Chapitre 4 :

Filtres : Détection de Contours

C’est quoi une bordure (contour) ?

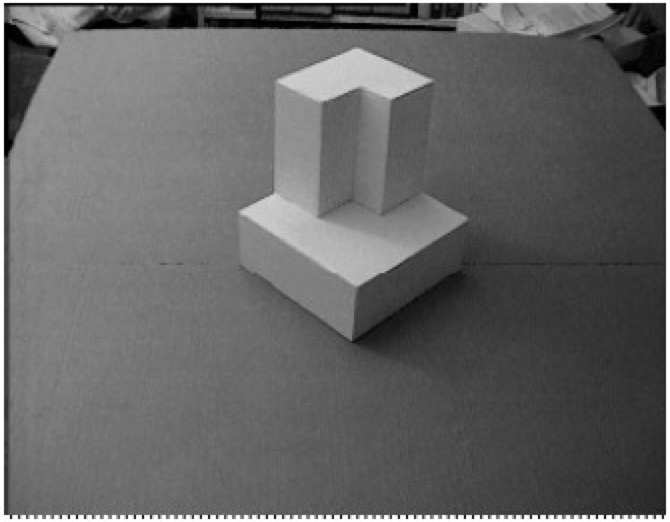
- **Un contour dans une image est une discontinuité (changement brusque) de l’intensité.**

- **Les contours représentent les frontières entre les objets dans l’image.**

- **Ils peuvent aussi apparaître dans le même objet du faite de :**

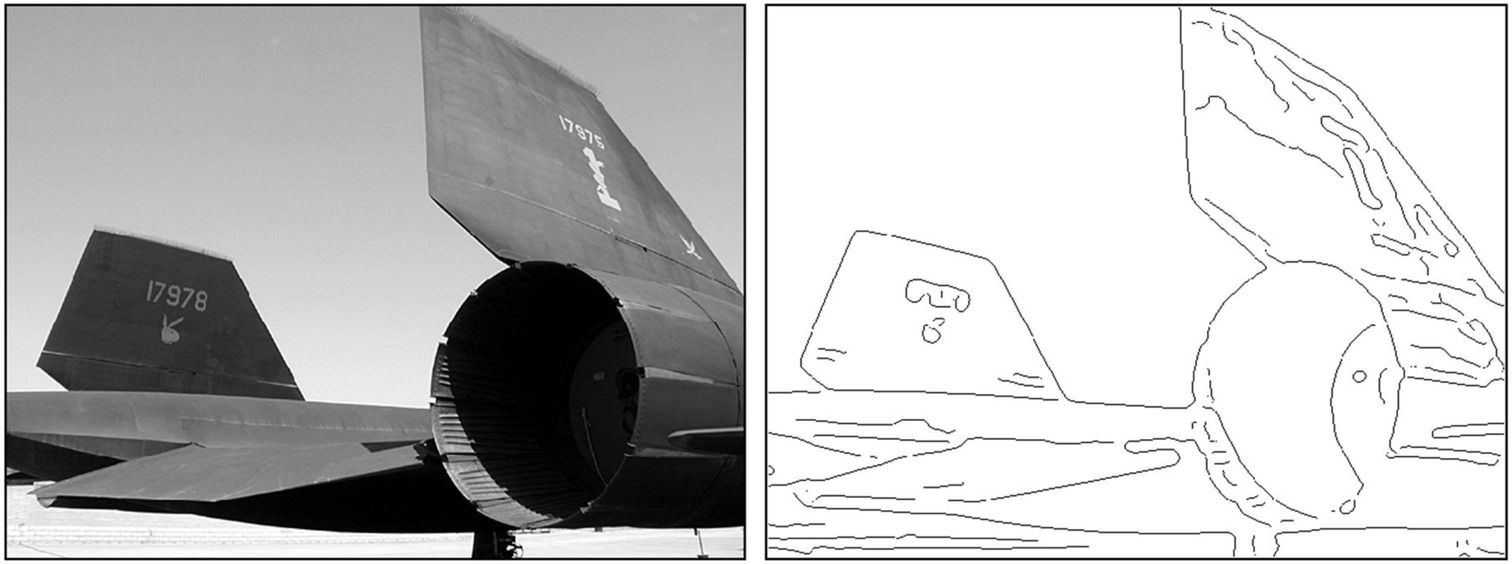
  - **Changement dans l’orientation d’une surface**
  - **Changement du luminance**
  - **Reflet des objets ..**

C’est quoi une bordure (contour) ?



