



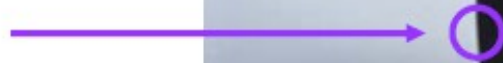
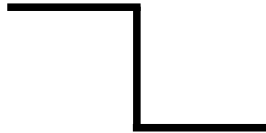
Traitement d'images

Détection de contours

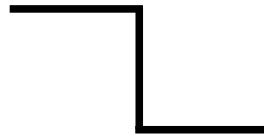
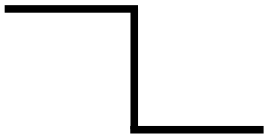
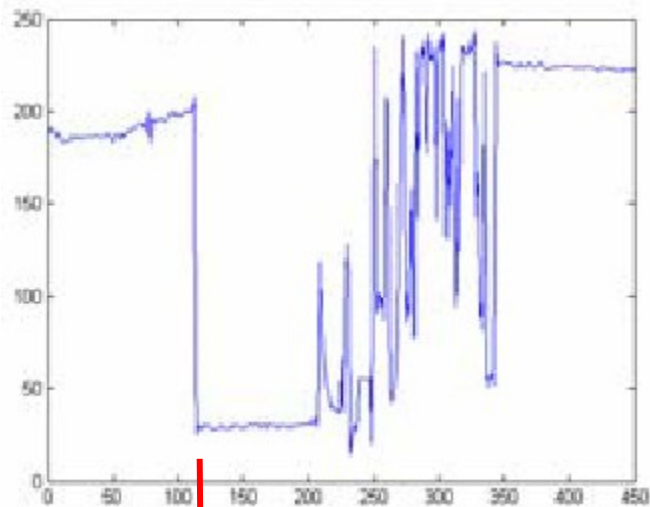
NGUYEN Thi Oanh – IPH
oanhnt@soict.hust.edu.vn

Qu'est-ce qu'un contour ?

Un contour est une variation brusque d'intensité



Qu'est-ce qu'un contour ?



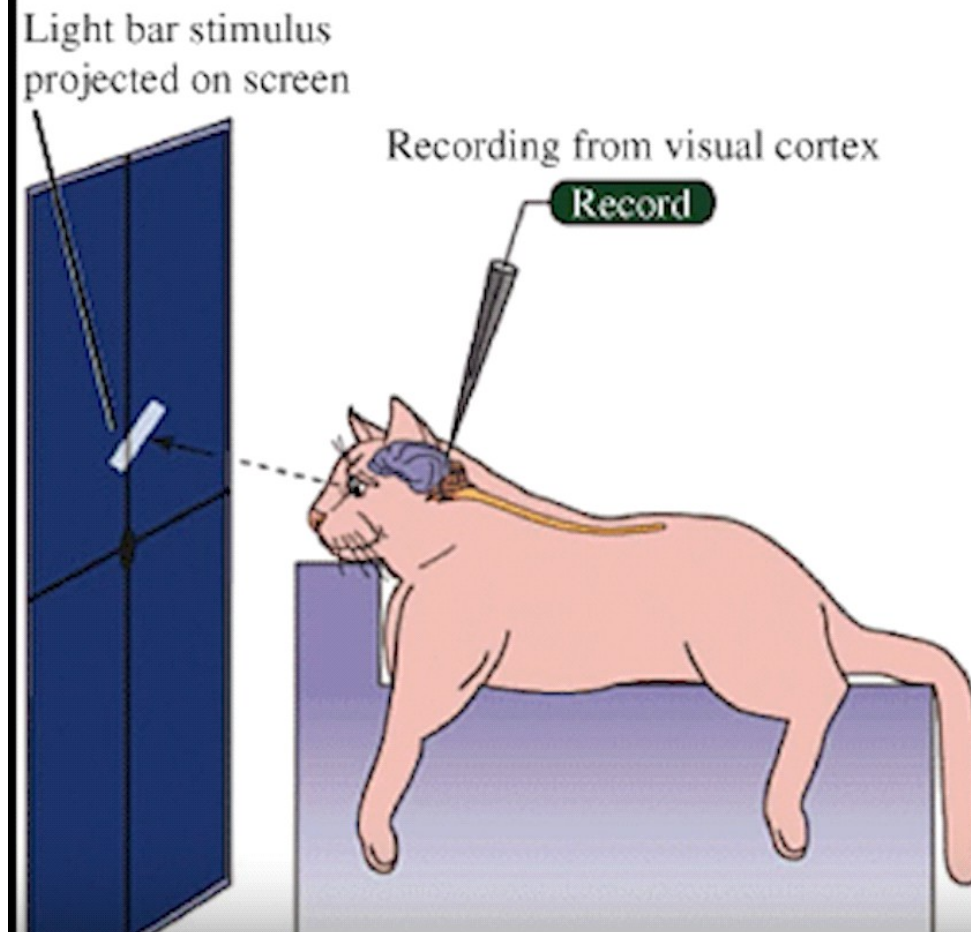
Pourquoi contours ?

- (A) Cave painting at Chauvet, France, about 30,000 B.C.;
- (B) Aerial photograph of the picture of a monkey as part of the Nazca Lines geoglyphs, Peru, about 700 – 200 B.C.;
- (C) Shen Zhou (1427-1509 A.D.): Poet on a mountain top, ink on paper, China;
- (D) Line drawing by 7-year old I. Lleras (2010 A.D.).



Pourquoi contours ?

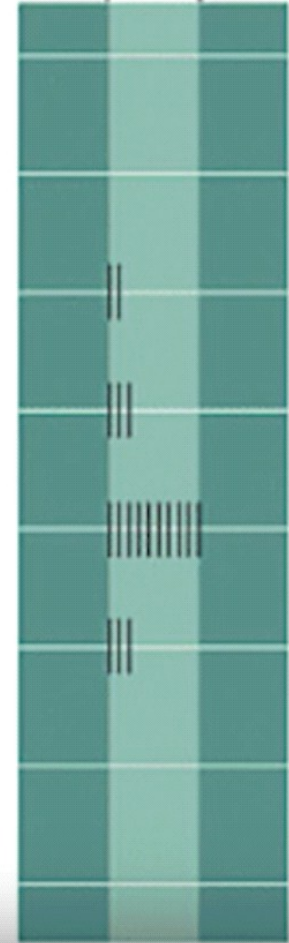
A Experimental setup



B Stimulus orientation



Stimulus presented



Pourquoi contours ?

152 Biederman

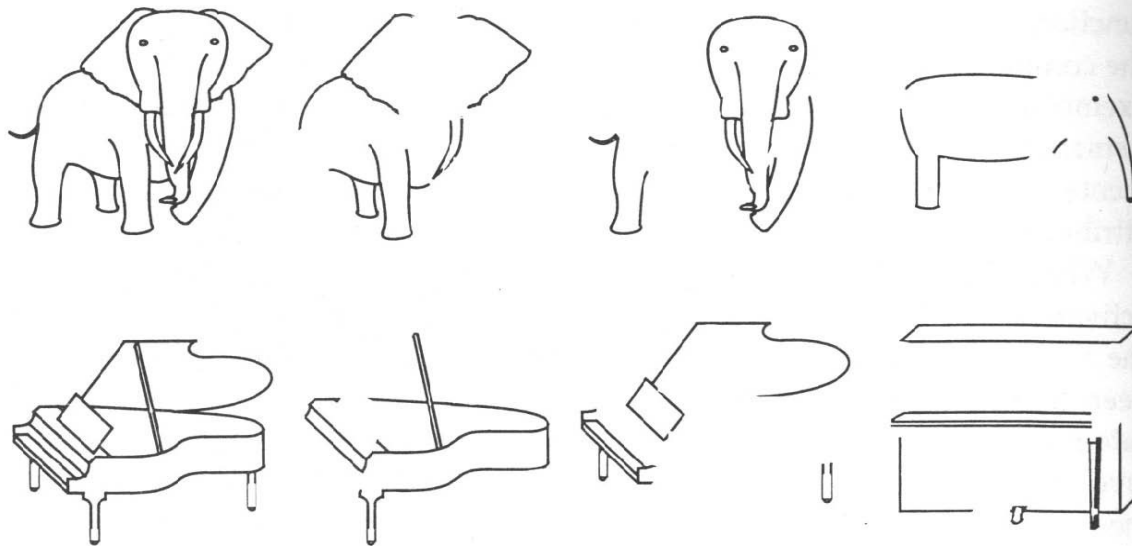


Figure 4.14

Complementary-part images. From an original intact image (left column), two complemen-

Can we recognize these objects?



Définition du contour

- Par définition, un **contour** est la **frontière** qui sépare deux objets dans une image.
 - Une *discontinuité* de l'image
- Dans notre cas, nous détecterons toutes les lignes marquant des **changements d'intensité**
 - *Pas seulement les contours !*
 - *Abus de langage sur la notion de contours !*

