

**Spécialité**

**Automatique et Informatique Industrielle**

**Cours**

**IMAGERIE ET VISION INDUSTRIELLE**



**Introduction à la Détection de Contour**



# Introduction à la Détection de Contour

## Contenu de ce chapitre

- **Segmentation : Introduction à la Détection de Contour**
  - Objectifs
  - La segmentation
- **Approche contour**
  - Les différentes étapes de la détection de contours
- **Détection de Contour**
  - Gradient
  - Principes
- **Opérateurs du Gradient**
  - Opérateur de Roberts
- **Opérateurs du Gradient par masques**
  - Opérateurs de Prewitt et de Sobel
  - Opérateur gradient boussole
- **Filtrage de Canny**
  - Filtre Gaussien et opérateur LOG ( Laplacian of Gaussian)
  - Étapes d'une détection de contours par filtrage de Canny
  - Code Matlab

## Segmentation : Introduction à la Détection de Contour

### ● Objectifs

- Les traitements « bas-niveau » ont pour but d'extraire l'information utile et pertinente contenue dans l'image en regard de l'application considérée.
- Ces traitements opèrent sur des données de nature numérique et doivent donc simplifier l'image sans trop la dégrader.
- Le résultat d'un tel traitement donne une description structurée de l'image sans qu'il y ait de lien avec le contexte de la scène réelle.
- Le principal outil utilisé est la segmentation d'image.

### ● La segmentation

- La segmentation est un traitement de bas-niveau qui consiste à créer une partition de l'image  $I$  en un sous ensemble  $R_i$  appelés régions tels que :
  - $\forall i, R_i \neq \emptyset$
  - $\forall i, j (i \neq j), R_i \cap R_j = \emptyset$
  - $I = \cup_i R_i$
  - Les régions doivent correspondre à des objets ou des parties d'objets de la scène réelle.

#### ■ Deux approches duales :

Pour segmenter une image en régions ou objets bien qui y sont distincts contenus deux approches sont utilisées, l'une est basée sur l'information de contour ou bord qui délimite un objet, et l'autre utilise l'information région de même texture ou intérieur d'un objet. Pour aborder ce problème de segmentation d'une image on applique donc l'une des deux approches suivantes :

- ✓ L'approche contour (ou frontière)
- ✓ L'approche région

Pour détecter un objet dans une scène la recherche de caractéristiques dans une image est nécessaire. Plusieurs types de caractéristiques sont disponibles dans une image :

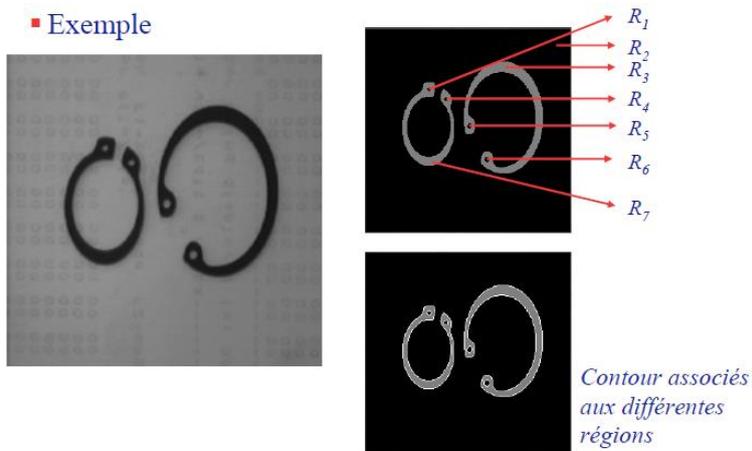
- Contours, squelette, régions,...

La détection de contour permet de garder que les informations significatives.

## Applications

La segmentation d'images est une étape primordiale dans un grand nombre d'applications en relation avec l'analyse d'image et la vision artificielle :

- reconnaissance de formes, d'objets,
- classification des scènes
- compression
- ----



## L'approche contour

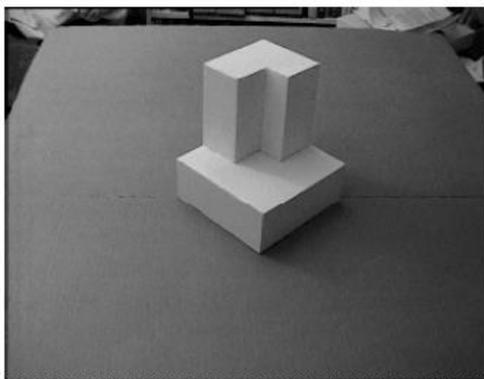
### • Définition

- Un contour (ou bord d'un objet) correspond à une variation d'intensité ou à une discontinuité entre les propriétés de deux ensembles de points.
- Un contour est une rupture d'intensité dans l'image suivant une direction donnée.
- Contour : frontière entre deux objets ou régions dans une image
- Toute discontinuité n'est pas nécessairement située à la frontière entre deux objets.

### Bords réels: Limites entre les objets

Un contour ou bord peut être observé dans une image suite à :

- Un changement brusque de luminosité peut également se produire dans l'objet
- Changements de réflectance
- Changement d'orientation de surface
- Modifications de l'éclairage. Par exemple. Limite d'ombre portée



- Les méthodes de détection de contours ne conduisent pas directement à une segmentation de l'image car les contours sont rarement connexes.
- Il faut souvent procéder à une étape de fermeture de contours.

