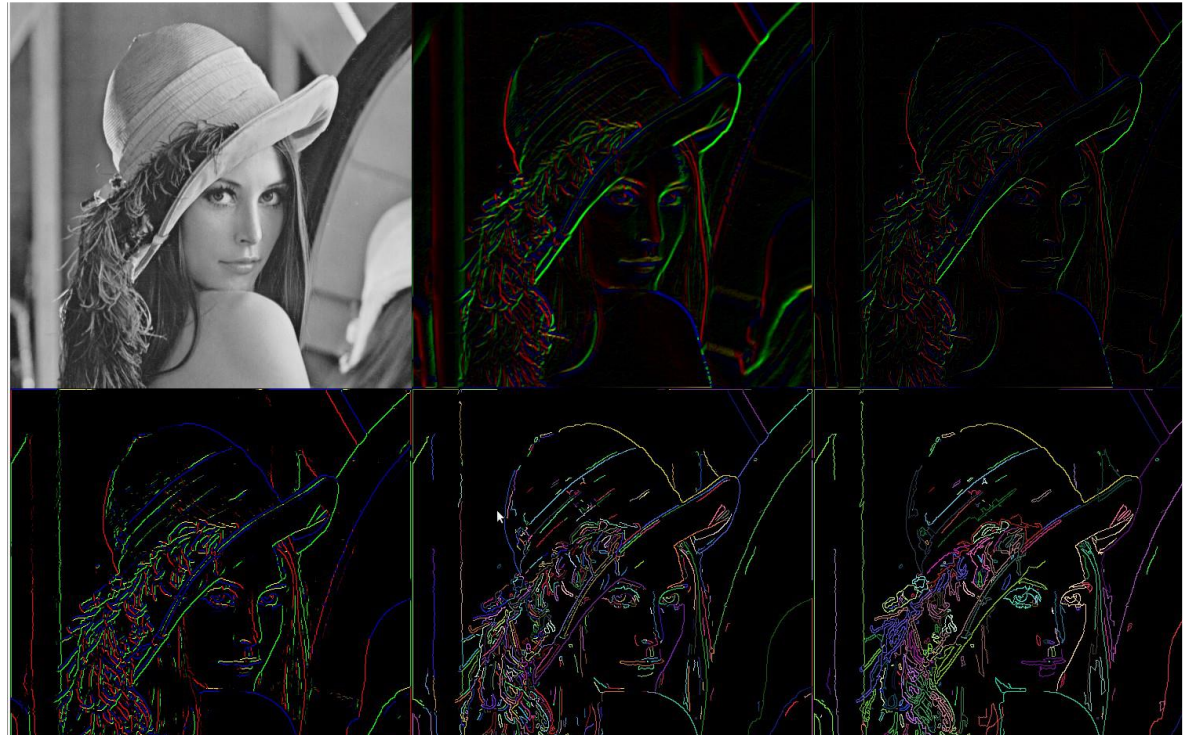
- L'opérateur H est antisymétrique (fonction impaire)
  - H est antisymétrique (impaire)
    - $\sum_i \sum_j H(i, j) = 0$
  - $\sum_i \sum_j H(i, j)_+ = 1 \quad \sum_i \sum_j |H(i, j)|_- = 1$ 
    - Normalisation des coefs

La somme des coefficients positifs = la somme des valeurs absolue  
les coefficients négatifs =1





Visualisation des contour : image binaire :  
contour/ non-contour



Résultat d'une détection de contour : opérateur du Kirsh.  
Les contour sont visualisés selon leur direction (1 couleur par direction)

## Post-traitement

Problématique : 2 points

1. Opérateurs différentiels : sensibles au bruit les variations d'intensité dues au bruit peuvent être des points de fort gradient



- besoin de pré- prétraitements : filtrage bruit
- besoin de post-traitements

2. Résultat ensemble de point sans structure  
structurer le résultat (!) → si besoin

Résultat sans post-traitement : affichage des contours (valeur du saut normalisée)



## Post-traitement

### 1. Seuillage ou valeurs de tolérance

But : sélectionner les valeurs significatives

### 2. Seuillage par hystérésis des maximums locaux

But : limiter la fragmentation des contours obtenus

**Seuillage par hystérésis** : pour éviter un seuillage unique brutal

--> 2 seuils :  $S_h > S_b$

On suit les lignes de crêtes

- les maximums locaux de valeur supérieure à  $S_h$
- les maximums locaux de valeur supérieure à  $S_b$   
appartenant à une composante connexe de maxima locaux  
contenant au moins une valeur  $S_h$

Résultat : une image binaire (image des contours)



## Résultats

Valeur du seuillage à 0.5,



0.2,

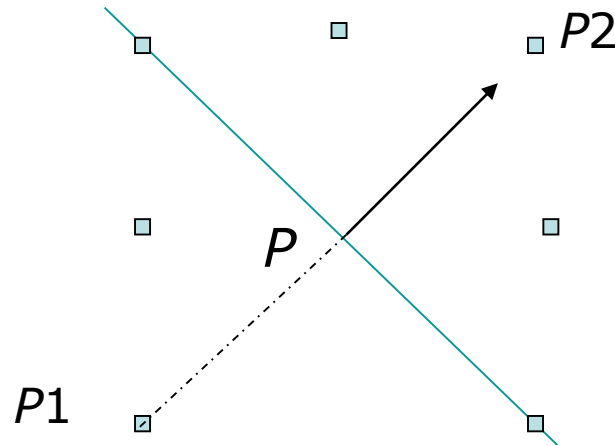


0.7



## Affinage des contours

- Extraction des maxima locaux de la norme du gradient.
- On se positionne dans une géométrie à 8-connexité



$Gr$  : norme du gradient en  $P$

$Gr1$  : norme du gradient en  $P1$

$Gr2$  : norme du gradient en  $P2$

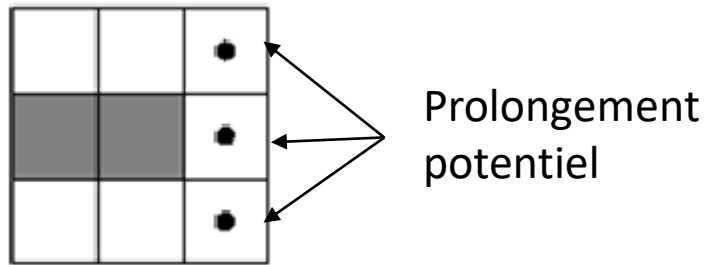
Présence d'un maximum local :  $Gr \geq Gr1$  et  $Gr \geq Gr2$

## Résultat : Image des contours

### Fermeture des contours

- Idée : suivre une ligne de crête dans l'image de la norme du gradient à partir de chaque extrémité de contour (et la compléter)

**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.

**Résultats seuillage**

Seuillage bas / haut : 0.4 / 0.6,



0.25/0.75



0.25/0.6



Contours affinés

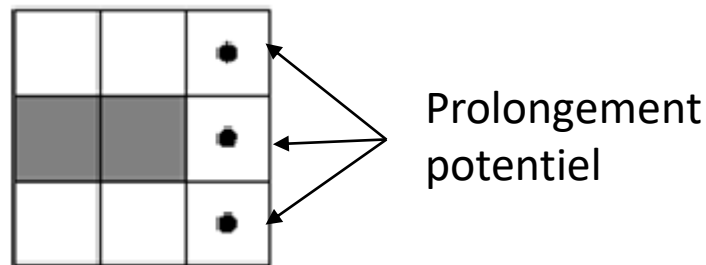




## Fermeture des contours

- Idée : suivre une ligne de crête dans l'image de la norme du gradient à partir de chaque extrémité de contour.