Lignes/contours dans une image

Exemples de détection des discontinuités

de profondeur



d'orientation
de surface

de réflectance

d'illumination



Différents types de contours



Marche d'escalier

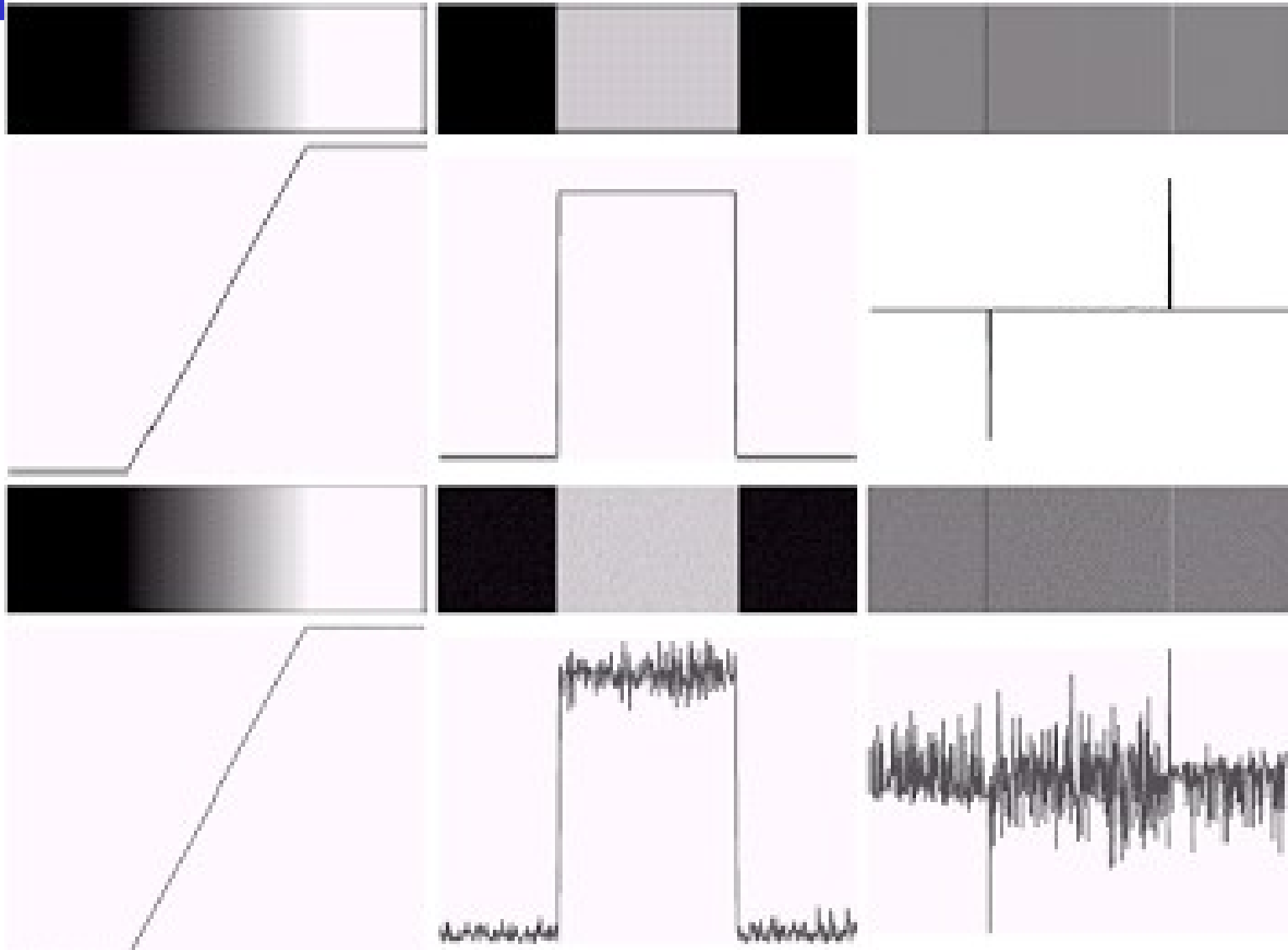


Rampe



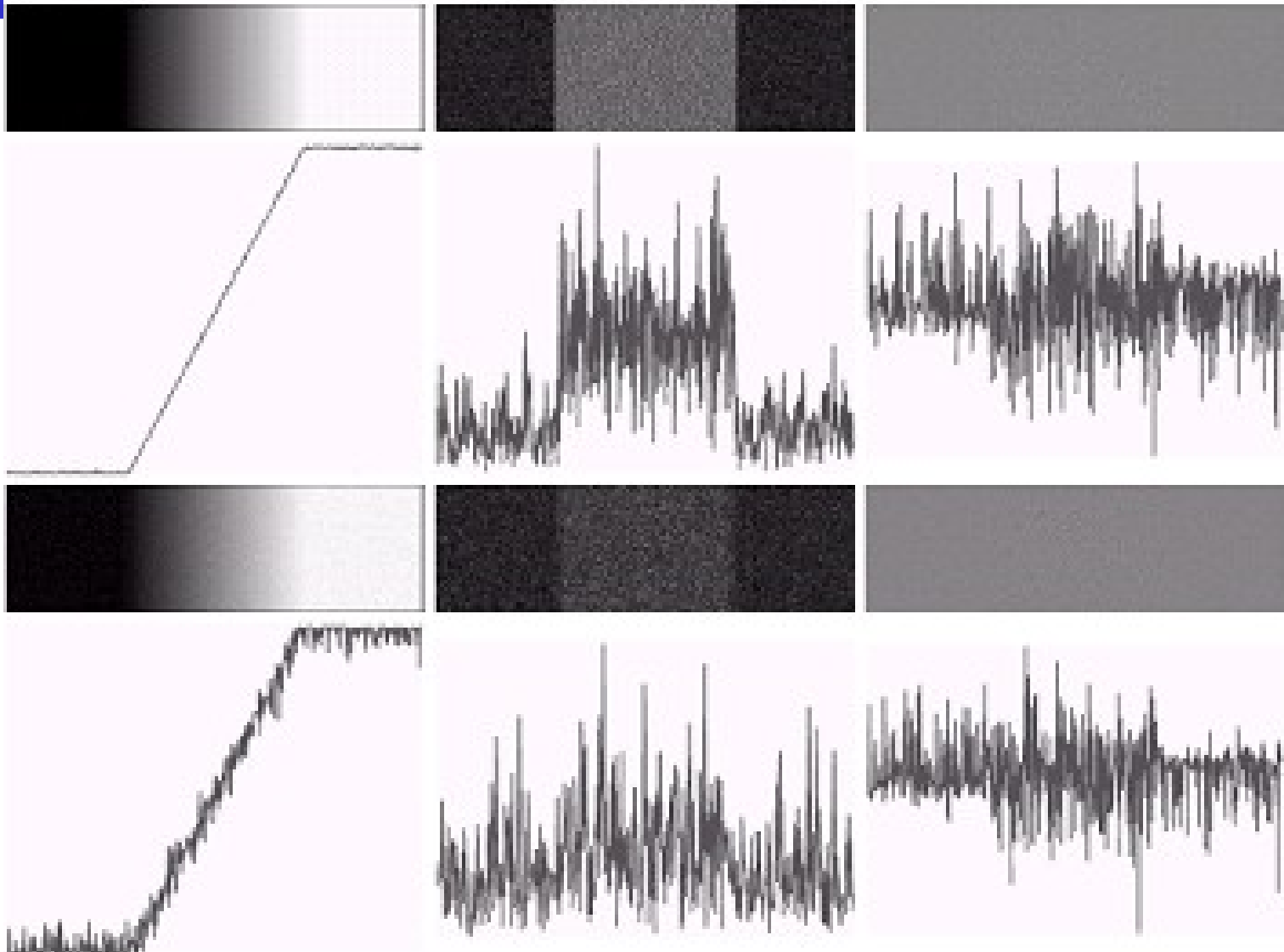
Toit

Contour avec un peu de bruit...





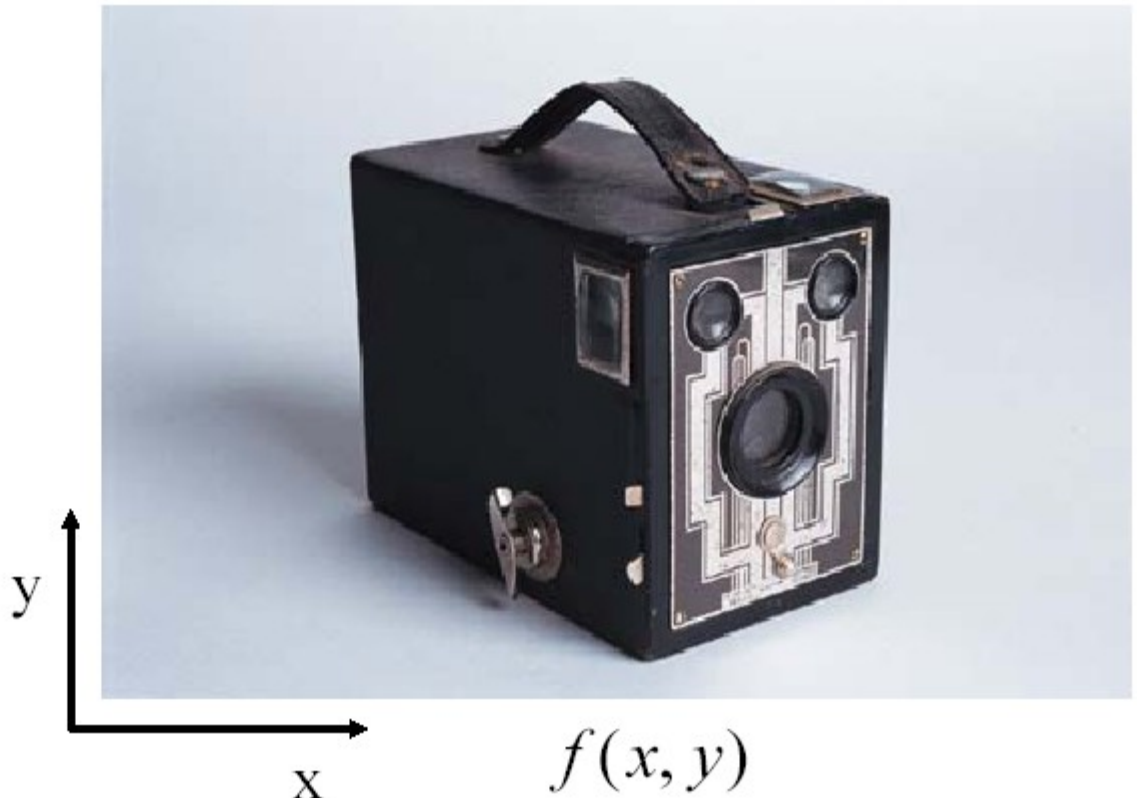
...ou beaucoup de bruit



Dérivée d'une image

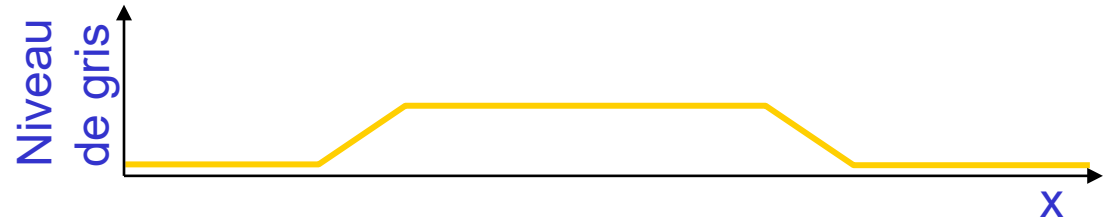
La première dérivée de l'image est l'opérateur de base pour mesurer les contours dans l'image

$$|\nabla f| \equiv \left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2$$

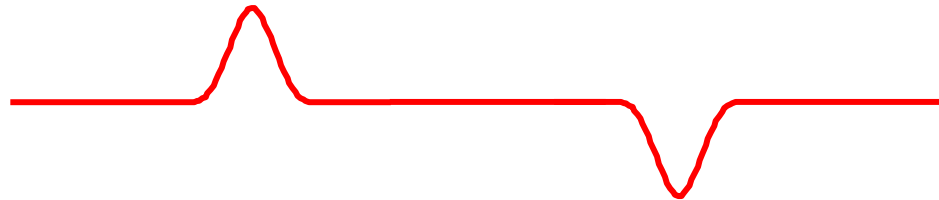


Dérivée d'une image et contours

Image 1D $f(x)$



1ère dérivée $f'(x)$



$|f'(x)|$



Pixels contours:

$|f'(x)| > \text{Seuil}$





Dérivée discrète

On utilise la première dérivée de l'image pour les contours :

$$\frac{\Delta I}{\Delta x} = \frac{I(x + \Delta x) - I(x)}{\Delta x}$$

Approximation simple de la dérivée discrète :

-1	1
----	---

-1
1

ou encore :