- **Les différentes étapes de la détection de contours**

- Filtrage
- Calcul du module et de la direction du gradient
- Extraction des maximums locaux ou des passages par zéro.
- Seuillage simple ou seuillage par hystérésis
- Fermeture de contours et élimination du bruit
- Suivi et localisation des contours

- **Filtrage**

- Les contours représentent des hautes fréquences de l'image.
- Un filtrage passe-haut est donc utilisé
- L'opérateur de dérivation permet ce filtrage
- Deux approches :
- Estimation de la dérivée première
- Estimation de la dérivée seconde

## Détection de Contour

- La détection de contour est une technique de réduction d'information dans les images.
- Technique qui consiste à transformer l'image en un ensemble de courbes, pas forcément fermées, formant les frontières significatives de l'image.

Il existe plusieurs méthodes de détection de contours.

Deux approches les plus utilisées :

- **Approximations du gradient (dérivée première directionnelle)**
- **Approximations du Laplacien (dérivées secondes)**

## Gradient

- Calculer d'abord le gradient de l'image dans les deux directions orthogonales ( $x$  et  $y$ )
- Calculer le module du gradient
- Il s'agira ensuite d'effectuer une sélection de contours
- Il s'agira ensuite d'effectuer une sélection de contours les plus marqués et pertinents (les point de plus fort contraste par un seuillage).

Les directions des contours étant orthogonales à la direction du gradient déterminée en chaque pixel de l'image.

## Principes

En considérant l'image dans un repère orthogonal ( $Oxy$ ) tel que ( $Ox$ ) désigne l'axe horizontal et ( $Oy$ ) l'axe vertical, le **Gradient  $G$  de l'image** (ou plutôt de la **luminance**  $I(x, y)$ ) en tout point ou pixel de coordonnées  $(x, y)$  est défini par :

$$\mathbf{G} = \nabla I(x, y) = \left( \frac{\partial I(x, y)}{\partial x}, \frac{\partial I(x, y)}{\partial y} \right)^T$$

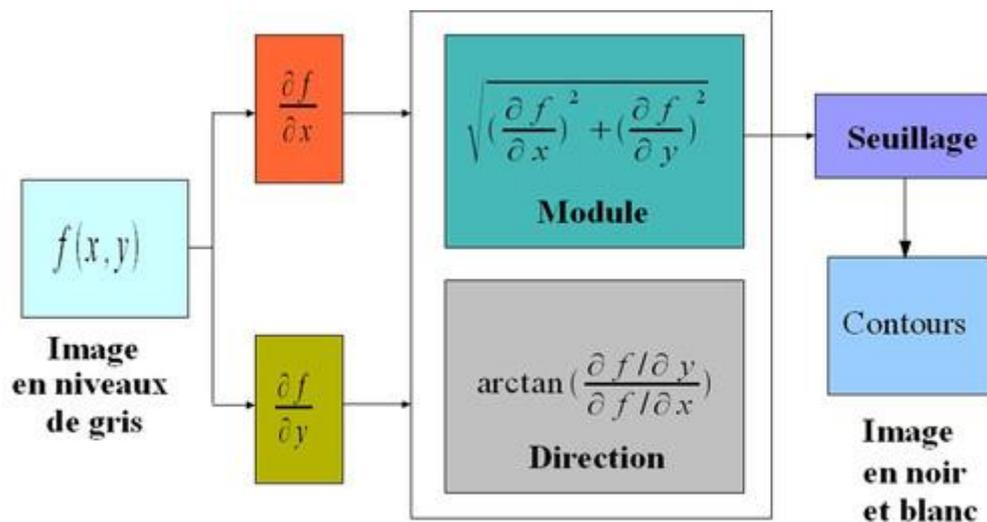
Le **module du gradient** permet de quantifier l'importance du contour mis en évidence, c'est-à-dire l'amplitude du saut d'intensité relevé dans l'image :

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

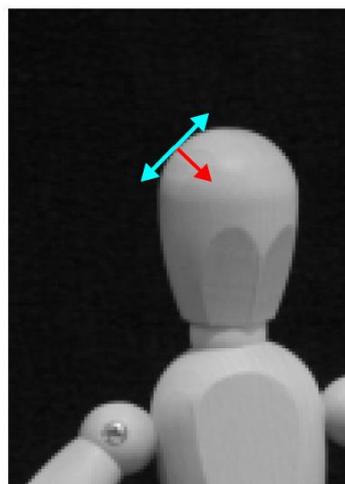
La **direction du gradient** permet de déterminer l'arête présente dans l'image. En effet, la direction du gradient est orthogonale à celle du contour :

$$\theta = \text{arctg} \left( \frac{\partial I(x, y)}{\partial y} / \frac{\partial I(x, y)}{\partial x} \right)$$

Le principe de la **détection de contours par l'utilisation du gradient** consiste à calculer d'abord le gradient de l'image dans deux directions orthogonales puis le module du gradient. Il s'agira ensuite d'effectuer une sélection des contours les plus marqués, c'est-à-dire les points de plus fort contraste par un seuillage adéquat, les directions des contours étant orthogonales à la direction  $\theta_0$  déterminée en tout pixel de l'image (voir ci-dessous).