**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.

Annexe : pour compléter les connaissances

## Détection de contours : Modèle de Canny-Deriche

- Modélisation du problème
  - Recherche d'un filtre de convolution  $f$  qui optimise les critères :
    - critère de détection --> robustesse au bruit ,
    - Critère de précision --> bonne localisation ,
    - Unicité de la réponse --> non duplicité des contours

## Détection de contours : Modèle de Canny-Deriche

- Modèle :

- Un contour est un seuil d'amplitude  $a$  auquel on ajoute un bruit gaussien  $n$  de moyenne  $E = 0$

et de variance  $\sigma^2$ , ce qui donne :

$$\rightarrow I(x) = aU(x) + n(x)$$

où  $U(x)$  est la fonction de heaviside unité :

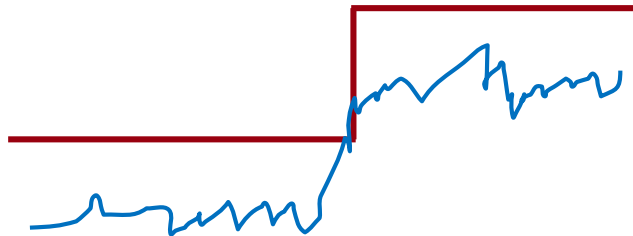
$$U(x) = 0 \text{ si } x < 0$$

$$U(x) = 1 \text{ si } x \geq 0$$

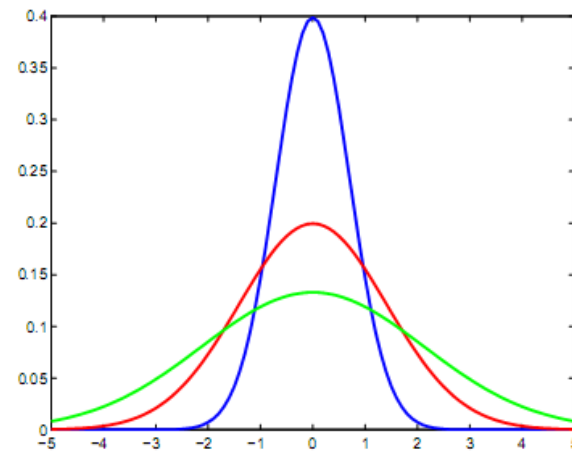
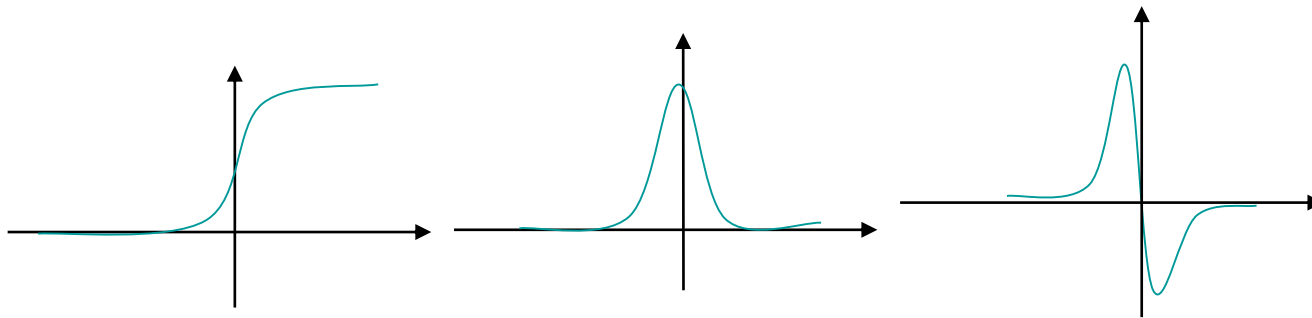
Le contour effectif est situé en  $x = 0$

- Modèle préconisé par Canny :

⇒ Modèle : contour en escalier (saut quantitatif)



Contour idéal  
Contour réel



⇒ critère de détection : robustesse au bruit

$f$  antisymétrique  $\Rightarrow f(t) = -f(-t)$

$$I_1(x_0) = \left( \int_{-\infty}^{+\infty} I(t) \cdot f(-t) dt \right) = \left( \int_{-\infty}^{+\infty} (aU(t) + n(t)) \cdot f(-t) dt \right)$$

$$O_s(0) = a \int_0^{\infty} f(-t) dt = \left( -a \int_0^{\infty} f(t) dt \right) = \left( a \int_{-\infty}^0 f(t) dt \right)$$

Rép. bruit :

$$O_n(0) = \left( \int_{-\infty}^{+\infty} f(-t) n(t) dt \right)$$

⇒ Robustesse au bruit

Le bruit est caractérisé par son écart type



Robustesse  $\equiv$  réponse signal / écart-type bruit

Critère de détection  
*ratio (signal/bruit)*

⇒ hypothèse nature du bruit

Bruit blanc stationnaire  
 $E(n(x)) = 0$  ; *variance*  $= \sigma^2(n)$

$$\begin{aligned}
 C_1 &= \frac{\text{Signal}}{\text{Bruit}} \\
 &= \frac{a}{\sigma} \frac{\int_{-\infty}^0 f(t) dt}{\sqrt{\int_{-\infty}^{\infty} f^2(t) dt}} \\
 \text{or } \frac{a}{\sigma} &\text{ est constante}
 \end{aligned}$$

Le filtre recherché doit maximiser ce rapport



$$C_1 = \frac{\int_{-\infty}^0 f(t) dt}{(\int_{-\infty}^{\infty} f^2(t) dt)^{\frac{1}{2}}}$$



Critère de localisation :

Idée : limiter lissage de la zone de transition par l'opération du lissage

Le point contour doit être aussi près que possible de la position théorique

Pour l'étude on se positionne en  $(0,0)$  ; le point doit rester proche du point  $(0,0)$

Possibilité : optimiser l'écart type de la position des différents points détectés contour dans cette zone



Maximiser l'inverse de l'espérance de la distance entre le point théorique et le point estimé.

Critère de localisation :

$$C = \frac{a}{\sigma} \frac{|f'(0)|}{\left(\int_{-\infty}^{\infty} f'^2(x) dx\right)^{\frac{1}{2}}}$$

$$C_2 = \frac{|f'(0)|}{\left(\int_{-\infty}^{\infty} f'^2(x) dx\right)^{\frac{1}{2}}}$$

Ce critère peut être retrouvé en étudiant

En étudiant le « déplacement » du maximum  
de la réponse en se plaçant en  $x_0 = 0$

$$\text{Max}(I(x) \rightarrow I'(X)$$

$$I'(x) = \frac{d}{dx} \int_{-\infty}^{+\infty} f(x)I(x-t)dt = 0$$

$$I'(x) = \int_{-\infty}^{+\infty} f'(x)I(x-t)dt = 0$$

Deux composantes :  $\begin{cases} \text{signal} \\ \text{bruit} \end{cases}$

- Critère d'unicité

Le filtre doit fournir une réponse unique à un contour

Ce qui peut se traduire par : si  $(x_{\max})$  la réponse du filtre  $f$  au bruit alors :

$f$  doit être tel que :  $(x_{\max})$  est proportionnelle à la taille (ou une fraction de la taille) du support du filtre  $\rightarrow \mathbf{(x_{\max}) = kW}$  (support de  $f = [-W, W]$  )

Bruit blanc stationnaire donc

$$x \left( \frac{\int_{-\infty}^{\infty} f'^2(x) dx}{\int_{-\infty}^{\infty} f''^2(x) dx} \right)^{\frac{1}{2}} \text{Rice1944}_{\max}$$

- Filtre de Canny

En combinant les 3 critères, Canny propose un filtre  $f$  qui maximise le produit :

$$\Omega = C_1 C_2$$

Sous la contrainte de l'unicité :  $(x_{\max}) = kW$

Problème variationnel -> Euler-Lagrange

Qui se traduit par une équation différentielle dont la solution est

$$f(x) = a_1 e^{\alpha x} \sin \omega x + a_2 e^{\alpha x} \cos \omega x \\ + a_3 e^{-\alpha x} \sin \omega x + a_4 e^{-\alpha x} \cos \omega x + c$$

Avec les conditions aux limites :

$$f(0) = 0 \quad ; \quad f(W) = 0 \quad ; \quad f'(0) = s \quad ; \quad f'(W) = 0$$

- Filtre de Rachid Deriche

Généralisation du filtre de Canny,

$f$  est de support infini ( $w \rightarrow \infty$ )

On obtient :  $a_1 = a_2 = a_3 = a_4 = c = 0$

Problème variationnel  $\rightarrow$  Euler-Lagrange

Qui se traduit par une équation différentielle

$$f(x) = s e^{a|x|} \sin \omega x$$

Avec  $a$  qui définit la longueur du filtre : compromis (détection-localisation)

# Chapitre : Transformée de Hough

Recherche de formes représentées par un modèle  
paramétrique

## Problématique

- Reconnaissance de formes particulières
  - forme représentée par une fonction paramétrique

**Exemple type :** Etant donnée une forme donnée par une représentation paramétrique; comment détecter les différentes instance de cette représentation présente dans l'espace d'étude. Exemple : un segment de droite, un cercle, une ellipse, ...