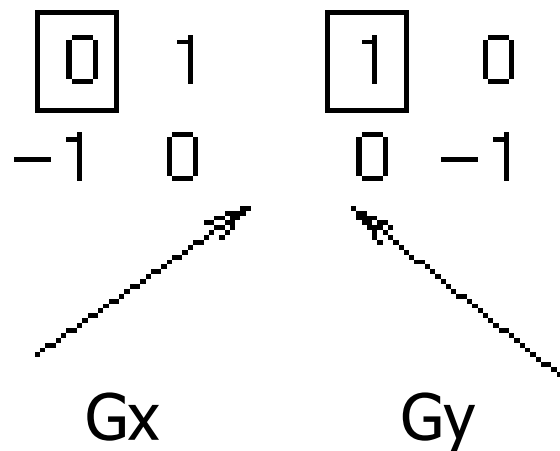
-1	0	1
----	---	---

-1
0
1



Filtre de Roberts

- Roberts (1965) fournit une première approximation de la première dérivée d'une image discrète
- Le calcul se fait avec 2 masques de convolution pour les 2 directions de la dérivée





Filtres pour la détection de contours

Plusieurs autres filtres existent pour la détection des contours dans l'image

On fait **lissage de l'image + dérivée de l'image** (sauf Roberts)

Il existe beaucoup d'autres filtres pour détecter les contours

Roberts:

1	0
0	-1

0	1
-1	0

Prewitt:

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

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

Sobel:

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

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



Lissage + dérivée de l'image

Filtre de Prewitt : Moyenneur + Dérivée

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \times \begin{pmatrix} -1 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} \times \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$$

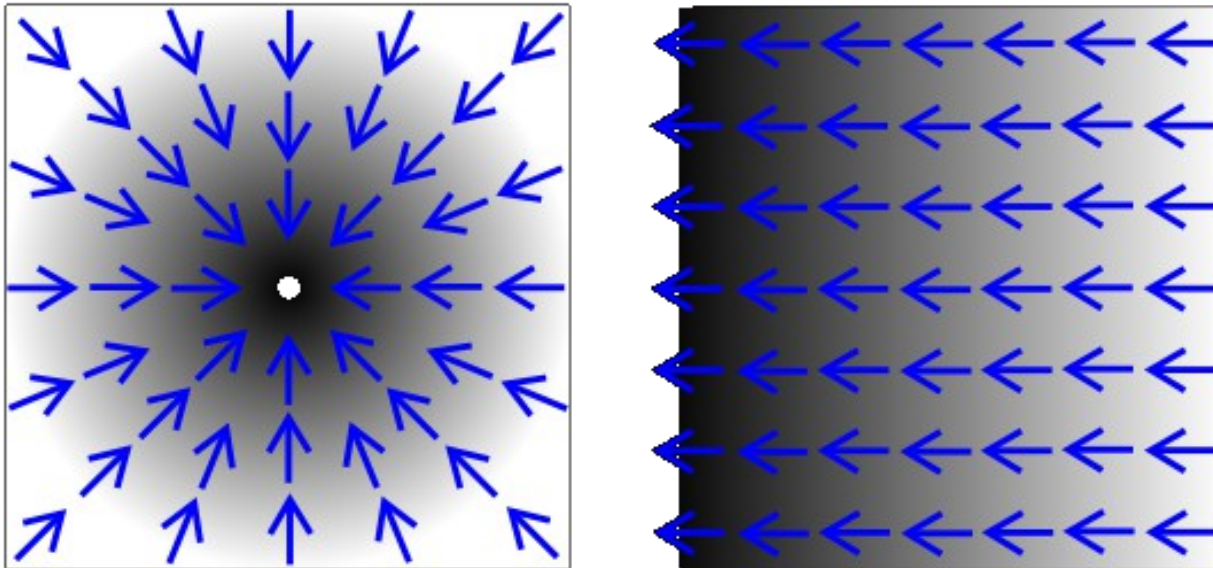
Filtre de Sobel : Gaussienne + Dérivée

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} \times \begin{pmatrix} -1 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} \times \begin{pmatrix} 1 & 2 & 1 \end{pmatrix}$$

⇒ Détection des contours moins sensible au bruit

Gradient de l'image

- En 2 dimensions, nous parlons de gradient de l'image
 - *dérivée en X + dérivée en Y*
 - *vecteur avec une norme et une direction*



Les lignes bleues représentent le gradient de couleur du plus clair vers le plus foncé



Gradient : norme et direction

Norme : Intensité du gradient en chaque pixel
(*mesure la plus utilisée*)

$$|G| = \sqrt{Gx^2 + Gy^2} \approx |Gx| + |Gy|$$

Direction : Direction du gradient le plus fort en chaque pixel

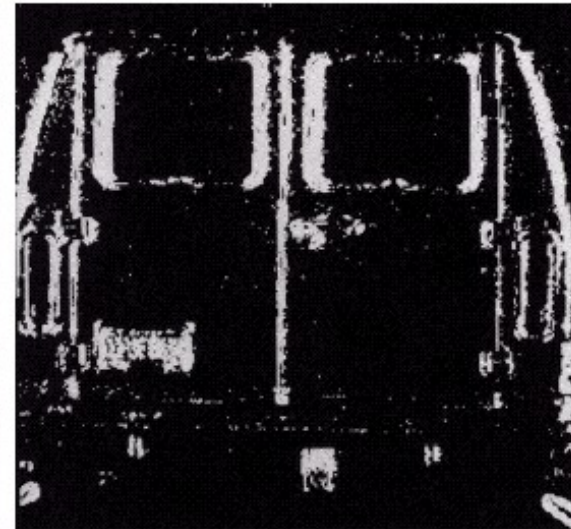
$$\theta = \arctan (Gy / Gx)$$

Exemple de détection de contours

a b
c d

FIGURE 10.16

(a) Input image.
(b) G_y component
of the gradient.
(c) G_x component
of the gradient.
(d) Result of edge
linking. (Courtesy
of Perceptics
Corporation.)

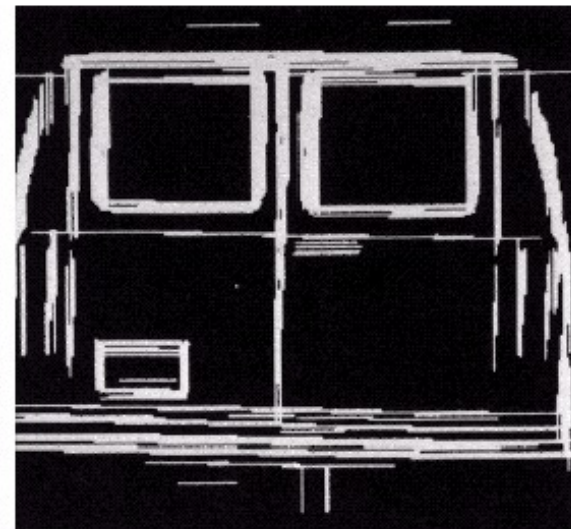
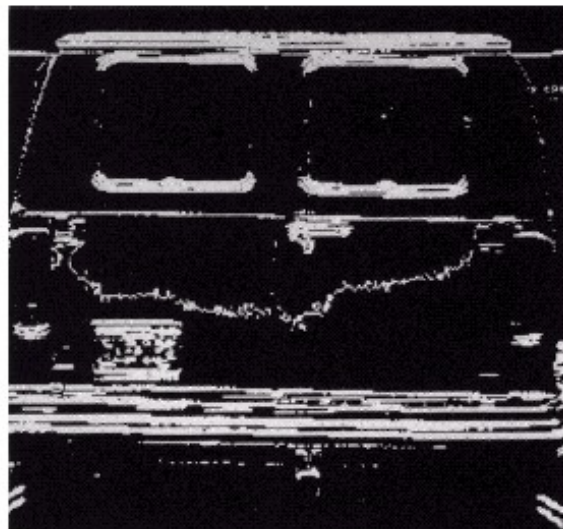


$$\frac{\partial f}{\partial x}$$

Contours
verticaux

$$\frac{\partial f}{\partial y}$$

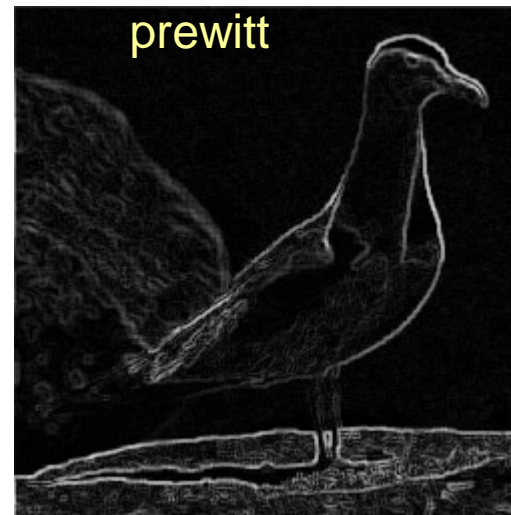
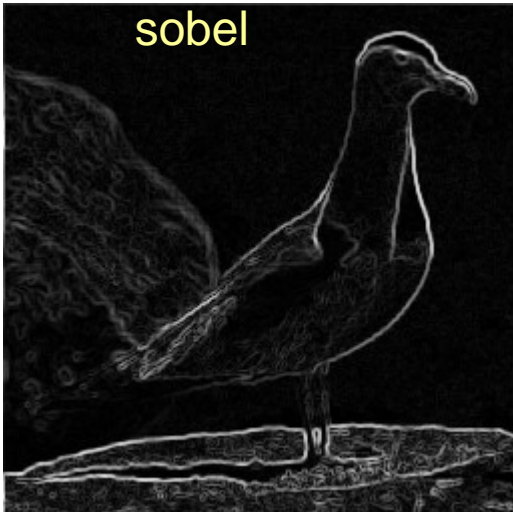
Contours
horizontaux



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

Norme

Exemples de détections de contours



Seuillage du gradient



Détection
avec Sobel
sans seuillage

Seuillage avec
 $S=25$



Seuillage avec
 $S=60$



Deuxième dérivée de l'image

Laplacien



Deuxième dérivée de l'image

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

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



Dérivées de l'image

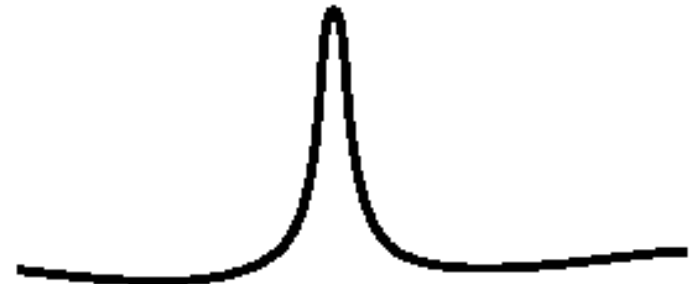
Les contours correspondent :

- Aux maxima de la première dérivée
- Aux passages par zéros de la deuxième dérivée

f



f'



f''