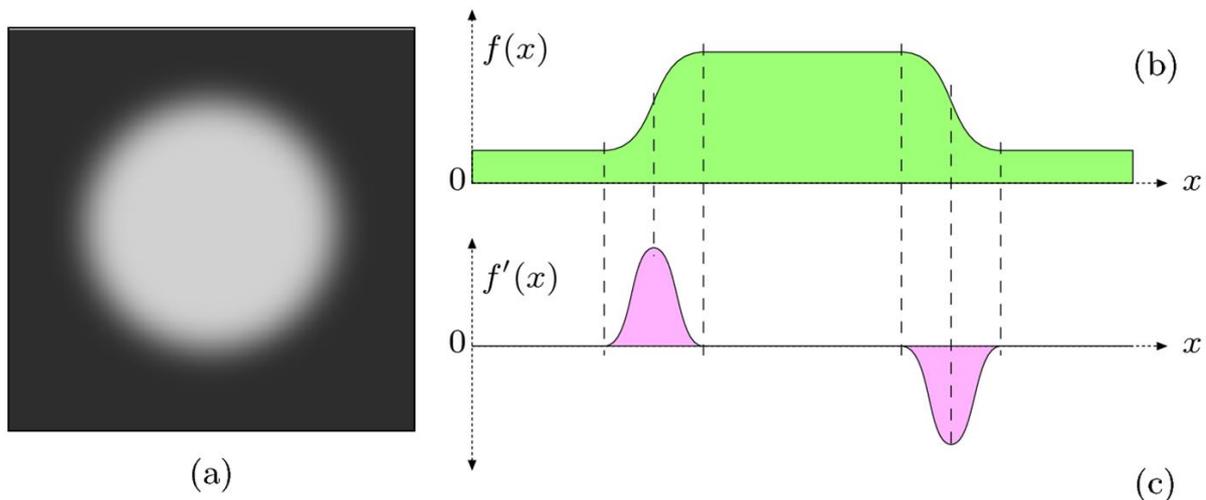


- Le module est invariant à une rotation de l'image, utilisé en détection de contour.



Les contours sont caractérisés par des valeurs assez élevées de la dérivée première :

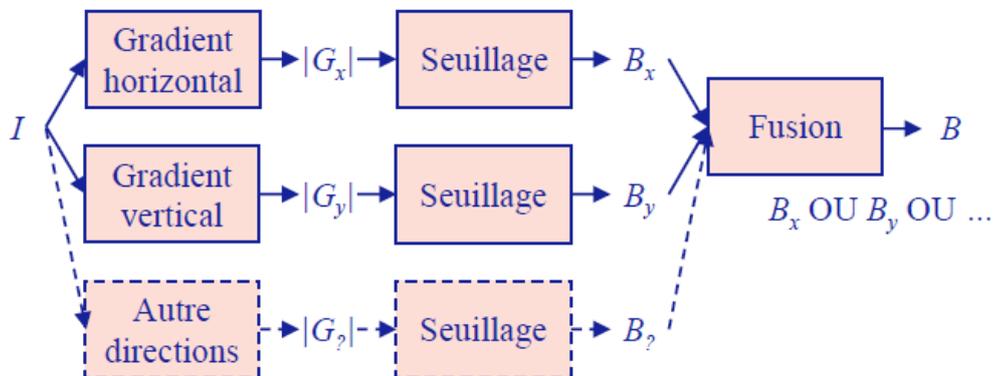
$$f'(x) = \frac{df}{dx}(x)$$



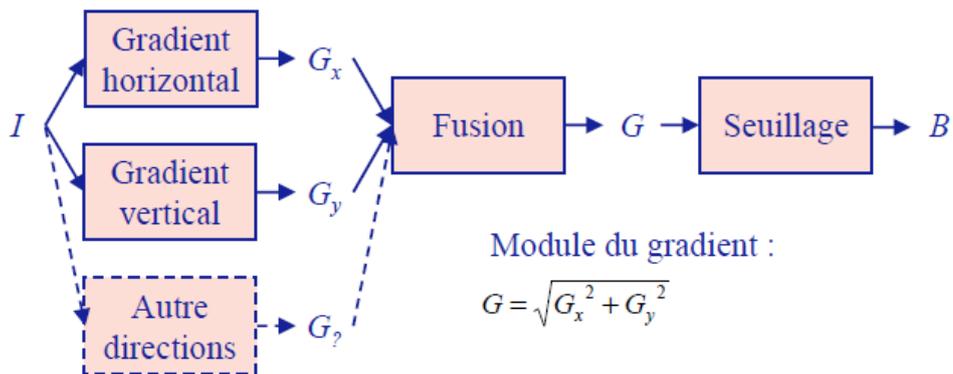
## Opérateurs du Gradient

- Ces opérateurs sont considérés comme des filtres spatiaux qui vont être corrélés à l'image.
- Les réponses impulsionnelles de ces filtres peuvent se présenter sous la forme de fonctions analytiques ou bien sous la forme de masques bidimensionnels
- Dans les deux cas le filtrage a lieu en deux étapes :
  - Filtrage suivant les lignes de l'image
  - Filtrage suivant les colonnes de l'image