- Par corrélation ??? Bcp de masques... taille des masques : calculs fastidieux 😞



# Transformée de Hough

- Transformée développée par Paul Hough en 1962  
et Breveté par IBM
- Son utilisation
  - un outils populaire pour l'analyse d'image
  - Utilisé pour la détection de forme paramétrique:
    - de droites :  $ax+by+c=0$  → 3 paramètres liés → 2 paramètres
    - de cercles  $(x-a)^2+(y-b)^2 = R^2$  → 3 paramètres
    - Facettes planes :  $ax+by+cz+d=0$  → 4 paramètres
    - Surface degré 2 : → 6 paramètres ....
- Repris de l'importance avec les architectures neuronales

## Principe

- Représentation paramétrique de la forme : suppose la connaissance a priori
- Recherche dans un espace de paramètres

$$I \rightarrow R^n$$

$$(x, y) \mapsto (a_1, \dots, a_n)^t$$

$I$  : ensemble des pixels

pixel  $\rightarrow$  vecteur de paramètres

- Principe : accumulation d'évidences (ou d'hypothèse)
  - L'accumulateur de Hough est un mécanisme d'accumulation d'avis dans l'espace de paramètres

## Mise en œuvre

- Mise en œuvre : processus en 2 étapes

- Etape 1 : modélisation de  $E \subset R^n$

A chaque point  $(x,y)$  de  $E$ , on fait correspondre un vecteur de paramètre  $(a_1, \dots, a_n)^T$  avec un indice de confiance

- Etape 2 : estimation de l'indice de confiance

Dans l'espace  $R^n$ , pour chaque vecteur  $(a_1, \dots, a_n)^T$ ,  
calculer le nbre de point de  $E$ , ayant « choisi »  $(a_1, \dots, a_n)^T$   
Accumulation de votes : gérer une matrice de confiance

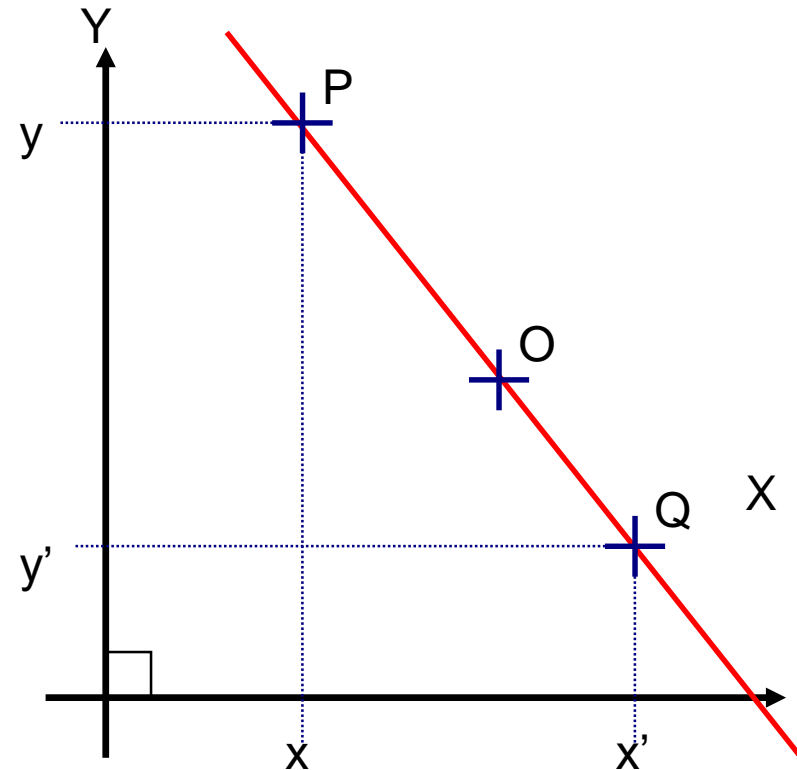
Une bonne forme c'est celle qui aura accumulé le plus grand indice de confiance

## Mise en œuvre : généralisation

- Dans  $\mathbb{R}^n$ , l'accumulateur est un hyper cube
- Problème continu : impossible à réaliser pour tous les points sauf pour un nombre fini.
- Espace discret et fini : on peut représenter l'accumulateur par une matrice de dimension  $n$

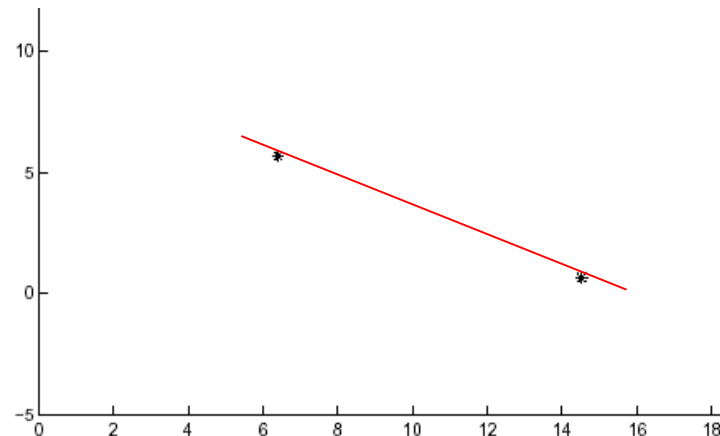
## Cas d'étude : Détection de droite

- 1 point correspond à 1 infinité de droites cartésiennes passant par P
- Droite d'équation  $y = a.x + b$   
→ paramètre (a,b)
- **Unicité de la droite passant par les points P & Q**
- Déséquilibre possible introduit par un 3<sup>ème</sup> point



## Cas d'étude : Détection de droite

- 1 point correspond à 1 infinité de droites cartésiennes passant par P
- Droite d'équation  $y = ax + b$   
→ paramètre  $(a, b)$
- Unicité de la droite passant par les points  $P$  &  $Q$



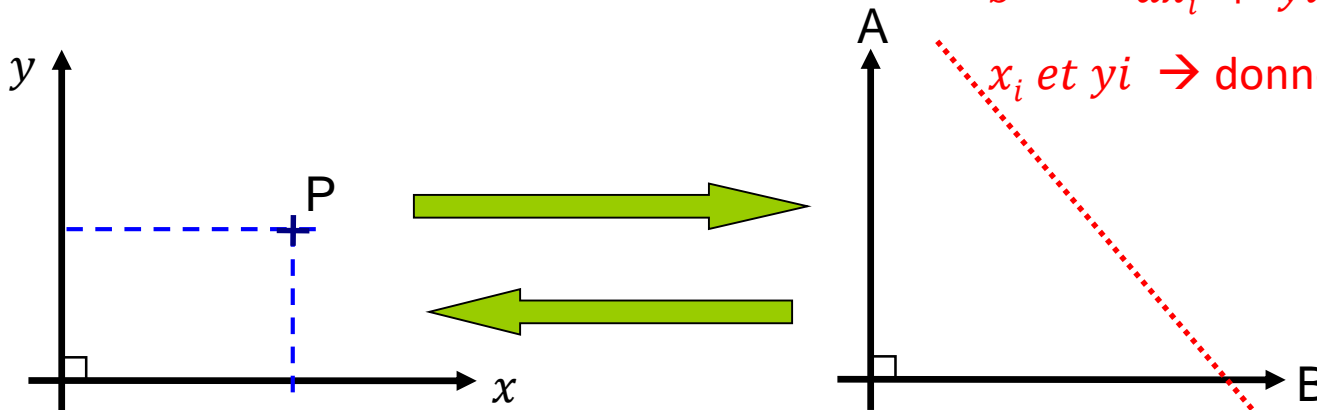
## Transformation

- A un point  $P_i(x_i, y_i)$  de l'espace  $I$ 
  - Association d'une droite d'équation:  $ax + by + c = 0$  à un point  $p(a, b, c)$  de  $H$

Pour une équation simplifiée :