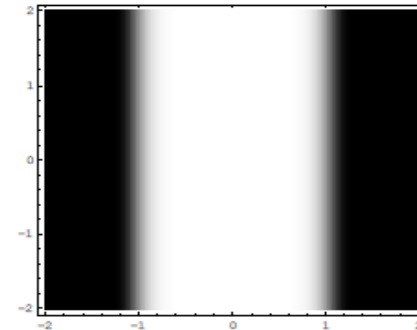
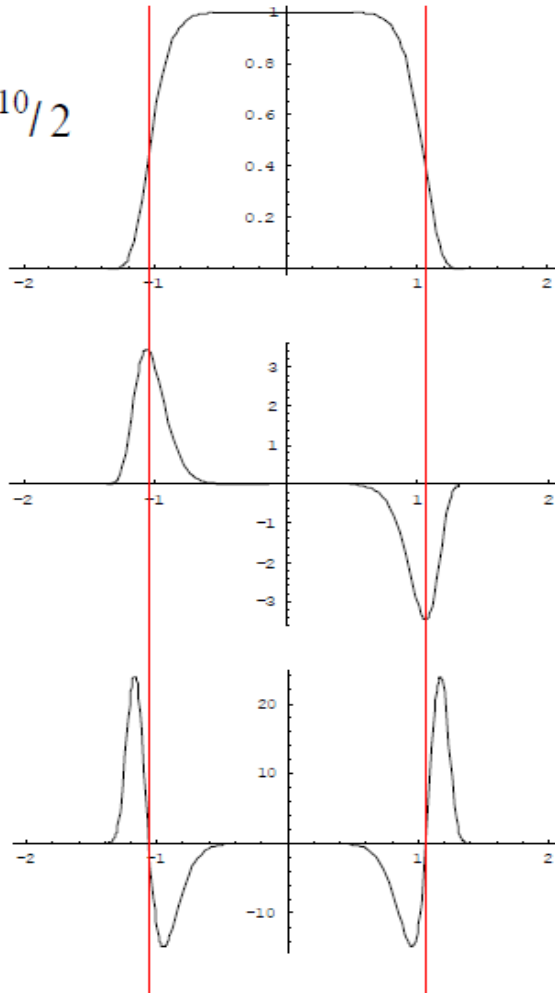


Dérivées de l'image

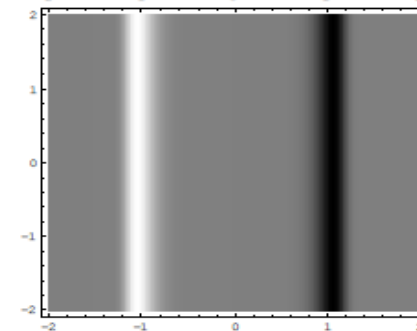
$$f(x, y) = e^{-x^{10}/2}$$

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

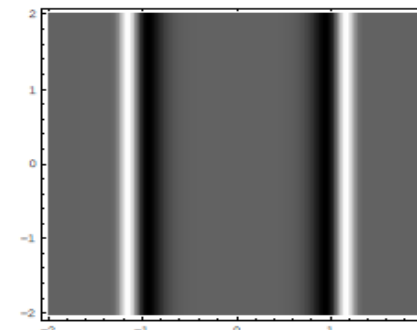
$$\frac{\partial^2 f}{\partial x^2}$$



Image



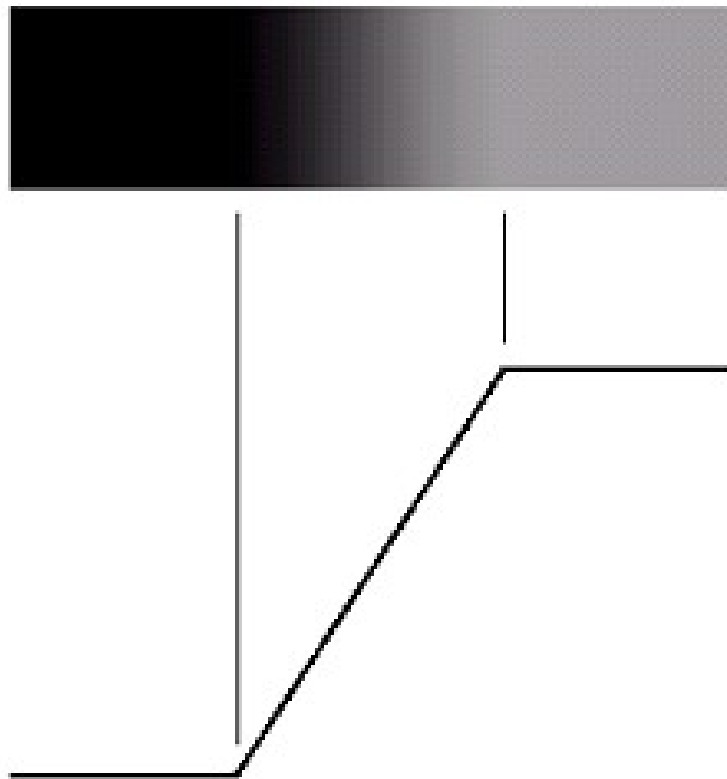
Première
dérivée



Deuxième
dérivée

Exemple avec un contour « rampe »

Détection de la frontière



Première
dérivée

Deuxième
dérivée

Foncé

Clair

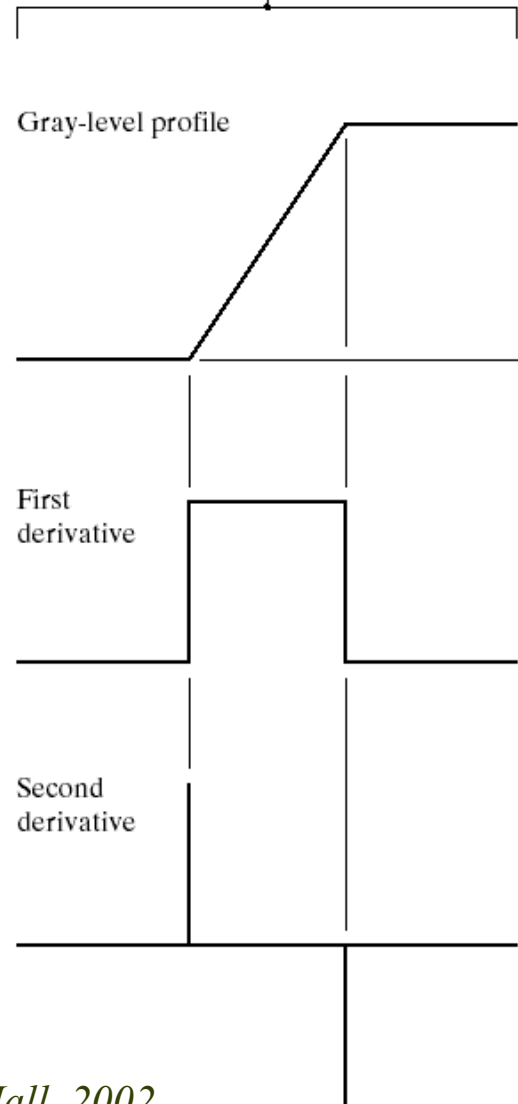
Passage à zéro

Dérivées de l'image

a b

FIGURE 10.6

(a) Two regions separated by a vertical edge.
(b) Detail near the edge, showing a gray-level profile, and the first and second derivatives of the profile.





Laplacien par convolution

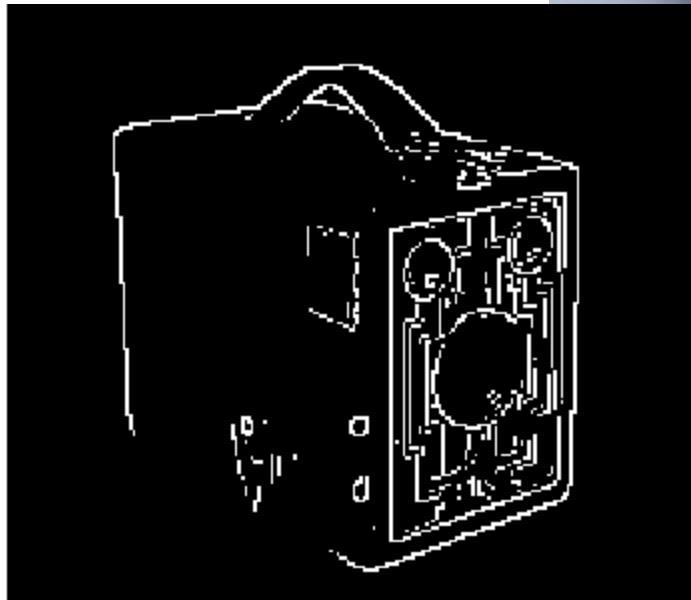
- Plusieurs approximations discrètes du Laplacien existent

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

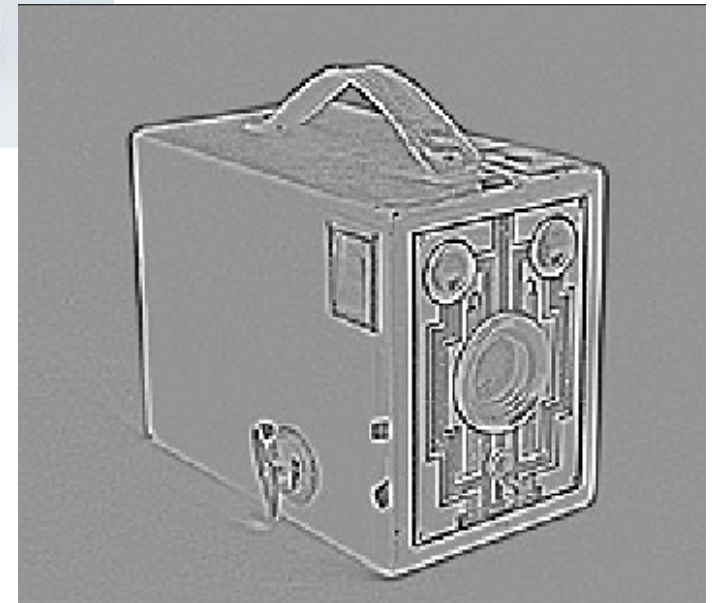
- Une seule matrice de convolution
- Symétrique en rotation

Comparaison Gradient / Laplacien

Gradient



Laplacien





Filtrage optimal

Filtre de Canny



Filtrage optimal : Canny

- Filtre optimal pour la détection des contours
 - *Filtre en plusieurs étapes (pas seulement une convolution)*
- Etant donnés
 - un modèle de contour (marche)
 - un modèle de bruit (blanc gaussien)
- Caractériser les performances en termes de :
 - **détection** (surtout pour les contours faibles)
 - **localisation** (contour détecté proche du contour réel)
 - **réponse unique** (un contour = une seule réponse)



Filtre de Canny : étapes (1)

1 – Appliquer un filtre Gaussien sur l'image

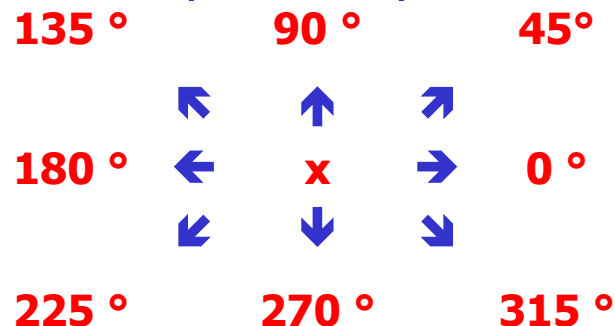
- Filtre passe-bas pour enlever le bruit