- **Module et direction du gradient**
  - ❖ Cas d'une image binaire



- Cas d'une image à niveau de gris



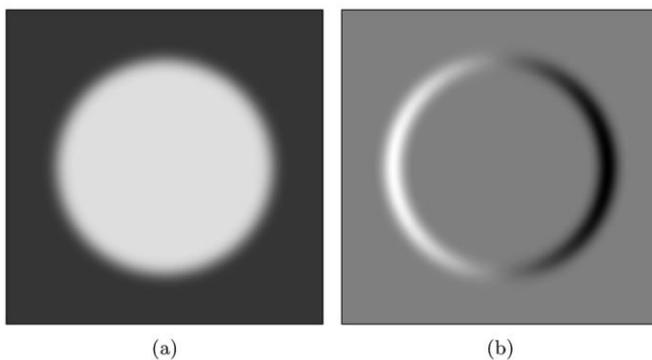
Module du gradient :

$$G = \sqrt{G_x^2 + G_y^2}$$

Direction du gradient :

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

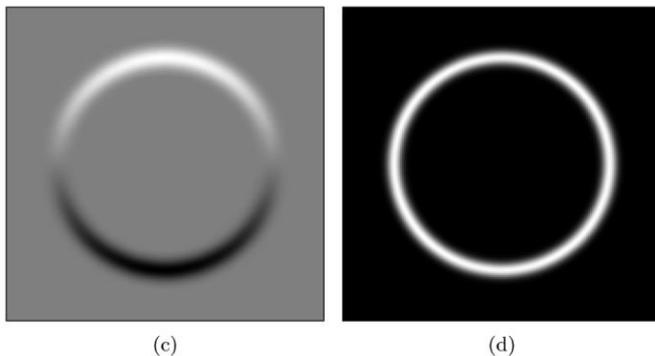
### Gradient dans la direction horizontale



(a) Image initiale

(b) Gradient horizontal :  $\frac{\partial I}{\partial x}(x, y)$

**Gradient dans la direction verticale et résultat de la fusion des gradients avec seuillage :**



(c) Gradient vertical :  $\frac{\partial I}{\partial y}(x, y)$

(d) Fusion et seuillage des gradients

**❖ Opérateurs du Gradient par masques**

**Opérateurs par masques**

- Pour chaque opérateur deux masques sont utilisés de façon à déterminer le gradient de l'image dans deux directions orthogonales.
- Approximation de base  
On utilise la première dérivée de l'image selon  $x$  et  $y$  pour détecter les contours :

$$\frac{\Delta f}{\Delta x} = \frac{f(x+\Delta x) - f(x)}{\Delta x} \quad \text{et} \quad \frac{\Delta f}{\Delta y} = \frac{f(y+\Delta y) - f(y)}{\Delta y}$$

- Le masque le plus intuitif à mettre en œuvre pour une approximation de la première dérivée est un masque à deux éléments :

pour le masque horizontal on a :

-1	+1
----	----

-1
+1

-1	+1
----	----

■  $h(m, n)$  avec  $h(0, 0) = -1$  et  $h(1, 0) = +1$

$$\sum_{m=0}^1 \sum_{n=0}^0 h(m, n) f(m+i, n+j) = h(0, 0)f(i, j) + h(1, 0)f(i+1, j)$$

$$= f(i+1, j) - 1 \times f(i, j)$$

pour le masque vertical on a :

-1
+1

■  $h(m,n)$  avec  $h(0,0) = 1$  et  $h(0,+1) = -1$

$$\sum_{m=0}^1 \sum_{n=0}^0 h(m,n) f(m+i, n+j) = h(0,0)f(i,j) + h(0,-1)f(i,j-1)$$

$$= f(i,j) - 1 \times f(i,j-1)$$

• **Opérateur de Roberts**

Ce masque proposé en 1965 permet de calculer un gradient le long des diagonales de l'image :

L'origine des pixels de cet opérateur est le point du haut à droite :

1	0
0	-1

Masque de Roberts

la réponse impulsionnelle s'écrit :

$$h(-1,0) = 1$$

$$\text{et } h(0,-1) = -1$$

Pour la détection de l'autre diagonale on utilise l'opérateur suivant dont la réponse impulsionnelle s'écrit :

0	1
-1	0

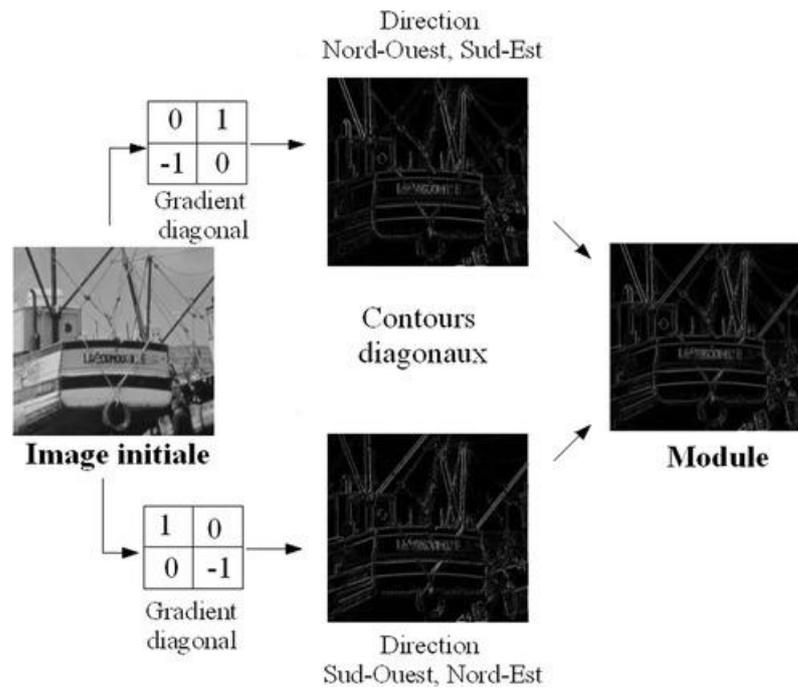
Masque de Roberts

$$h(0,0) = 1$$

$$\text{et } h(-1,-1) = -1$$

Ce deuxième masque se déduit du premier par rotation de  $\frac{\pi}{2}$

- Problématique de ces opérateurs : - Forte sensibilité aux bruits
- D'autres masques ont été proposés afin de rendre la détection de contour via le gradient moins sensible au bruit.