Détection de Contour

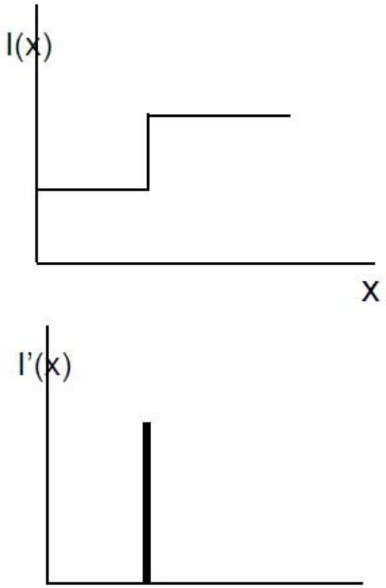
La détection de contour est un traitement d’image permettant de trouver les bordure dans une image.

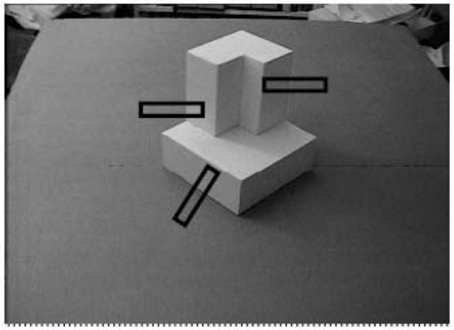


(a) (b)

Caractéristiques d'un contour

- Contour : un changement brusque de l'intensité.
- Un contour idéal est une fonction échelonnée (step function) dans une direction.

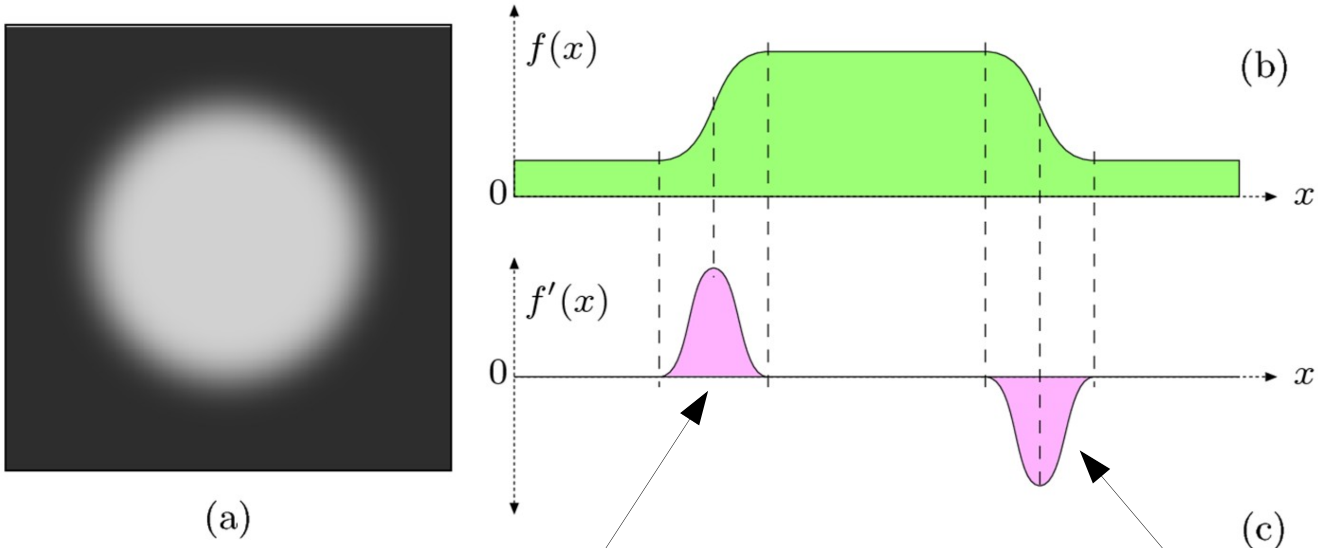




Caractéristiques d’un contour

Les contours peuvent être caractérisés par une grande valeur pour la première dérivée.