2 – Calculer l'intensité du gradient dans l'image

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

3 – Calculer les directions du gradient dans l'image

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

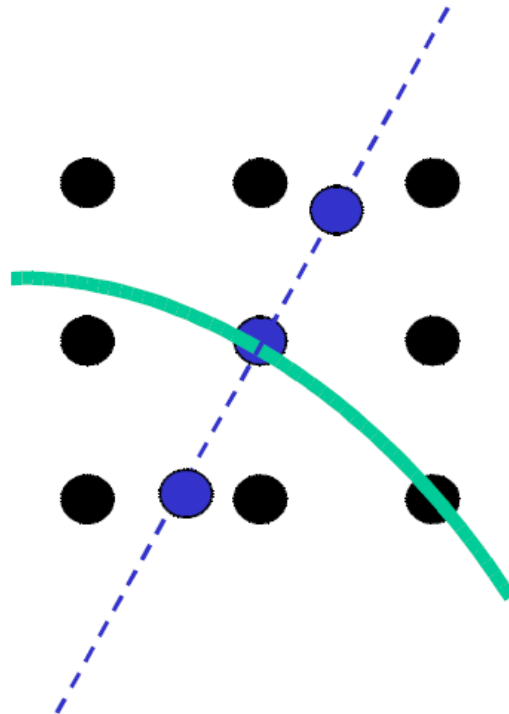




Filtre de Canny : étapes (2)

4 – Suppression des non-maxima

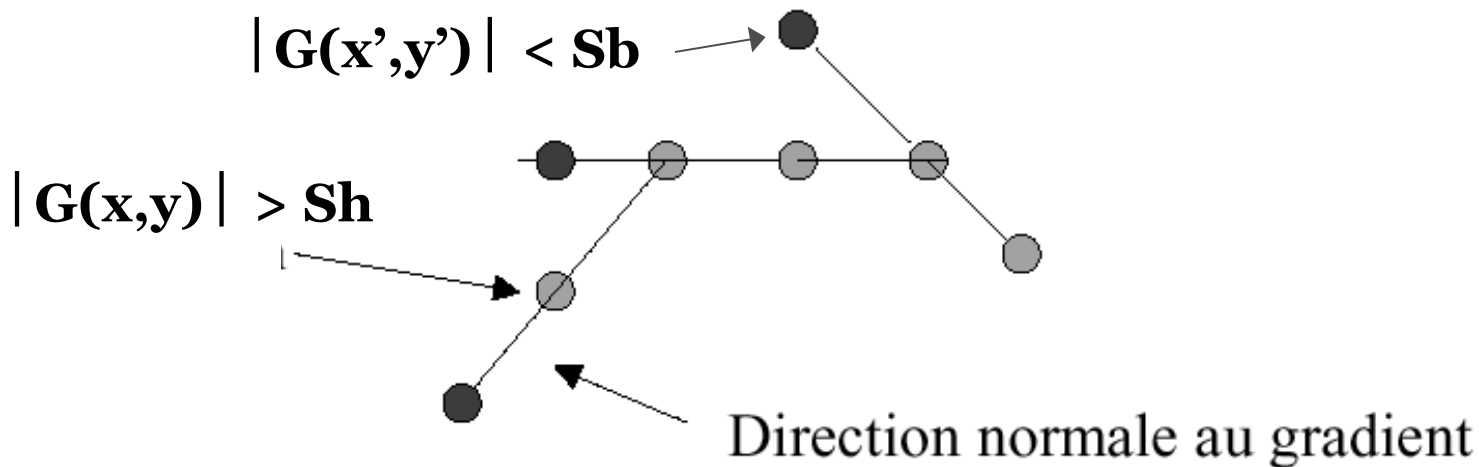
- Si la norme du gradient en un pixel (x,y) est inférieure à la norme du gradient d'un de ses 2 voisins le long de la direction du gradient, alors mettre la norme pour le pixel (x,y) à zéro



Filtre de Canny : étapes (3)

5 – Seuillage des contours (hystérésis)

- Utilise deux seuils : un seuil haut (S_h) et un seuil bas (S_b)
- *Pour chaque pixel de la norme du gradient :*
 - Si $\text{norme}(x,y) < S_b$, alors le pixel est mis à zéro (non-contour)
 - Si $\text{norme}(x,y) > S_h$, alors le pixel est contour
 - Si $S_b \leq \text{norme}(x,y) \leq S_h$, alors le pixel est contour s'il est connecté à un autre pixel déjà accepté comme contour



Filtre de Canny : exemple

Image
originale



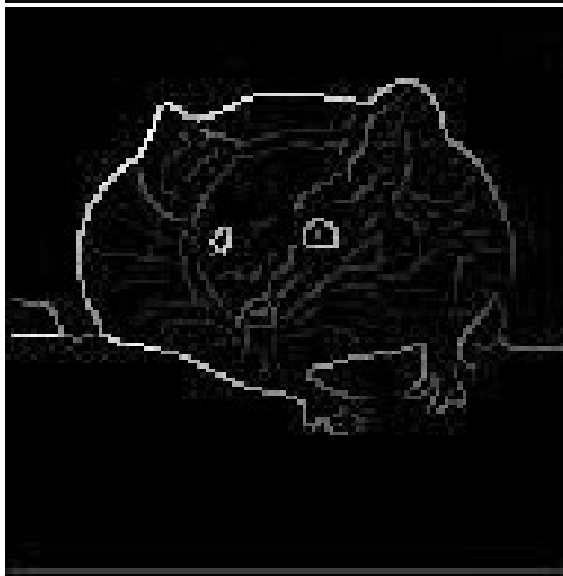
Sobel



Seuillage



Suppression des
non-maxima





Quel filtre choisir pour les contours ?

- 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 manques*
 - *il y a des erreurs de position et d'orientation des pixels contours*
- Chacun semble avoir sa préférence pour une méthode ou une autre
- Un opérateur de détection de contour n'est qu'une première étape dans la chaîne de segmentation



Approche globale pour les contours

Transformée de Hough (optionnel)



Approche globale pour les contours

- Il existe des **approches globales** pour les contours
 - On ne recherche pas seulement des pixels contours
 - On cherche le contour au complet
- On cherche plusieurs pixels correspondant à un contour
 - Comment définir le contour ? Problème ?
- Différentes techniques
 - Ici : Transformée de Hough



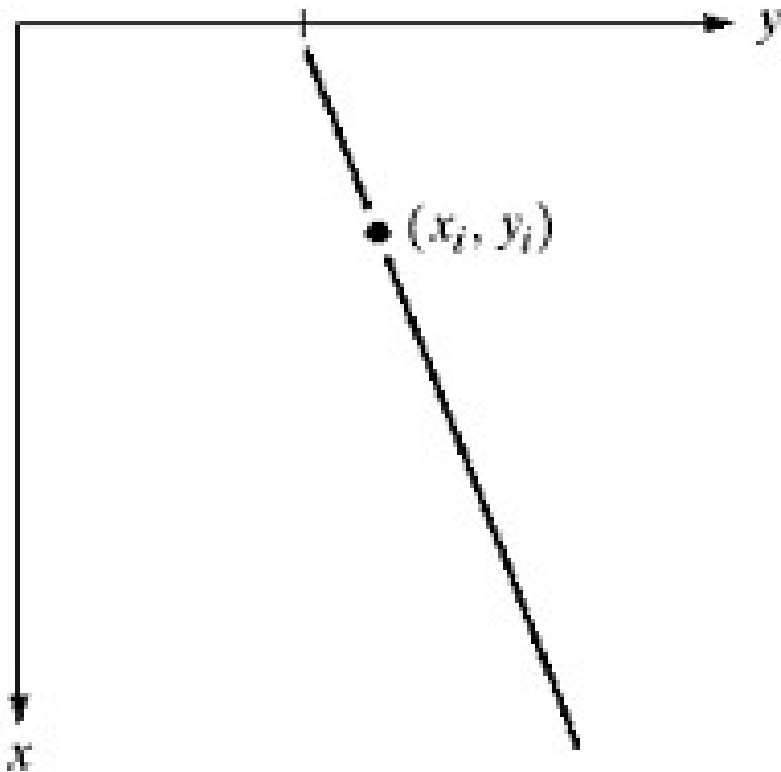
Transformée de Hough

- Approche **globale** pour détecter des contours continus
 - *Du plan x - y au plan paramétrique a - b*
- *Plan x - y*
 - $y_i = a x_i + b$
 - Une infinité de lignes passent par (x_i, y_i)
 - Une seule ligne pour la paire (a, b)
- *Plan paramétrique a - b*
 - $b = -x_i a + y_i$
 - Une seule ligne pour la paire (x_i, y_i)
 - Une infinité de lignes passent par (a, b)