• **Opérateurs de Prewitt et de Sobel**

- Moins sensibles au bruit que le calcul direct de la dérivée
- Le calcul du gradient est toujours obtenu par l'intermédiaire de deux masques
  - Le premier effectuant une approximation du gradient horizontal
  - Le deuxième effectuant une approximation du gradient vertical

-1	-C	-1
0	0	0
1	C	1

Opérateur horizontal

-1	0	1
-C	0	C
-1	0	1

Opérateur vertical

- $C = 1$  : il s'agit de l'opérateur de Prewitt
- $C = 2$  : il s'agit de l'opérateur de Sobel
- Chaque pixel des masques est normalisé par :  $\frac{1}{C+2}$
- L'origine de ces masques est toujours le pixel central

### Masques de Prewitt

$$\frac{1}{3} \times \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Opérateur horizontal

$$\frac{1}{3} \times \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$$

Opérateur vertical

### Masques de Sobel

$$\frac{1}{4} \times \begin{array}{|c|c|c|} \hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Opérateur horizontal

$$\frac{1}{4} \times \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$$

Opérateur vertical

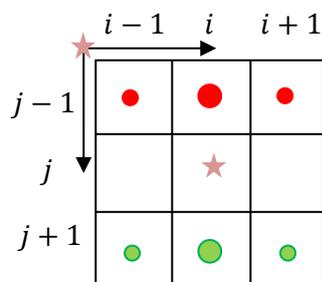
Ces masques ont l'avantage de produire deux effets :

- Ils effectuent le calcul du gradient dans une direction.
- et un lissage de l'image dans la direction orthogonale.

Ce lissage rend ces masques un peu moins sensibles au bruit que les précédents.

Pour les contours horizontaux :

Ces opérateurs ont la structure et la réponse impulsionnelle suivantes



Opérateur horizontal

★ : origine (0,0)

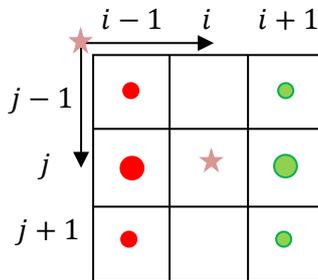
$$h(i, j): \quad h(-1, 1) = h(1, 1) = 1$$

$$h(-1, -1) = h(1, -1) = -1$$

$$h(0, 1) = C \text{ et } h(0, -1) = -C$$

**Pour les contours verticaux :**

Ces opérateurs ont la structure et la réponse impulsionnelle suivantes



★ : origine (0,0)

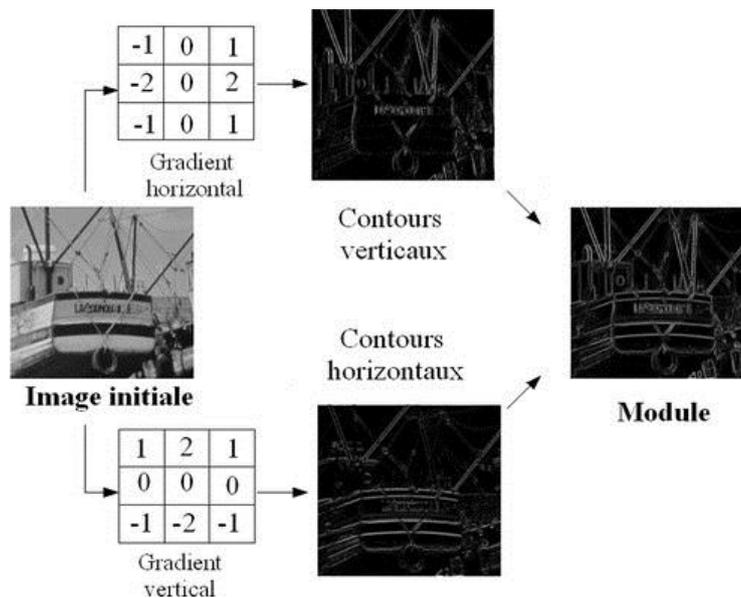
$h(i,j): \quad h(1,-1) = h(1,1) = 1$

$h(-1,1) = h(-1,-1) = -1$

$h(1,0) = C \text{ et } h(-1,0) = -C$

Opérateur vertical

Le principal avantage de ces masques est leur facilité de mise en œuvre ainsi que la rapidité de leur traitement sur ordinateur. Cependant leur inconvénient reste encore leur sensibilité au bruit. De plus les contours obtenus sont souvent assez larges. L'opérateur de Sobel est très utilisé en vision industrielle nécessitant des contraintes temps-réel.