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



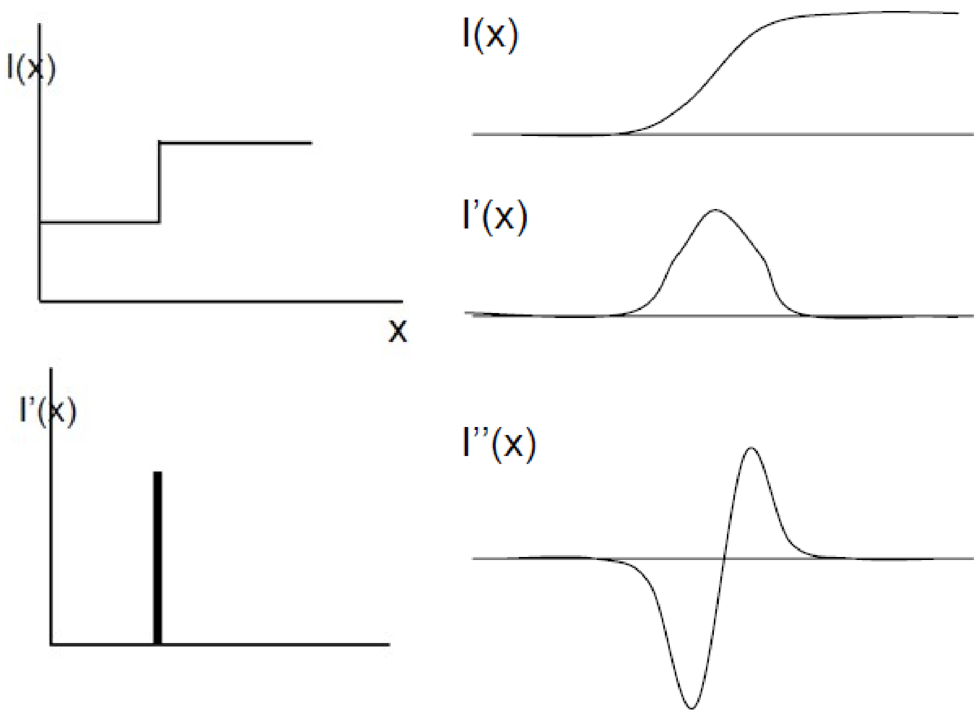
Une pente montante induit une grande valeur positive de la première dérivée

Une pente descendante induit une grande valeur négative de la première dérivée

Caractéristiques d’un contour

- Un contour idéal est une fonction échelonnée (step function) dans une direction.
- La première dérivée de I(x) a une pointe (peak) au contour.
- La deuxième dérivée de I(x) a un passage par zéro au contour.

Bordure idéale



Bordure réelle

Première dérivée
Contenant une pointe

Deuxième dérivée
Contenant un passage par 0

Caractéristiques d'un contour

$$\frac{\partial f}{\partial x} = \lim_{\varepsilon \to \infty} \frac{f(x+\varepsilon, y) - f(x, y)}{\varepsilon} \quad \Rightarrow \quad \frac{\partial f}{\partial x} \approx f_{i+1,j} - f_{i,j}$$

Équivalent à une convolution par un masque 1D

0	0	0
0	-1	1
0	0	0

Caractéristiques d'un contour

Ça peut être aussi approximé par $\frac{\partial f}{\partial}x = \frac{f(x+1) - f(x-1)}{2}$

Ce qui est équivalent à une convolution par le masque 1D :

0	0	0
-1	0	1
0	0	0

Caractéristiques d'un contour

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)$$

Gradient

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

Amplitude (norme) du gradient

- Le gradient est un filtre dérivateur = passe haut !
- $\nabla_x f$ détails suivant x, $\nabla_y f$ détails suivant y.
- $|| \nabla f ||$ détails de l'image.
- $||\nabla f ||$ > seuil extrait les contours des objets.