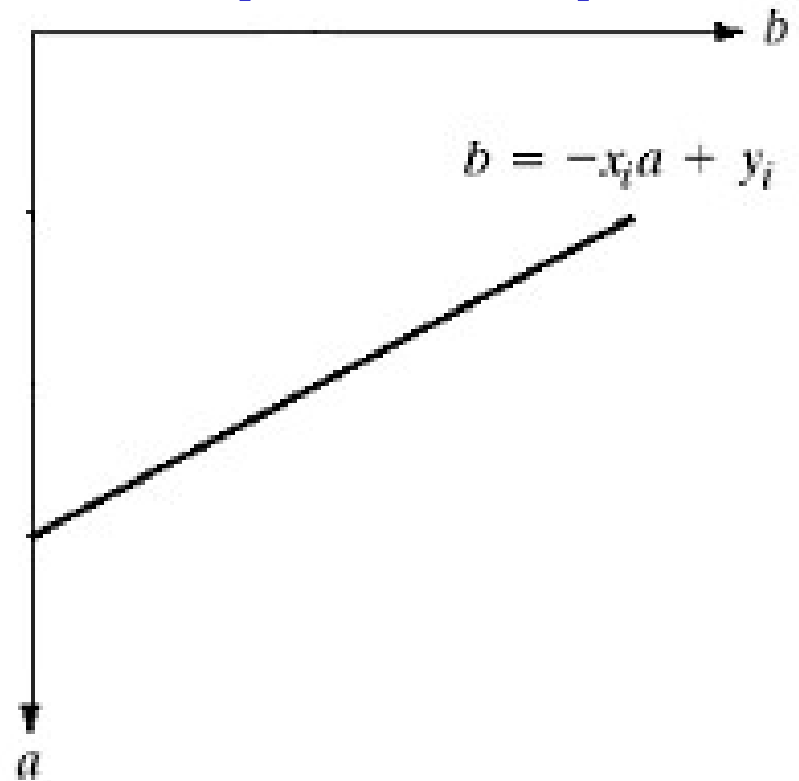
Plan x-y vs Plan a-b

Plan x-y



$$y_i = a x_i + b$$

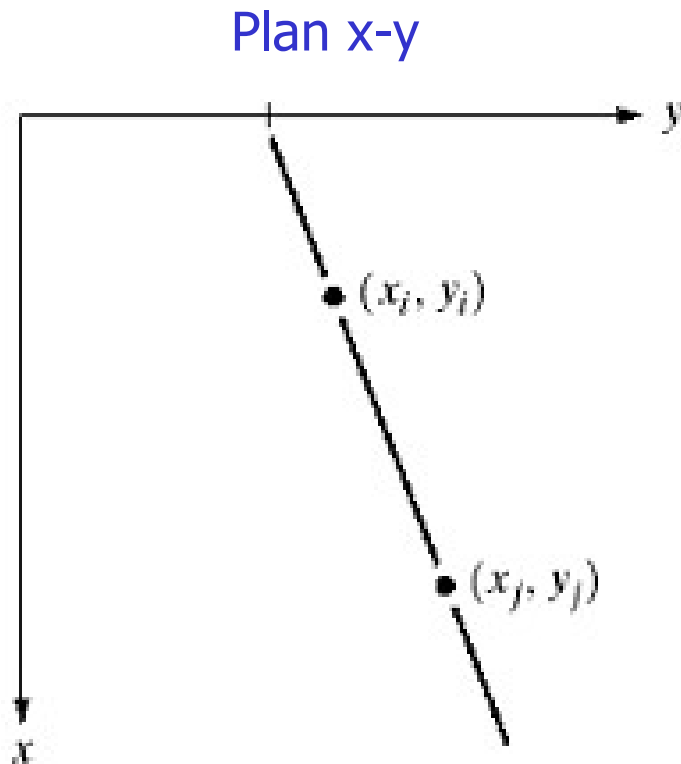
Plan paramétrique a-b



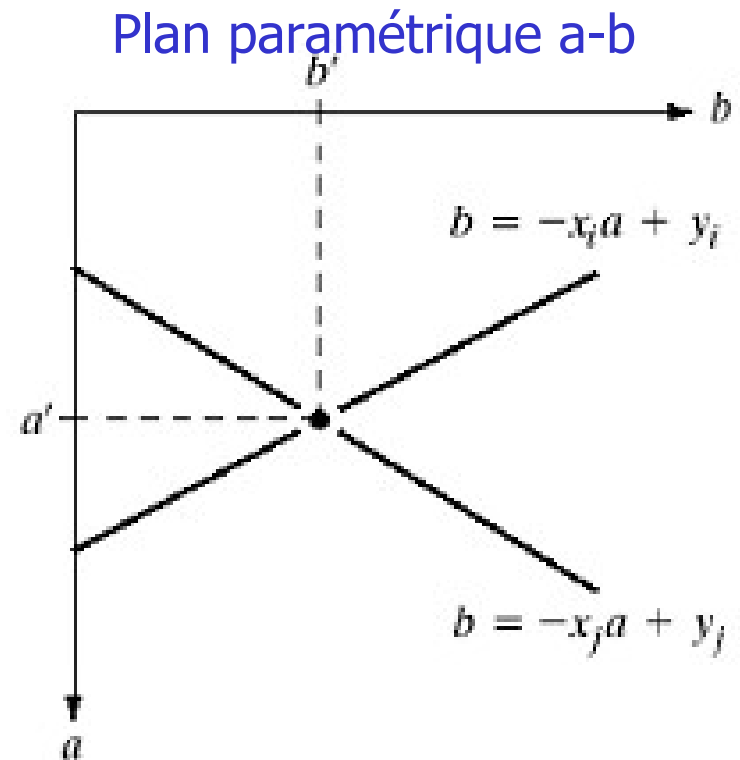
$$b = -x_i a + y_i$$

Droites vs Points

Tous les points (x,y) sur une ligne du plan x - y passent par un seul point (a', b') dans le plan paramétrique a - b



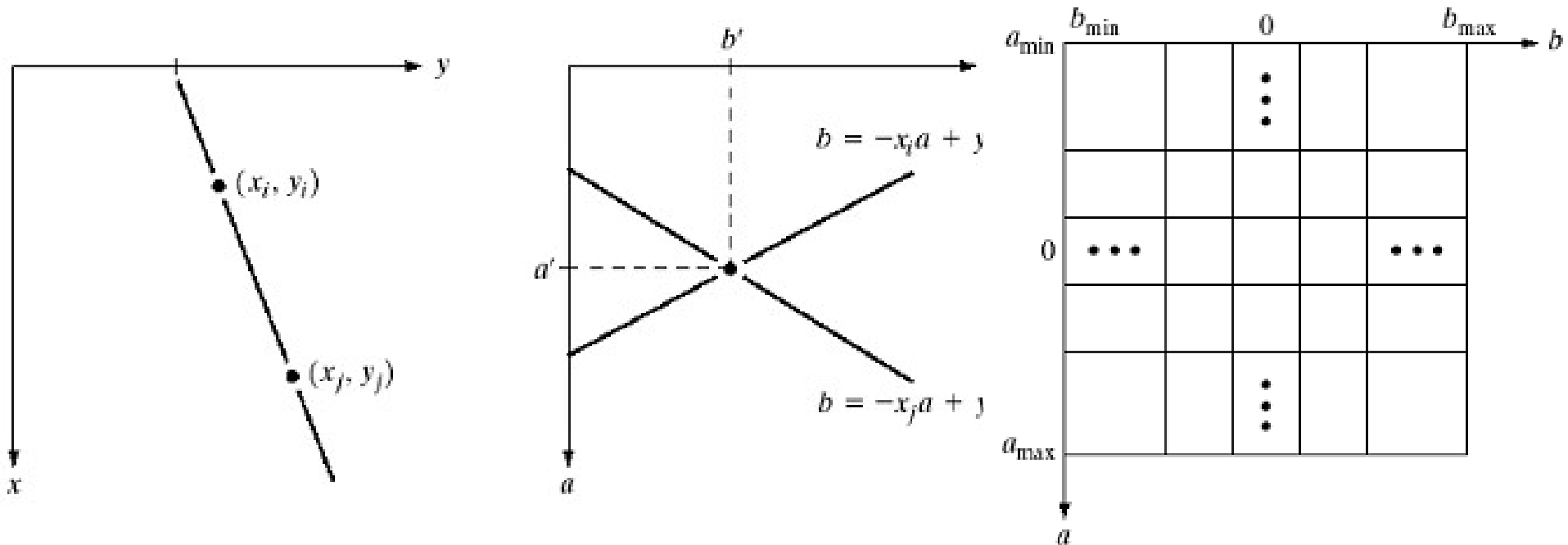
$$y_i = a x_i + b$$



$$b = -x_i a + y_i$$

Principe de la transformée de Hough

- Cellules d'accumulation - Matrice (a,b)
- On construit une **image des votes**
 - *chaque point permet de voter pour une droite particulière*
- Les droites recevant le plus de votes sont conservées





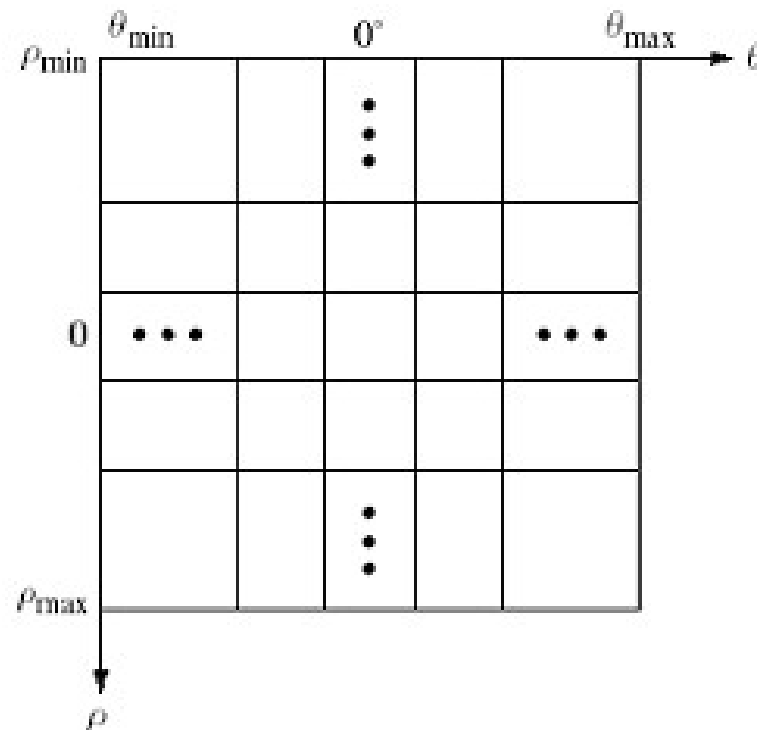
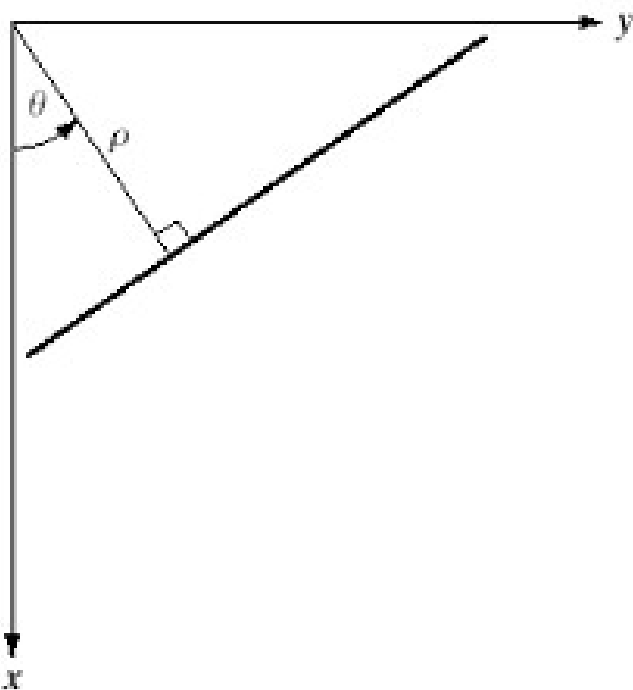
Calcul de la transformée de Hough

- On calcule le **gradient** de l'image originale
 - *Sobel, Prewitt, Canny, ...*
- Pour chaque point du gradient, on calcule une droite **(a,b)**
 - *On obtient une droite dans le plan a-b pour chaque pixel (x,y)*
- Les pics maximum dans le plan paramétrique a-b indiquent les droites avec le maximum de points du plan x-y
 - *Les points de croisement des droites dans le plan a-b indiquent les vraies droites existantes dans le plan x-y*

Problème avec un espace (a,b)