Exemple d'application de l'opérateur de Sobel

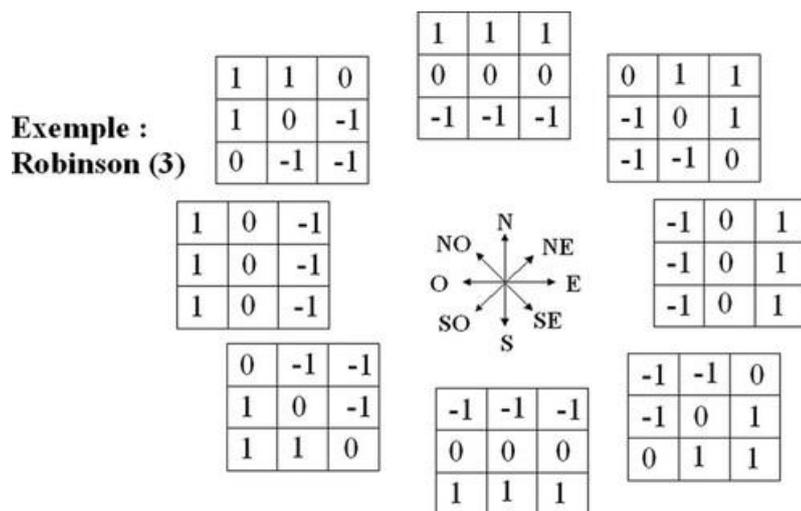
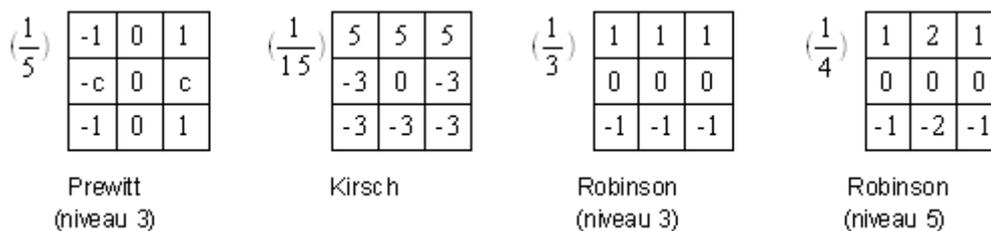
Il existe bien sûr beaucoup d'autres masques utilisés pour déterminer le gradient d'une image. Le principal intérêt de ces masques est leur facilité de mise en œuvre ainsi que la rapidité de leur traitement. Leur inconvénient est leur grande sensibilité au bruit. De plus les contours obtenus sont souvent assez larges. Le filtre de Sobel est le plus utilisé dans les applications industrielles nécessitant des contraintes temps-réel.

• **Opérateur gradient boussole**

Les opérateurs dits boussole mesurent le gradient dans des directions sélectionnées. L'image est successivement filtrée par un ensemble de masques  $h_k(i, j), k = 0, 1, \dots, 7$ , dont chacun représente une approximation discrète d'un contour idéal dans une orientation spécifique. Le résultat du filtrage de l'image  $f(i, j)$  avec le *k*<sup>ième</sup> masque est  $g_k(i, j)$ .

Il s'agit alors de garder les contours correspondant à l'orientation du masque ayant conduit au maximum des fonctions  $g_k(i, j)$  avec  $k$  allant de 0 à 7, représentatif des huit principales directions d'une boussole.

Plusieurs masques peuvent être utilisés. La démarche consiste à choisir un type de masque puis à effectuer des permutations circulaires dans les huit directions possibles du gradient. Des exemples d'opérateurs gradient boussole dans la direction Nord sont présentés ci-dessous en recourant aux masques de Prewitt, de Kirsch, de Robinson de niveau 3 ou 5. Le terme de niveau désigne le nombre de valeurs différentes présentes dans le masque.



Exemple d'opérateurs Gradient boussole avec le masque de Robinson de niveau 3

## En résumé : liste des principaux opérateurs gradient

- Lissage des approximations de la dérivée première

$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix}$
<i>Prewitt vertical</i>	<i>Sobel vertical</i>	<i>Roberts vertical</i>	<i>Kirsch vertical</i>
$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}$
<i>Prewitt horizontal</i>	<i>Sobel horizontal</i>	<i>Roberts horizontal</i>	<i>Kirsch horizontal</i>

- Autres directions

$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$
<i>Sobel 0°</i>	<i>Sobel 90°</i>	<i>Sobel 45°</i>	<i>Sobel 135°</i>

## • Filtrage de Canny

### ➤ Dérivation par filtrage optimal

Les dérivations présentées consistent à convoluer l'image par des masques de petites dimensions. Ces approches sont donc dépendantes de la taille des objets traités, elles sont aussi très sensibles au bruit. Un autre type d'approche plus récente repose sur la définition de critères d'optimalité de la détection de contours; ces critères débouchant sur des filtres de lissage *optimaux*.

- Soit  $h$  le filtre de lissage alors :
  - $I(x, y) * h(x) * h(y)$  est l'image lissée,
  - $I(x, y) * h'(x) * h(y)$  et  $I(x, y) * h(x) * h'(y)$  sont les images des dérivées suivant  $x$  et  $y$
  - $I(x, y) * (h''(x) * h(y) + h(x) * h''(y))$  est l'image du Laplacien
- L'implantation se fait de manière récursive (i.e., la valeur de sortie du filtre en un point est déterminée en fonction de celles de ses voisins) par deux balayages de l'image (propriété des filtres séparables)

Pour le calcul du filtre de Canny, une approche analytique plus élaborée est employée. Il s'agit d'une technique de filtrage optimal. Canny a en effet cherché à déterminer de façon analytique en 1986 un filtre à partir de trois critères :

- un critère de bonne détection garantissant une réponse forte en sortie du filtre même en présence de faibles contours sur l'image d'entrée,
- un critère de bonne localisation du contour,
- un critère d'unicité de la réponse permettant d'assurer une seule détection pour un contour et ainsi d'éviter les effets de rebond.