Détecteur de Contour basé sur le Gardient

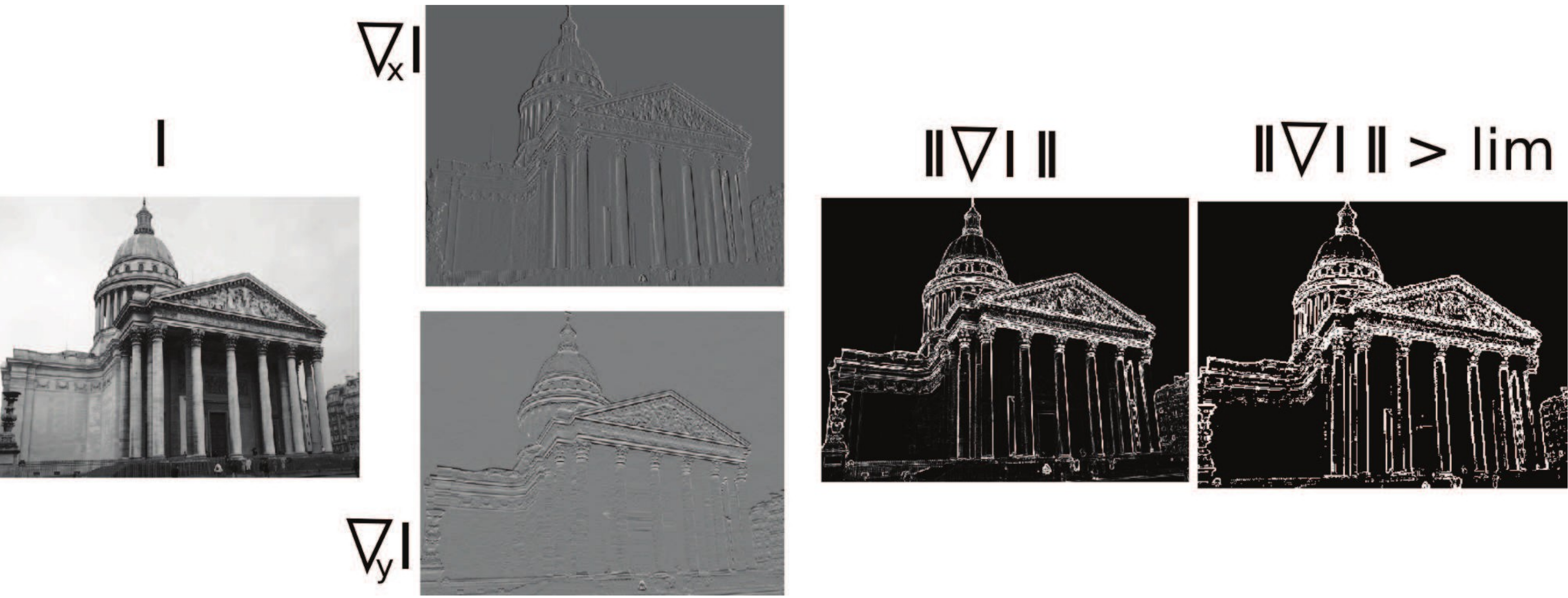
Un simple détecteur de contour utilisant l'amplitude du gradient :

- Calcule le vecteur gradient à chaque pixel par la convolution de l'image avec des filtres de dérivation horizontale et verticale.

- Calcule l'amplitude du gradient à chaque pixel

- Si l'amplitude à un pixel dépasse un seuil donné, marque le pixel comme un point de contour possible

Caractéristiques d'un contour



Caractéristiques d’un contour

- Exemple de filtres spatial d’accentuation (de détection de contours) basés sur le gradient :
  - Roberts
  - Prewitt
  - Sobel
  - ...

Détecteur de Contours basés sur le Gradient :

Détecteur de contour de Prewitt :

Deux masques pour approximer | G x | et | G y | dans | ∇ f | :

$$\left|\nabla f\right| \approx \left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right| = \left|G_x\right| + \left|G_y\right|$$

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

1	1	1
0	0	0
-1	-1	1

Détecteurs de Contours basés sur le Gradient :

Détecteur de contour de Sobel:

Deux masques pour approximer | G x | et | G y | dans | ∇ f | :

$$|\nabla f| \approx \left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right| = |G_x| + |G_y|$$

Il utilise un coefficient de 2 pour donner une importance au pixel du centre

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

1	2	1
0	0	0
-1	-2	1

Détecteurs de Contours basés sur le Gradient :

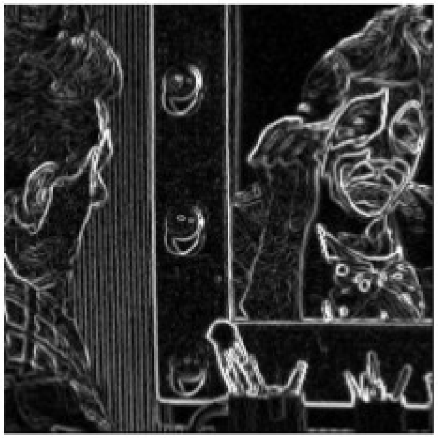
Détecteur de contour de Roberts (opérateur gradient en croix):

Deux masques pour approximer | G x | et | G y | dans | ∇ f | :

$$|\nabla f| \approx \left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right| = |G_x| + |G_y|$$

-1	0
0	1

0	-1
1	0



Détecteurs de Contours basés sur le Gradient :

Les Opérateurs Compas :

- Les opérateurs Sobel et Prewitt n’utilisent que 2 directions pour detecter les amplitudes des contours. Donc sont non sensitives à l’orientation.
- Solution utiliser plusieurs filtres, chacun sensitif à un petit intervalle d’orientation. (opérateurs compas).