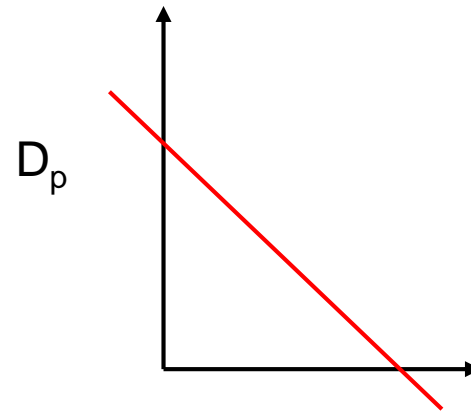
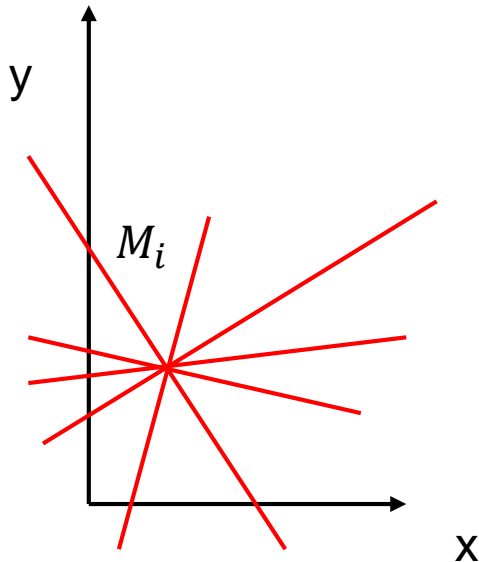
$b = -ax_i + y_i$  ;  $a, b$  : paramètres

$x_i$  et  $y_i \rightarrow$  données



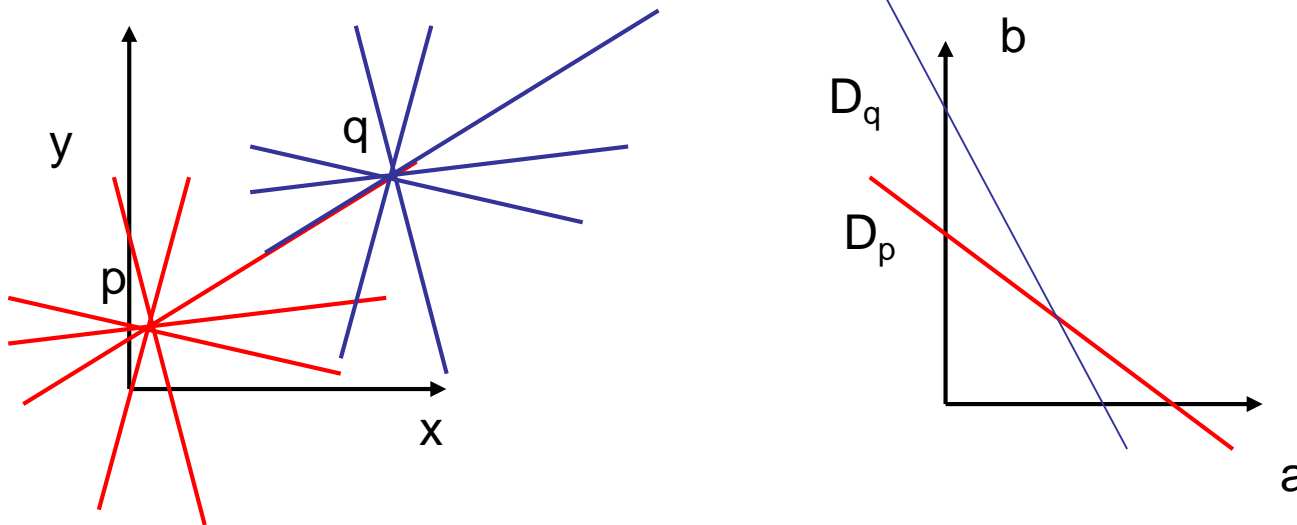
## Transformation

- Toutes les droites passant par un point  $M_i$  correspondent à un seul vecteur  $D_p$  dans l'espace de hough





## Dans une représentation cartésienne de la droite



- Ces deux faisceaux de droites dans l'espace  $(x, y)$  ont en commun la droite qui relie les points  $p$  et  $q$ .
- Cas général : dans le système cartésien, il faut 3 paramètres pour représenter toutes les droites du plan

## Généralisation : *coordonnées polaires*

- Coordonnées polaires

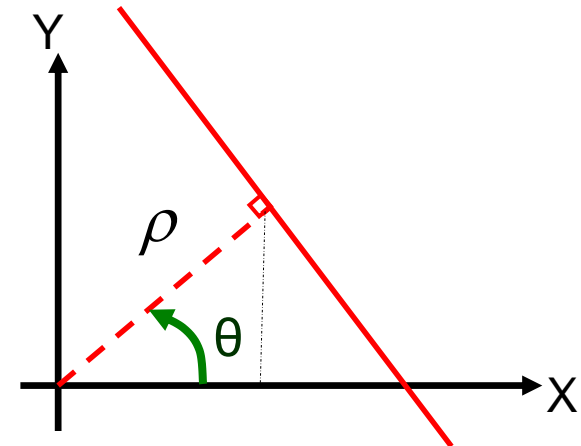
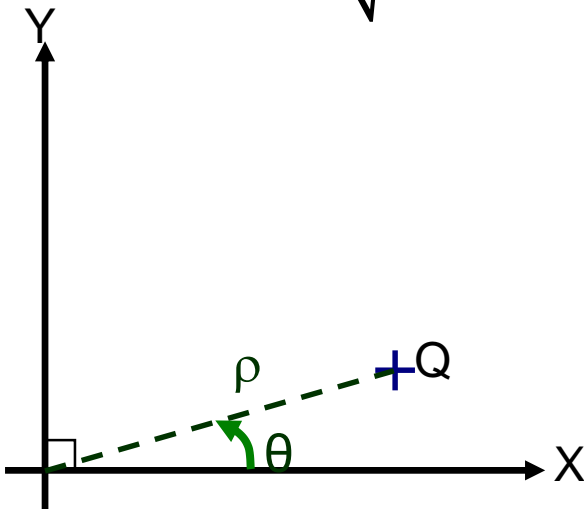
$$\rho = x \cdot \cos(\theta) + y \cdot \sin(\theta)$$

Relation liant  $\rho$  et  $\theta$

→ paramètre  $(r, \theta)$

- Paramètres polaires

$$\rho = \frac{\|x_i y_j - x_j y_i\|}{\sqrt{(x_i - x_j)^2 (y_i - y_j)^2}} ; \theta = \arctan \frac{y_i - y_j}{x_i - x_j}$$

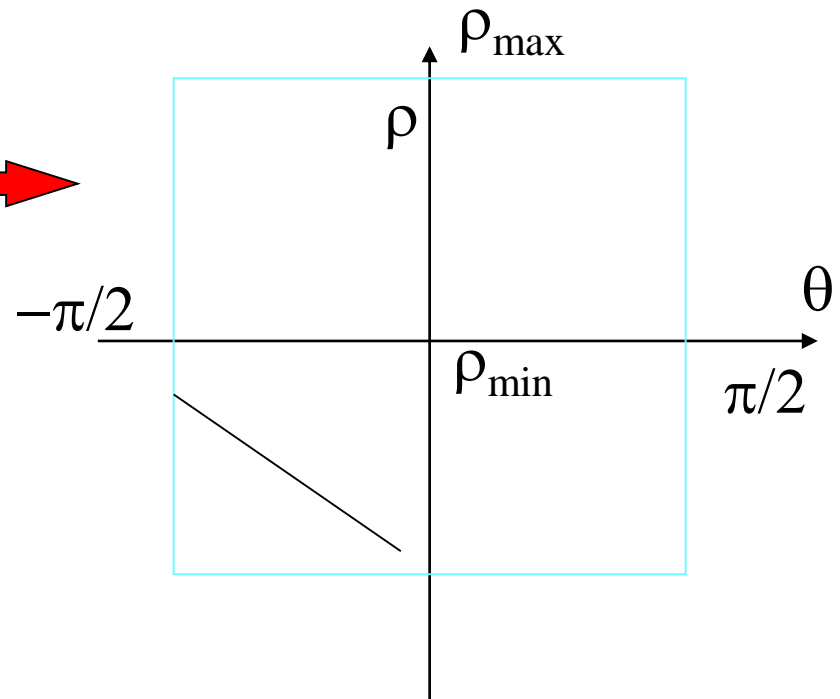
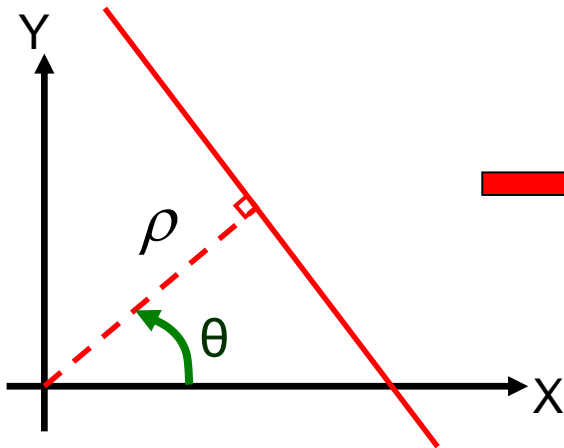