Canny définit ces trois critères de façon mathématique. L'optimisation des trois critères proposés permet de définir le filtre linéaire optimal pour la détection d'une marche d'escalier sous l'hypothèse d'un bruit additif indépendant du signal. Il s'agit de trouver la réponse impulsionnelle  $h(x)$  du filtre optimal qui permet d'obtenir une valeur maximum en sortie lorsqu'un contour est présenté en entrée. Canny propose un filtre à réponse impulsionnelle finie.

## • Principe

On se place dans le cas monodimensionnel. On suppose que la détection est effectuée en convoluant le signal par un filtre de réponse impulsionnel  $f_i$ , les contours étant caractérisé par les extrema de la sortie du filtre. Les contours envisagés ici sont des contours de types marche et le bruit est supposé blanc (de moyenne nulle). Les critères d'optimalités sont :

- détection : le contour doit être détecté, il faut minimiser les fausses réponses,
- localisation : le contour doit être localisé avec précision, il s'agit de minimiser

la distance entre les points détectés et le vrai contour.

- réponse unique : il s'agit de minimiser le nombre de réponse pour un seul contour.

- ❖ La minimisation des ces critères dans le contexte du modèle proposé débouche sur une équation différentielle dont la solution est de la forme :

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

### ▪ *Interprétation*

La détection de contour par calcul du gradient est sensible au bruit. Une idée avancée permettant d'atténuer ce problème consiste à filtrer passe-bas l'image avant d'appliquer l'opérateur Gradient. Mathématiquement cela revient à convoluer l'image initiale inchangée  $I(x, y)$  avec la dérivée de la réponse impulsionnelle  $h(x, y)$  du filtre passe-bas comme le montre l'écriture suivante dans le cas monodimensionnel : Il reste à déterminer le filtre passe-bas pour  $h(x, y)$ . Dans le cas d'un filtre gaussien, l'application du gradient consiste finalement à filtrer l'image initiale par une dérivée de Gaussienne  $h_G(x)$ . C'est en fait à peu de chose près l'approche du filtrage de Canny.

## ✚ Filtre Gaussien et opérateur LOG ( Laplacian of Gaussian)

Le filtre de lissage Gaussien a pour réponse impulsionnelle :

$$h(x) = c e^{-x^2/2\sigma^2}$$

et donc pour dérivée :

$$h'(x) = -c \frac{x}{\sigma^2} e^{-x^2/2\sigma^2}$$

Canny a montré que la dérivée de ce filtre reconstitue une approximation de la solution du filtre optimal. Ce filtre a été initialement introduit par Marr et Hildreth dans le cadre du calcul du laplacien (Laplacian of Gaussian LoG).

$$h(x, y) = c e^{-(x^2+y^2)/2\sigma^2}$$

en posant  $r^2 = x^2 + y^2$  on obtient

$$h(r) = c e^{-r^2/2\sigma^2}$$

soit pour dérivée :

$$h'(r) = -c \frac{r}{\sigma^2} e^{-r^2/2\sigma^2}$$

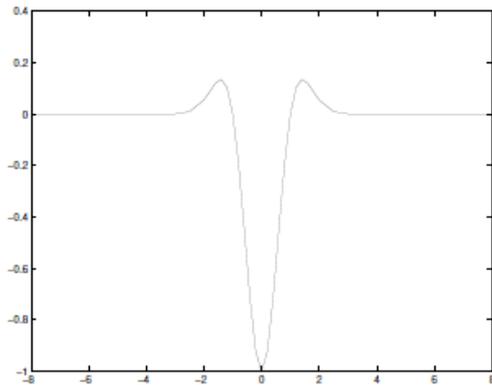
et pour dérivée seconde

$$h''(r) = c \left( \frac{r^2}{\sigma^2} - 1 \right) e^{-r^2/2\sigma^2}$$

Soit :

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

Ici  $c$  est une constante qui normalise la somme des éléments du filtre à zéro



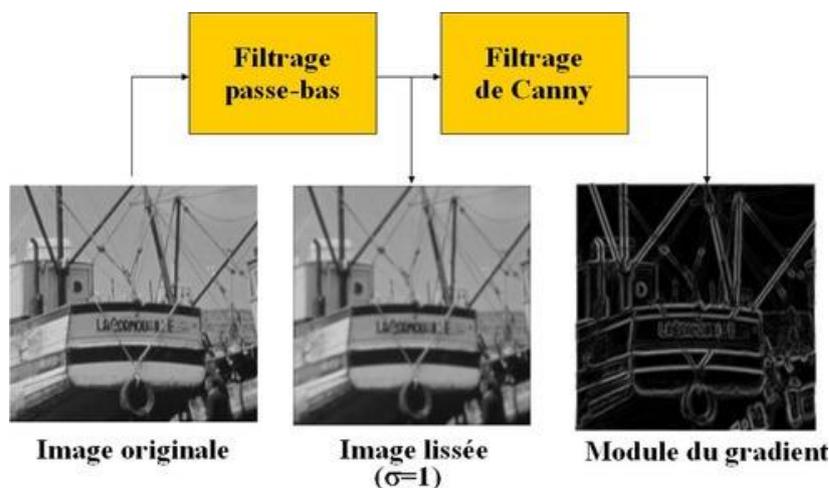
la fonction  $(\frac{r^2}{\sigma^2} - 1) e^{-r^2/2\sigma^2}$ .