Détecteurs de Contours basés sur le Gradient :

Les Opérateurs Compas :

L'opérateur de détection de contour proposé par Kirsh utilise 8 filtres avec des orientations espacé de 45 degrés.

$$H_0^K = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad H_4^K = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

$$H_1^K = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} \qquad H_5^K = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}$$

$$H_2^K = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \qquad H_6^K = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

$$H_3^K = \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix} \qquad H_7^K = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$$

n'a besoin de calculer que 4 filtres comme $H_4 = - H_0$, etc

Détecteurs de Contours basés sur le Gradient :

Filtre compas :

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix}$$

$\uparrow$ N $\quad \nwarrow$ NW $\quad \leftarrow$ W $\quad \swarrow$ SW

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}$$

$\downarrow$ S $\quad \searrow$ SE $\quad \rightarrow$ E $\quad \nearrow$ NE

Le gradient est défini par :

$$g(x,y) = \max_k |g_k(x,y)|$$

k donne l'orientation du gradient

Détecteurs de Contours basés sur le Gradient :

Détecteur de Canny :

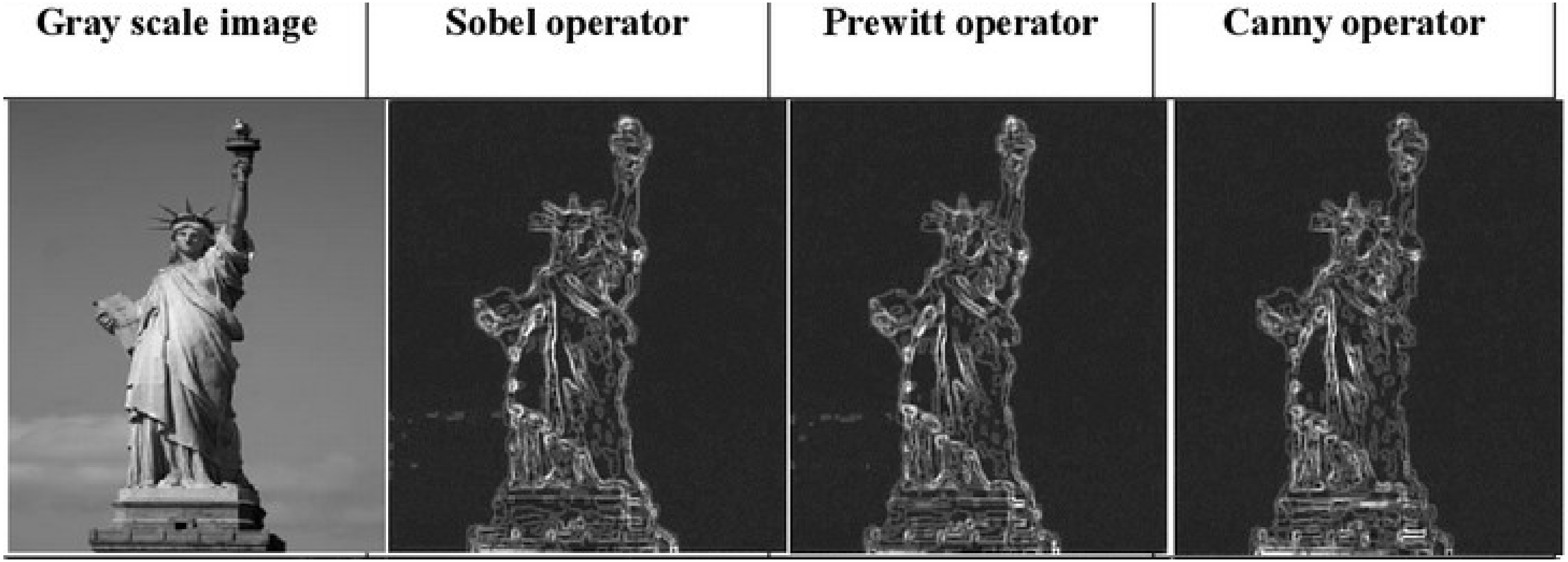
- Le plus utilisé des détecteurs
- Il tient compte que l'image peut être bruitée

Principe : Calculer le gradient de l'image convoluée par une gaussienne

Astuce : Le gradient d'une gaussienne est aussi une gaussienne :

$$I_x = G_\sigma^x * I \quad I_y = G_\sigma^y * I$$

Détecteurs de Contours basés sur le Gradient :



Détecteurs de Contours basés sur le Gradient :

Décision contour : Soit G le gradient au pixel (x,y)

.