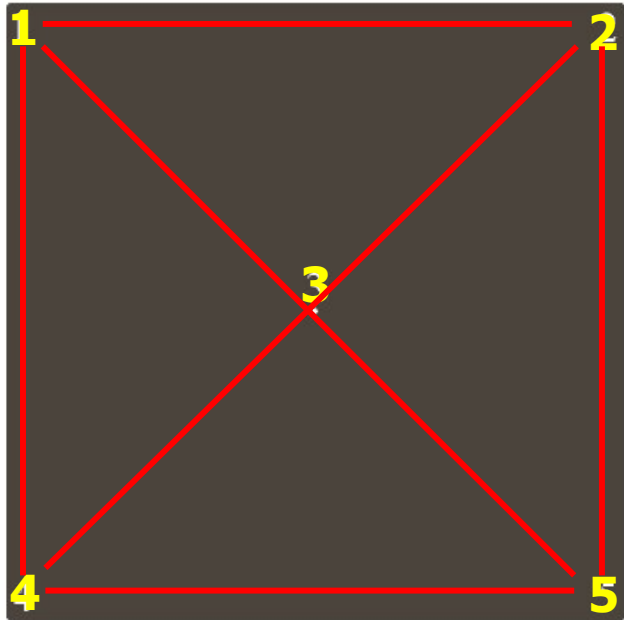
- **Problème** : pour une droite verticale, $b=\infty$!
- **Solution** : représentation sous forme polaire (ρ, θ)

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

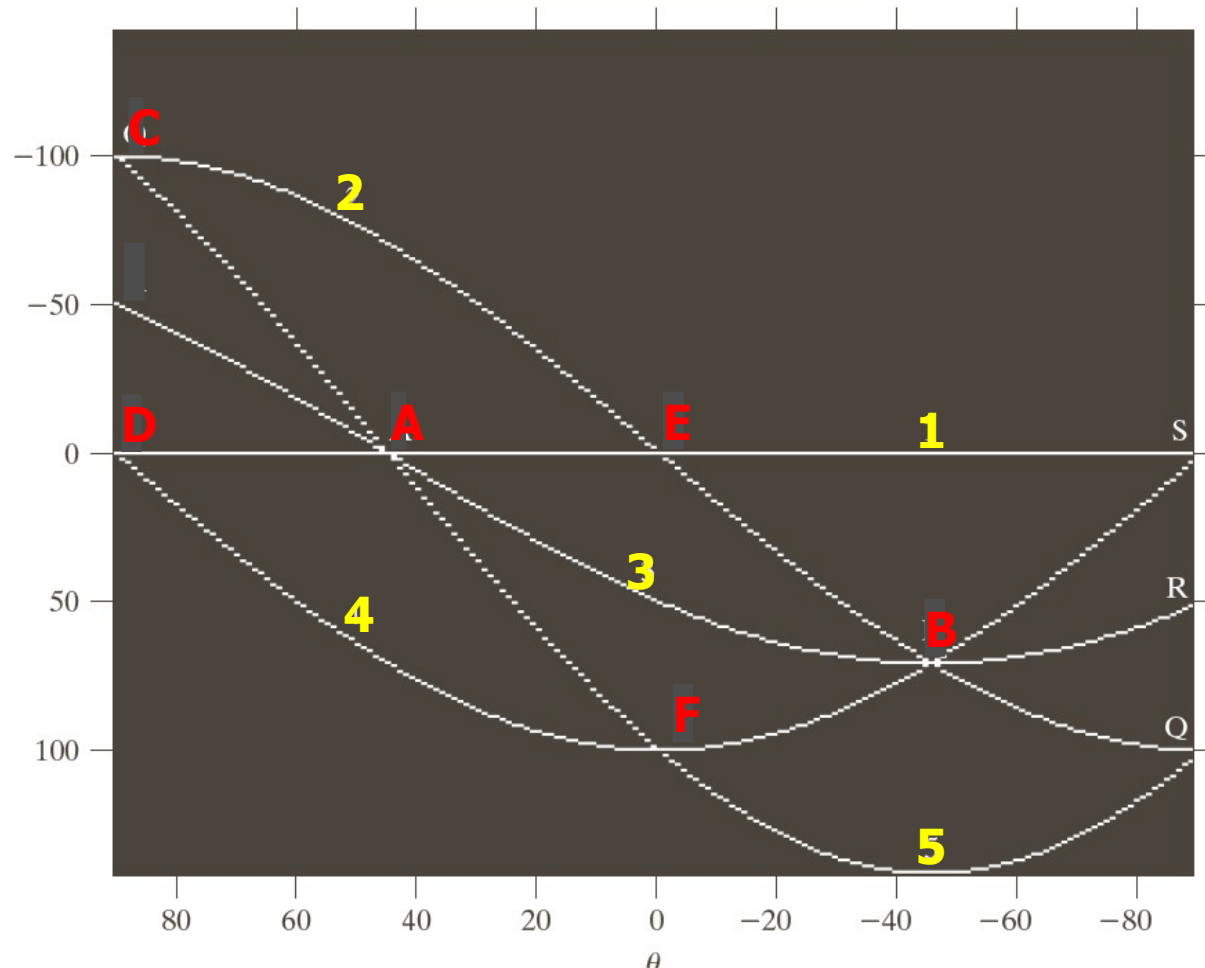


$$\theta = \pm 90$$

Exemple avec 5 points

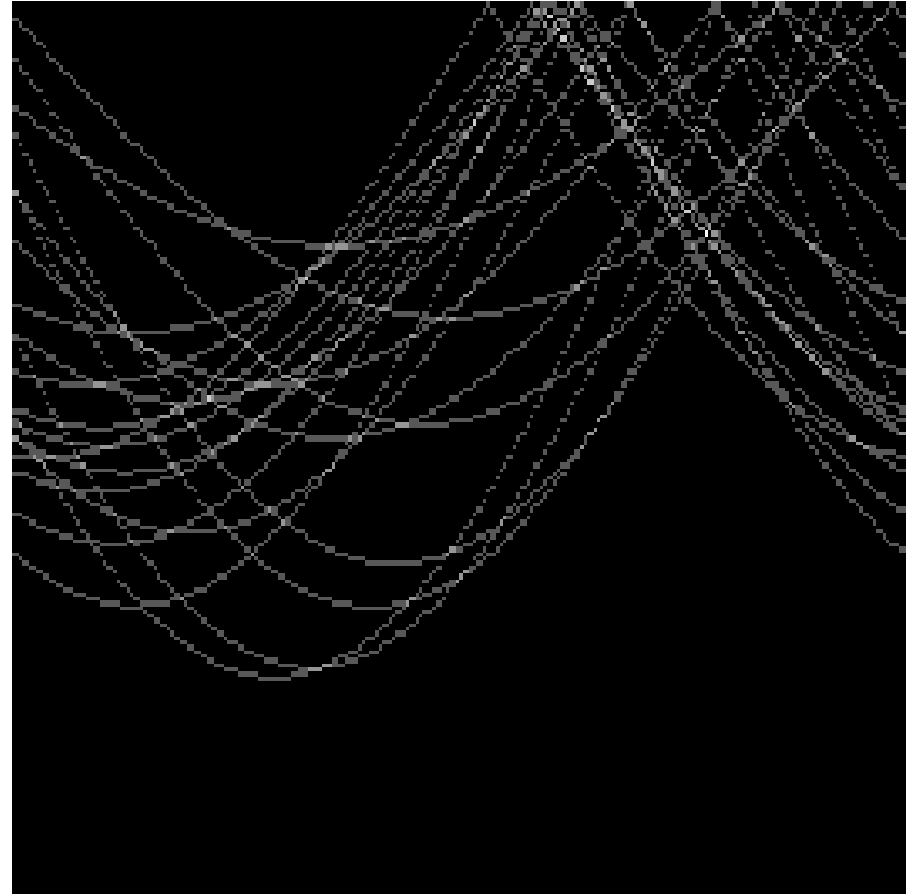
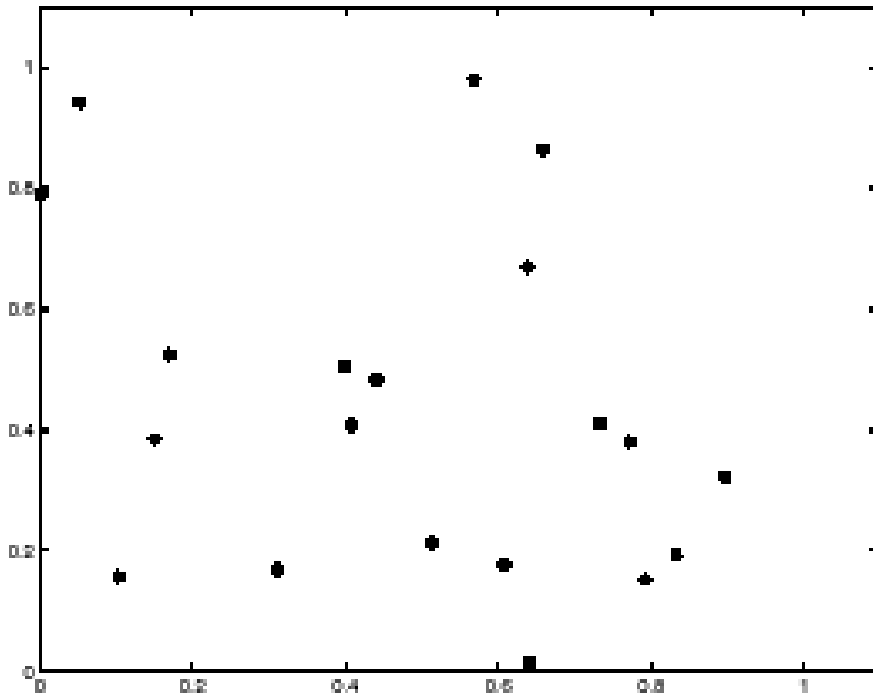