# Transformée de Hough

$$ax + by + C$$



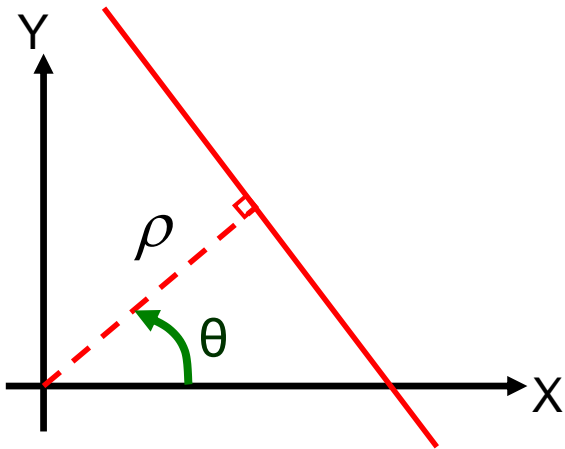
$$x \cos(\theta) + y \sin(\theta) = \rho$$



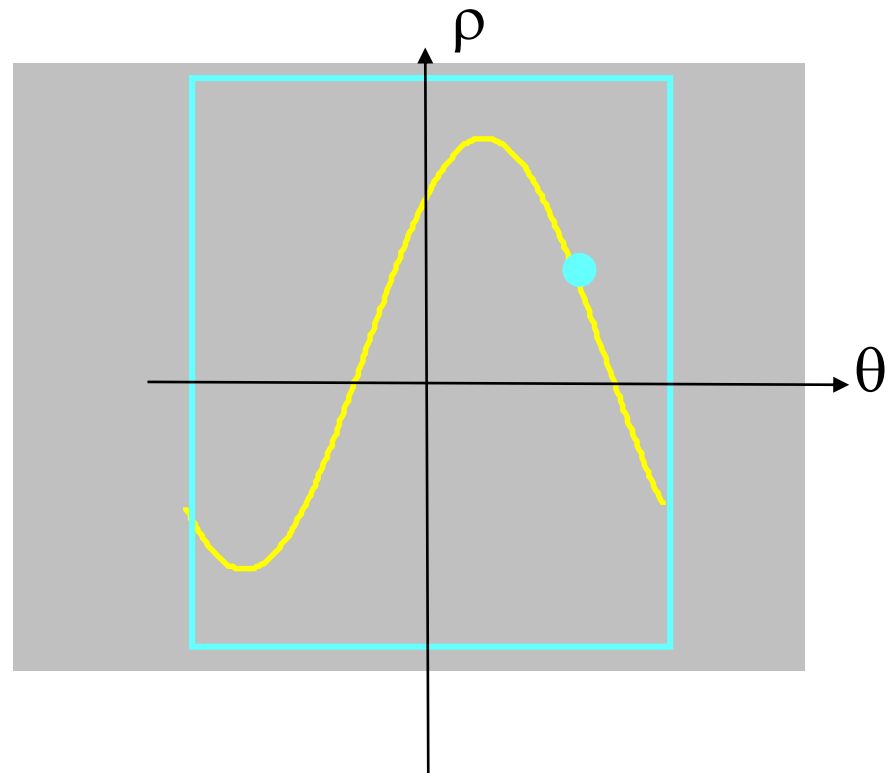
Espace des paramètres

# Transformée de Hough

$$x \cos(\theta) + y \sin(\theta) = \rho$$



Espace image



## Accumulateur

- Un élément de A est un compteur correspondant à une forme unique définie par **une instanciation du vecteur de paramètres**.

- Algorithme de calcul de A:

$$A \leftarrow 0$$

pour tout couple de pixel retenu :

- Calculer les paramètres de la droite passant par les deux pixels,

$$A(\tilde{\rho}, \tilde{\theta}) \leftarrow \left( \frac{\rho}{\Delta\rho}, \frac{\theta}{\Delta\theta} \right)$$

- Convertir les paramètres continus en indice

$$A(\tilde{\rho}, \tilde{\theta}) \leftarrow A(\tilde{\rho}, \tilde{\theta}) + 1$$