L'opérateur LoG est non directionnel (isotrope).

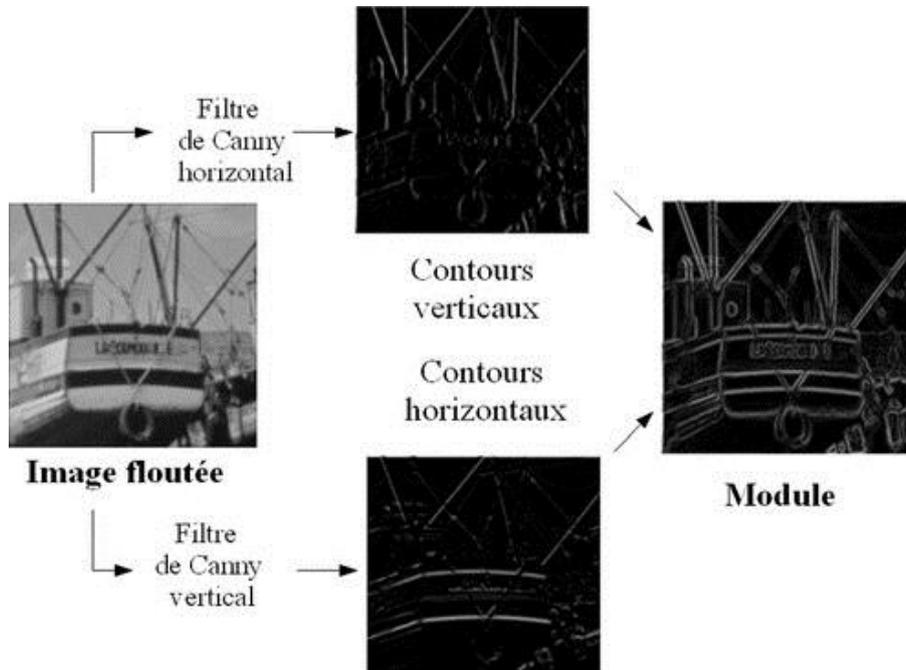
- **Étapes d'une détection de contours par filtrage de Canny**

L'objectif est bien de calculer le module du gradient de l'image analysée. Souvent avant d'appliquer le filtre de Canny, un filtrage préalable est opéré sur l'image au moyen d'un filtre gaussien. Les différentes étapes sont énumérées ci-après :

- convolution de l'image initiale avec un filtre passe-bas gaussien bidimensionnel (ou convolution 1D dans chacune des deux directions),
- convolution de l'image lissée avec le filtre de Canny ou la dérivée de gaussienne dans les directions horizontale et verticale,
- calcul du module du gradient à partir des deux images représentant les gradients de l'image filtrée passe-bas dans les directions horizontale et verticale.



Exemple d'application du filtrage de Canny



Obtention du module du gradient par filtrage de Canny

Filtre de Canny appliqué à une image photo



Source : [https://en.wikipedia.org/wiki/Edge\\_detection](https://en.wikipedia.org/wiki/Edge_detection)

### ❖ Code Matlab pour l'implémentation des différents opérateurs de détection de contours :

```

% Ce script Matlab permet de trouver les contours dans une image
% la fonction edge est utilisée avec différents opérateurs de détection de
contour.
%

close all; clear all;

% read an image
I = imread('cameraman.tif');
imshow(I);

% Détecter les contours dans l'image
BW = edge(I);

% visualiser le résultat
figure;imshow(BW)
%
% Cette fonction Matlab reçoit en entrée une image I binaire ou à niveaux
% de gris , et produit en sortie une image de même taille avec un '1' pour
% un point de contour et '0' autrement.
%
% syntaxe de la fonction edge(I) ou edge(I,'type d'opérateur',paramètres,
options...)
% BW = edge(I)
%
%     BW = edge(I,'sobel');
%     BW = edge(I,'sobel',thresh);
%         seuillage : thresh veut dire valeur du seuil choisi
%     BW = edge(I,'sobel',thresh,direction);
%     BW = edge(I,'sobel',...,options);
%     [BW,thresh] = edge(I,'sobel',...);
%     BW = edge(I,'prewitt');
%     BW = edge(I,'prewitt',thresh);
%     BW = edge(I,'prewitt',thresh,direction);
%     [BW,thresh] = edge(I,'prewitt',...);
%     BW = edge(I,'roberts');
%     BW = edge(I,'roberts',thresh);
%     BW = edge(I,'roberts',...,options);
%     [BW,thresh] = edge(I,'roberts',...);
%     BW = edge(I,'log');
%     BW = edge(I,'log',thresh);
%     BW = edge(I,'log',thresh,sigma);
%     [BW,thresh] = edge(I,'log',...);
%     BW = edge(I,'zerocross',thresh,h);
%     [BW,thresh] = edge(I,'zerocross',...);
%     BW = edge(I,'canny');
%     BW = edge(I,'canny',thresh);
%     BW = edge(I,'canny',thresh,sigma);
%     [BW,thresh] = edge(I,'canny',...);

```