*Image contenant
5 points*



Transformée de Hough

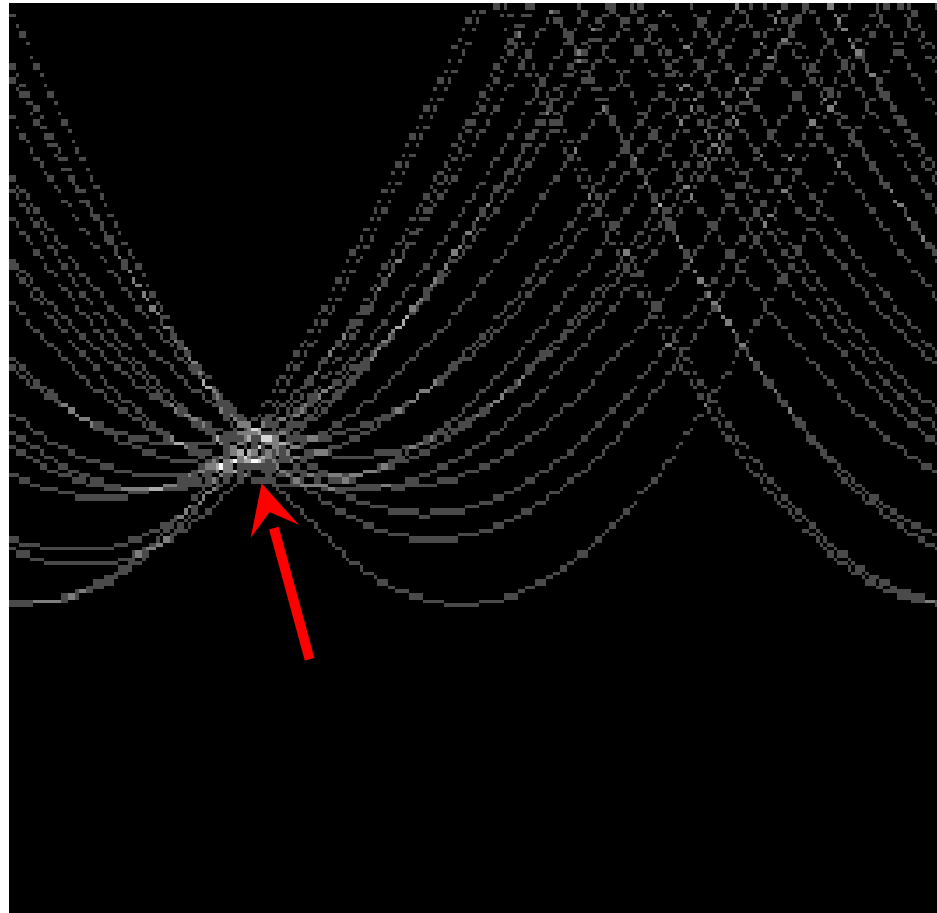
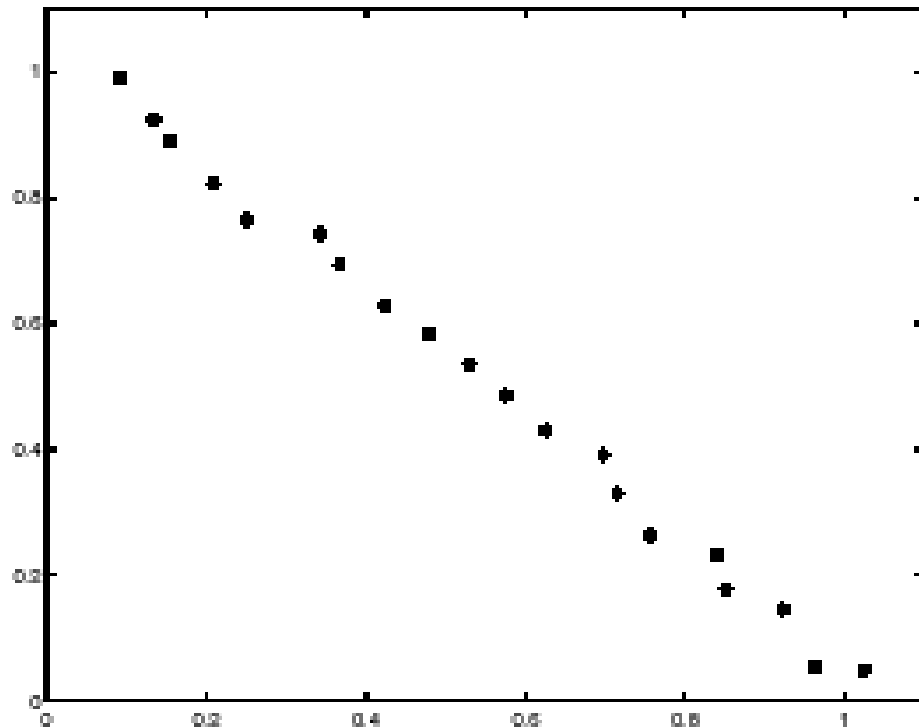
Transformée de Hough (points)



La transformée de points aléatoires ne donne rien de précis

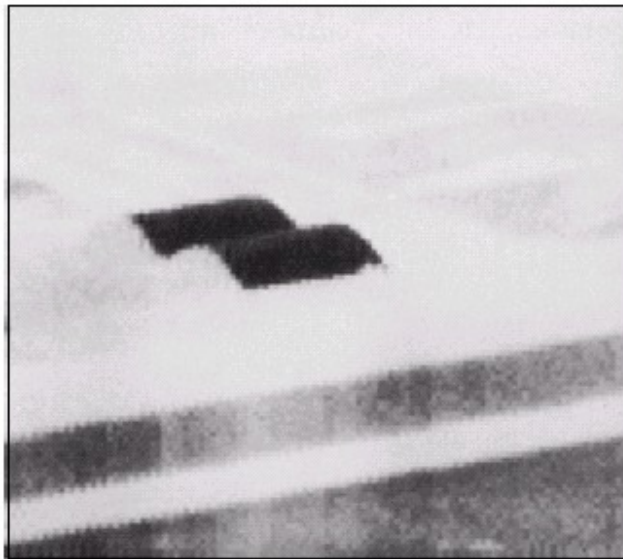
Transformée de Hough (droite)

La transformée de points alignés permet de retrouver la droite

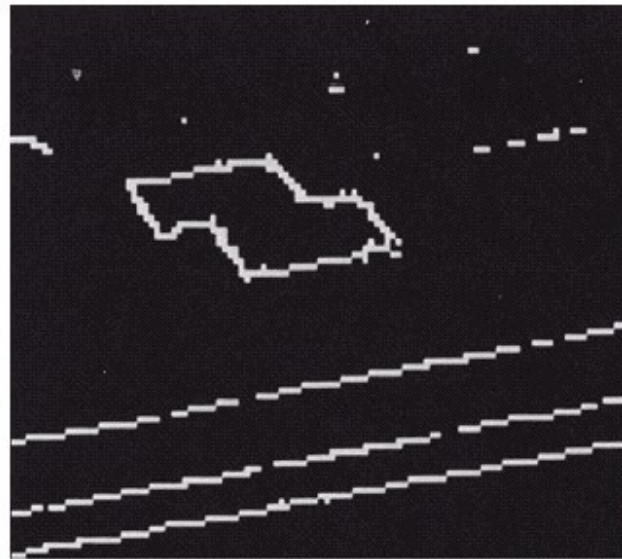


Example

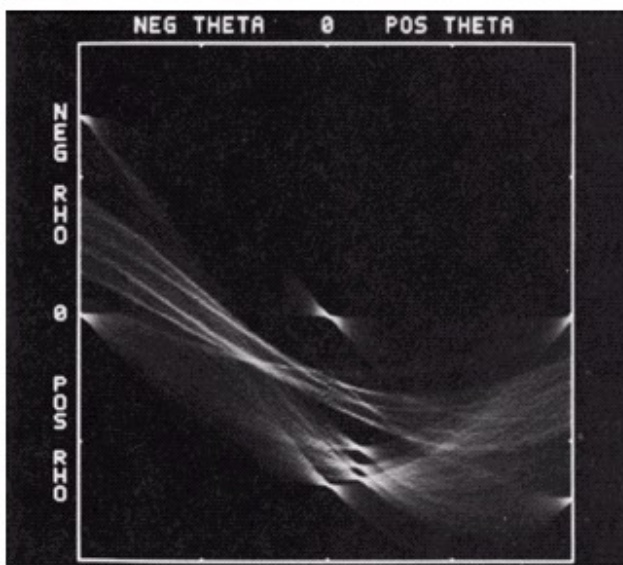
Image



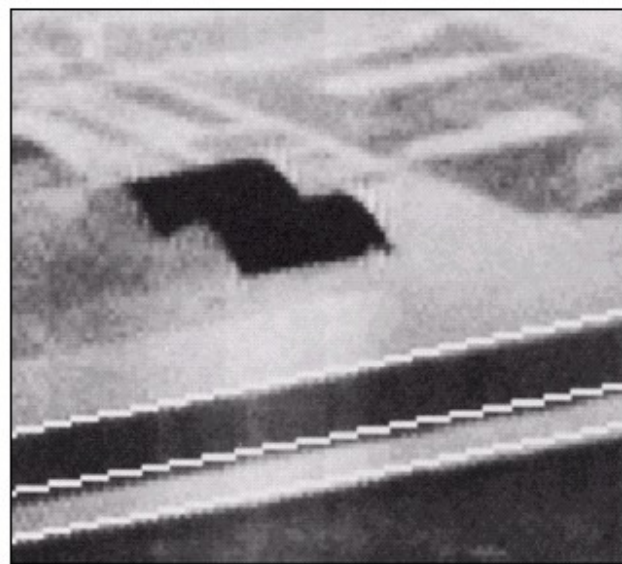
Gradient



Hough



Final

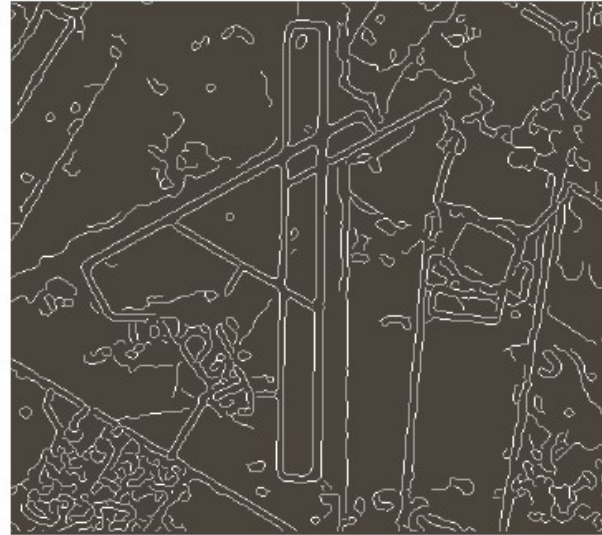


Autre exemple

Image



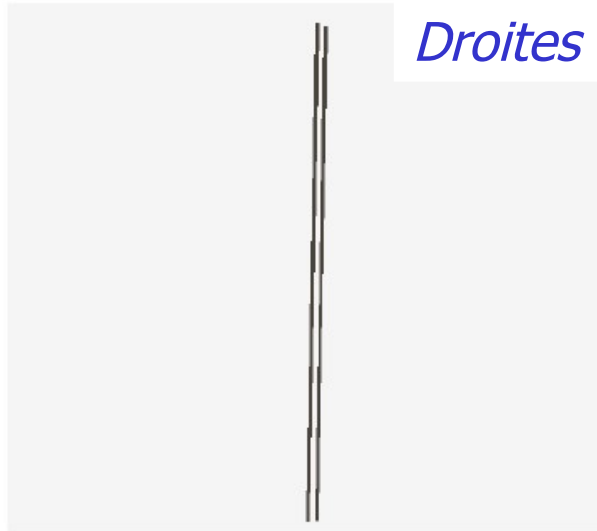
Canny



Hough



Droites



Final





Références

- Livres:
 - Introduction au Traitement d'Images (Lingrand), chapitre 6 et section 8.3
 - Digital Image Processing 2ed (Gonzalez & Woods), sections 3.7 et 10.2
- HIPR2 : Edge detectors (<http://homepages.inf.ed.ac.uk/rbf/HIPR2/featops.htm>) + Hough Transform (<http://homepages.inf.ed.ac.uk/rbf/HIPR2/hough.htm>)
- Quelques exemples de transformée de Hough:
<http://www.cs.iastate.edu/~baojie/acad/past/imageprocessing/hough.htm>
- Caroline Rougier. Cours de Traitement d'images (IFT2730). Université de Montréal (Canada) (liens ne plus disponibles)
 - <http://www-etud.iro.umontreal.ca/~rougierc/ift2730/>
Chap10 : Filtrage : lissage, réhaussement d'images, détection de contours