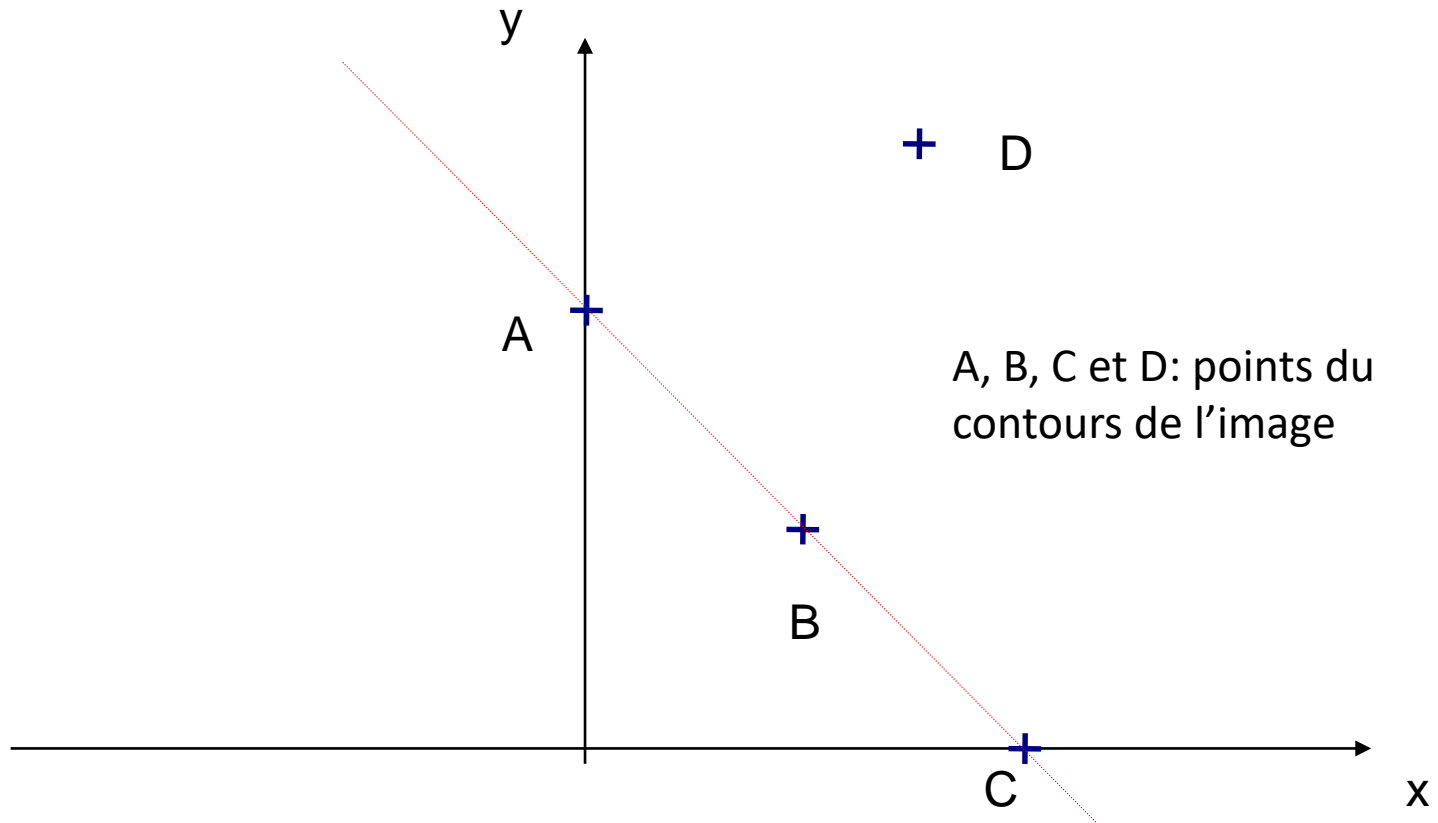
- Mise à jour de l'accumulateur

fin pour

## Calcul de l'accumulateur (exemple)

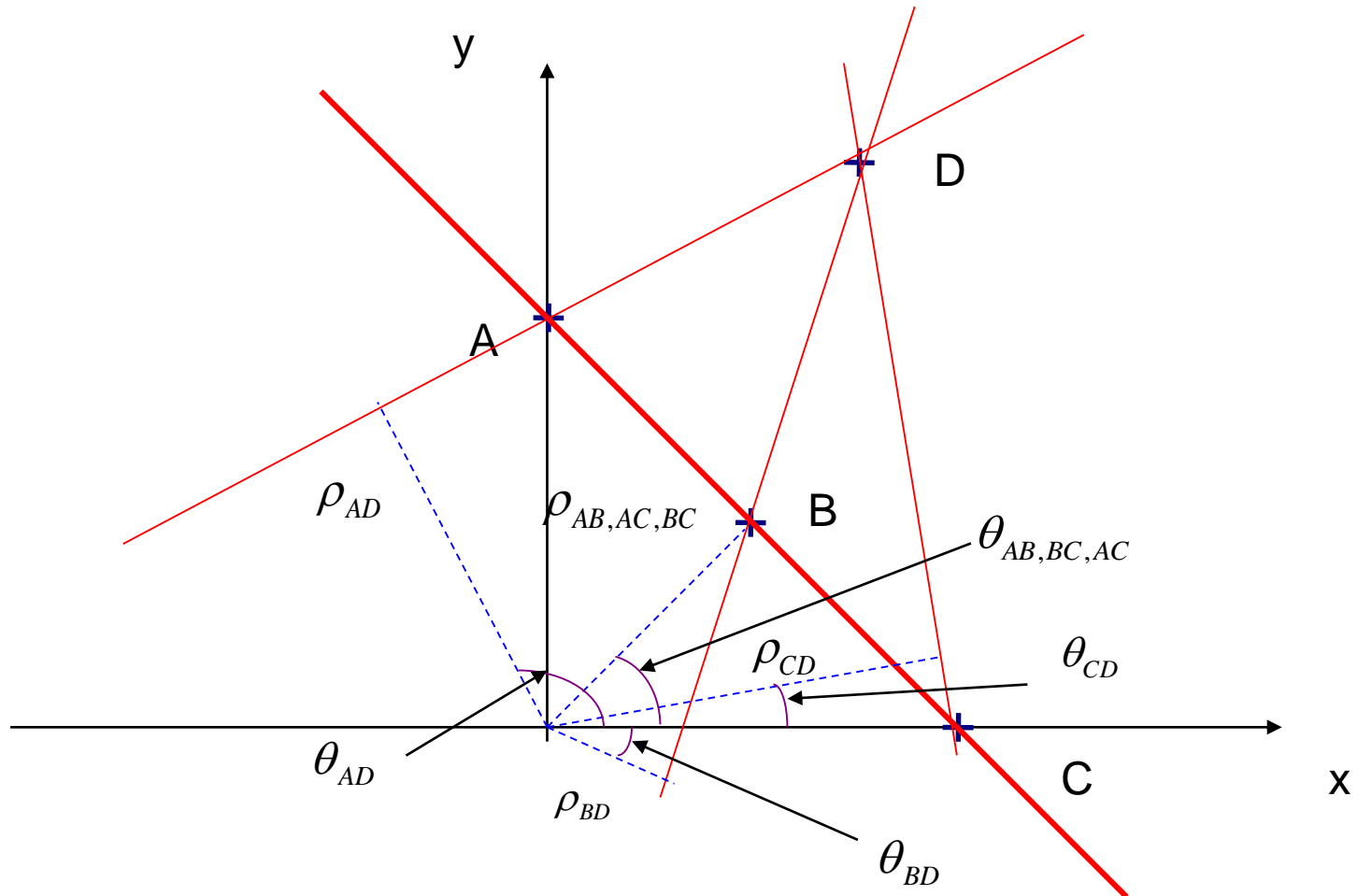
**But de l'algorithme:** détecter la droite formée par A, B et C

**Etape 1:** extraire les contours de l'image

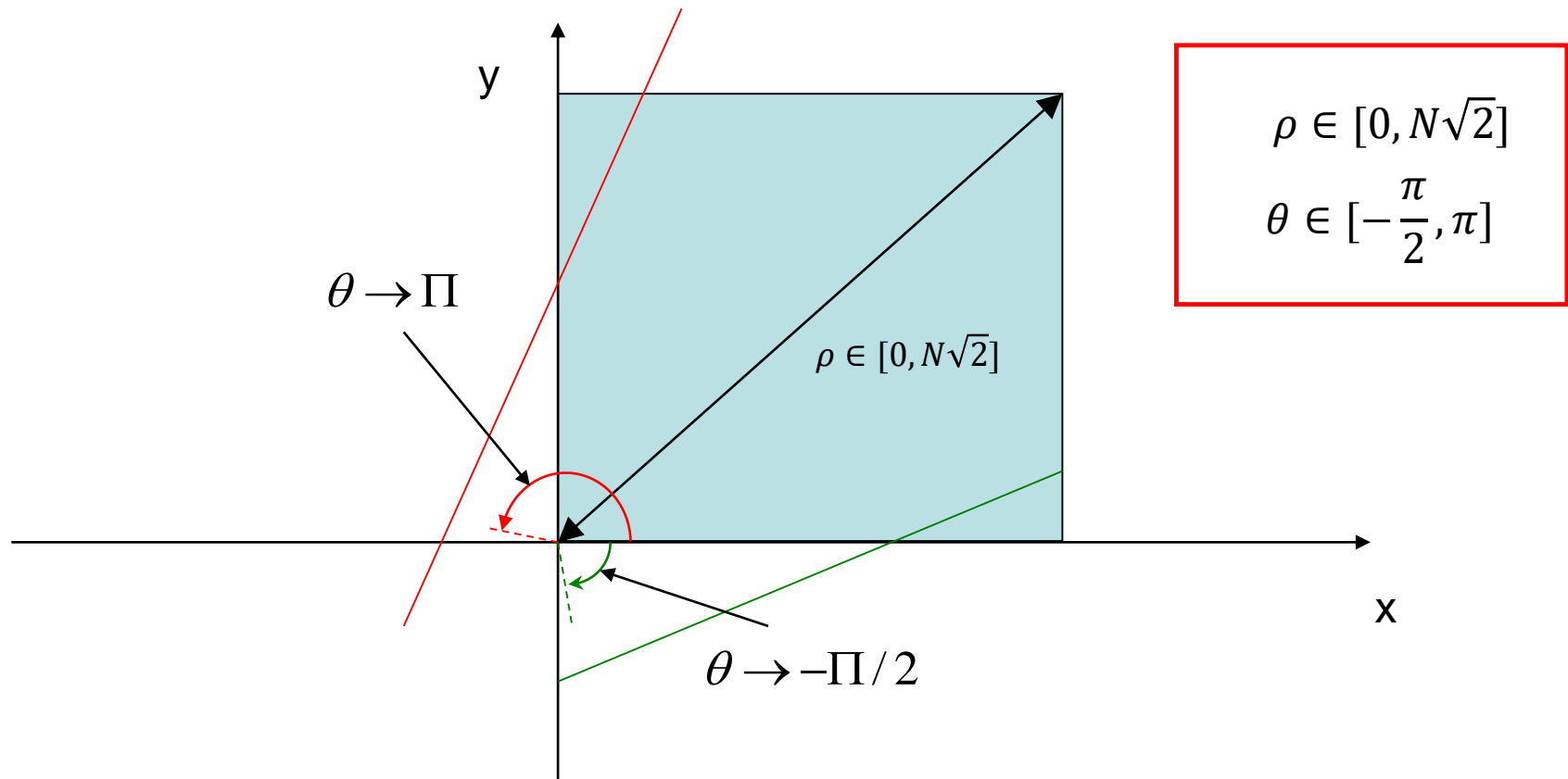


# Calcul de l'accumulateur

**Etape 2:** former toutes les droites possibles ( 6 possibilités)



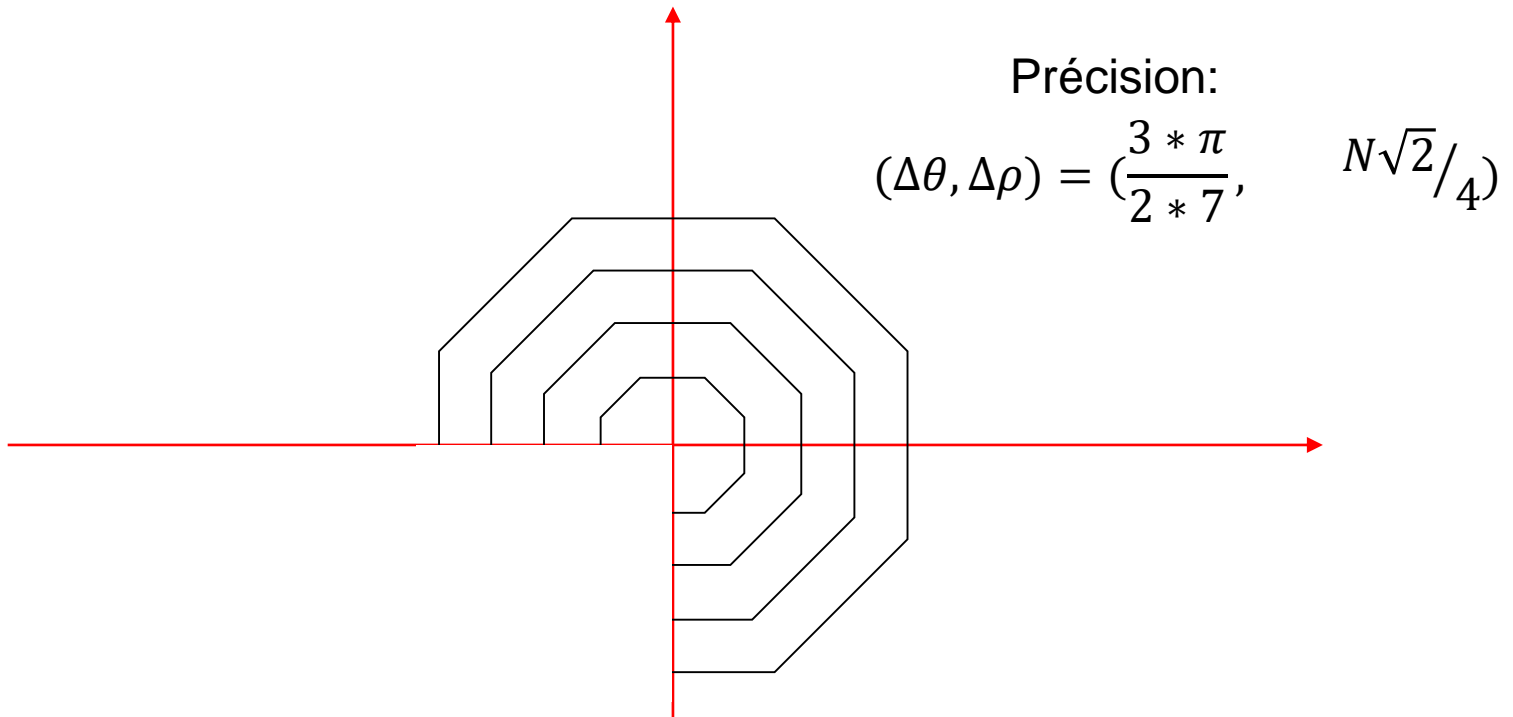
- Domaines de variation de  $\rho$  et  $\theta$  pour une image de taille  $N * N$ .





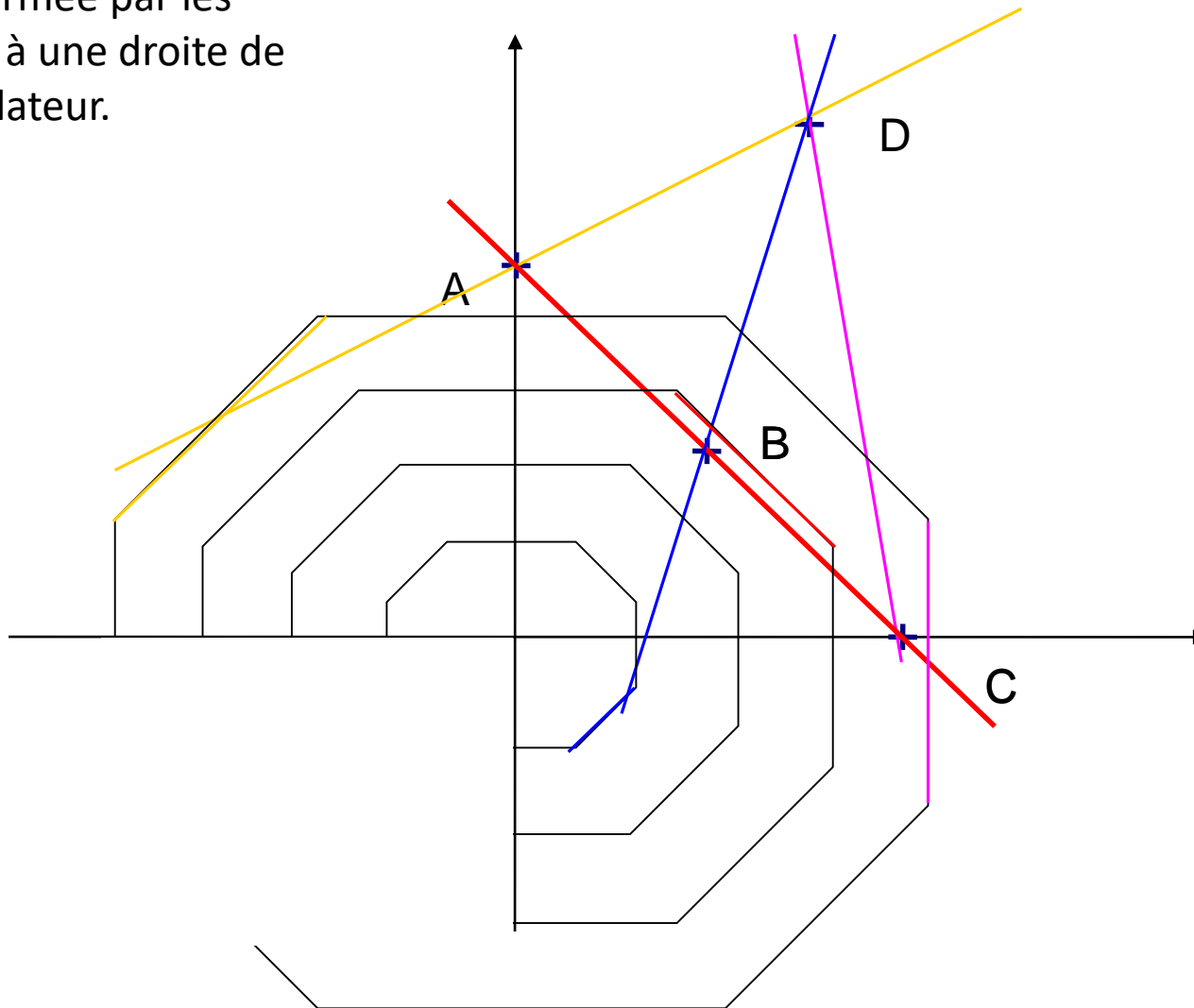
## Calcul de l'accumulateur

Un élément de A est un compteur correspondant à une droite unique définie par  $(\rho, \theta)$  On choisit A de taille  $7 \times 4$ .



## Calcul de l'accumulateur

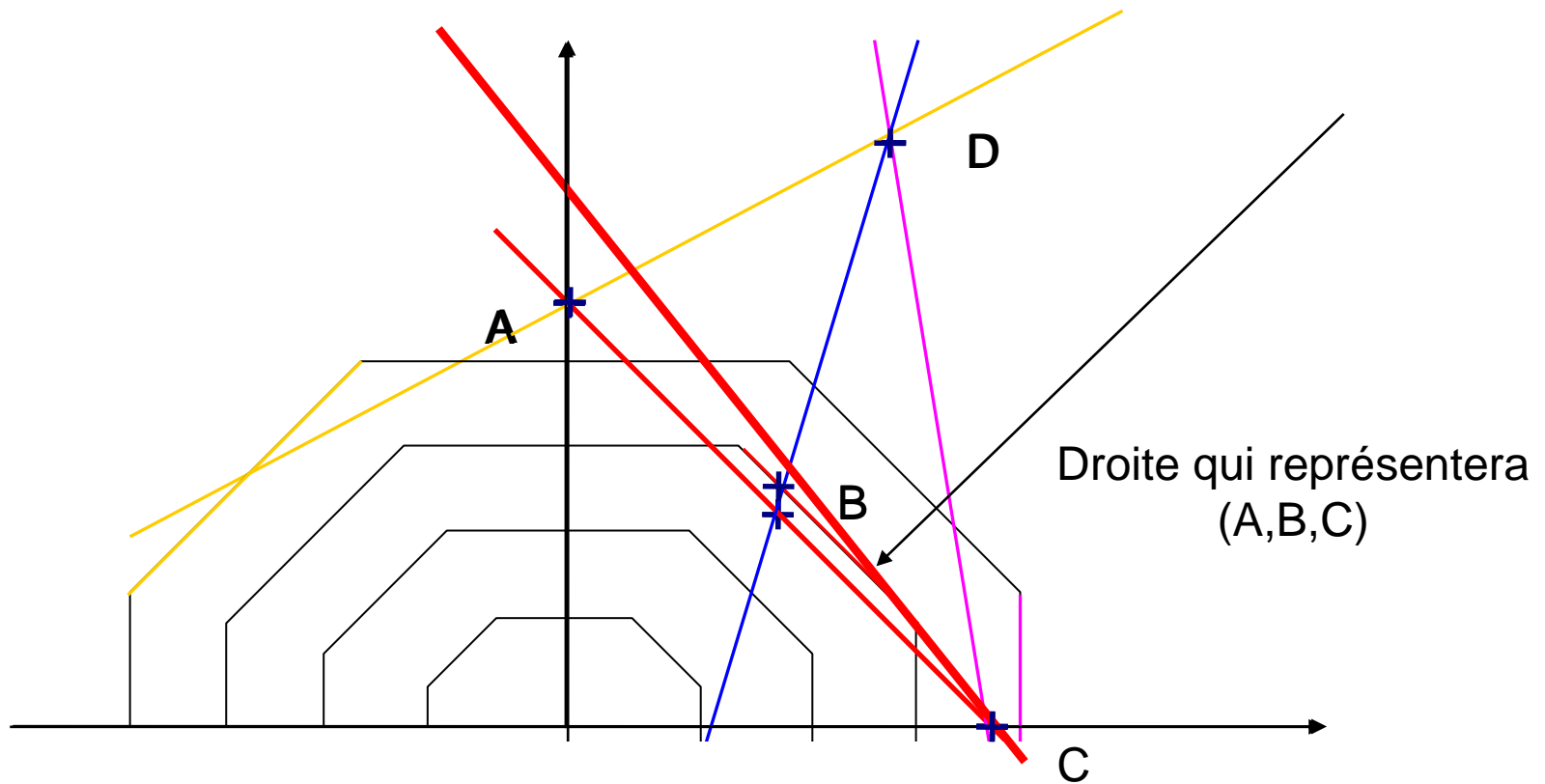
**Etape 3:** associer chaque droite formée par les contours à une droite de l'accumulateur.



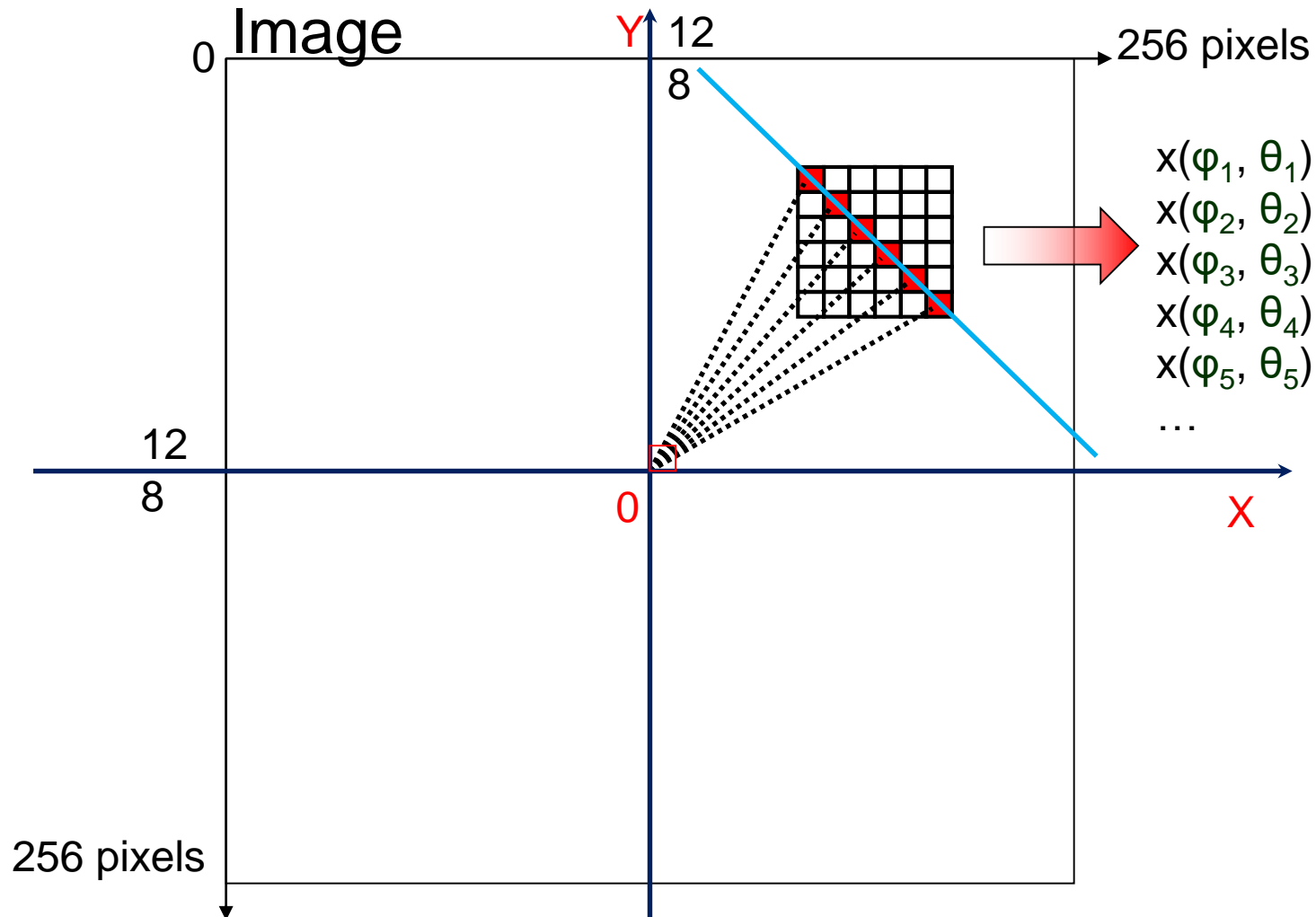
## Calcul de l'accumulateur

### Etape 4: détection des maxima locaux dans l'accumulateur

➤ On en déduit les formes (droites) représentatives



## Détection de droites dans une image



## Détection de droites dans une image

- Vote diversifié ou pondéré par le module du contour
- pour l'aspect complexité, multi-résolution avec un retour vers la résolution voulue
- ....

## Application à la détection de Cercles

- Partir de l'équation paramétrique d'un cercle

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

(a,b) coordonnée du centre du cercle, r rayon du cercle

- Accumulateur : tableau 3D
- Détection : pour un point (contour) (x,y) on incrémente tous les points (a,b,r) qui satisfont l'équation du cercle,
- On sélectionne les maxima de A

Chaque point (x,y) vote pour le vecteur (a,b,r)

- Méthode simple et robuste
$$(x-a)^2 + (y-b)^2 = c^2$$
- Détection de formes multiples:
  - Droites: 2 paramètres
  - Cercles : 3 paramètres
  - Facette plane: 3 paramètres
- Hough 3D: recherche de facettes planes
  - accumulateur de dimension 3 c'est le plus simple
  - au delà → problèmes de complexité (temps) → passer à d'autres structure de données, méthodes, architectures