- **Décision par un seuillage**

Le pixel (x,y) est considéré un point contour si G > seuil

- **Décision par seuillage par hystéresis :** On définit deux seuils S_b (seuil bas) et S_h (seuil haut). La classification en pts de contour ou non est donnée :

$D > S_h$ → point contour (PC)

$D < S_b$ → point non contour (PNC)

$S_b < D < S_h$ → point contour possible (PCP)

Un point de contour possible (PCP) est ensuite classé comme un PC lorsque il a un voisin PC, ou PNC dans le cas contraire

Détection de Contours utilisant le Laplacien :

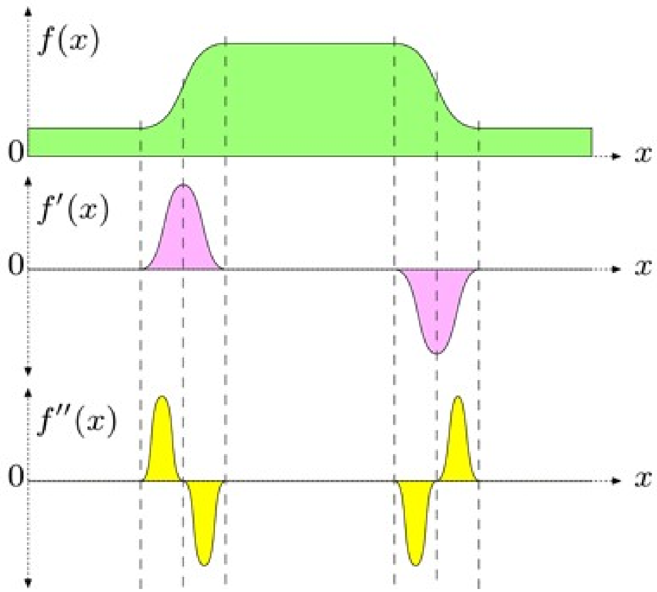
Problème avec les détecteurs de contours basés sur la première dérivée :

- Le contour est proportionnel à l'intensité de la transition.

- Le contour peut être difficile à localiser précisément.

<u>Solution :</u> Utiliser la dérivée seconde

Rappel : Un contour est passage par zéro de la dérivée seconde..



Première dérivée
Contenant des pointes

Deuxième dérivée
Contenant des passages par 0

Détection de Contours utilisant le Laplacien :

Opérateur Laplace : combine les dérivées secondes dans les directions horizontale et verticales.

L'opérateur de Laplace est défini par :

$$(\nabla^2 f)(x,y) = \frac{\partial^2 f}{\partial^2 x}(x,y) + \frac{\partial^2 f}{\partial^2 y}(x,y)$$

La dérivée seconde sur l'axe des x

La dérivée seconde sur l'axe des y

Détection de Contours utilisant le Laplacien :

L'approximation numérique du laplacien est donnée par :

$$\nabla^2 f(x,y) = [f(x+1,y) - f(x,y)] - [f(x,y) - f(x-1,y)] + [f(x,y+1) - f(x,y)] - [f(x,y) - f(x,y-1)]$$

$$\boxed{= [f(x+1,y) + f(x-1,y) + f(x,y+1) - f(x,y-1)] - 4f(x,y)}$$

Masque du filtre laplacien

0	1	0
1	-4	1
0	1	0

Détection de Contours utilisant le Laplacien :

L'approximation numérique du laplacien est donnée par :

Masque du filtre laplacien

0	1	0
1	-4	1
0	1	0

On peut aussi tenir compte
Des orientation de 45 degrés

1	1	1
1	-8	1
1	1	1

Détection de Contours utilisant le Laplacien :

Une fois, l'opérateur Δ approximant localement le Laplacien de l'image est défini

1. Ils sera convolué avec l'image I

$$\Delta \mathrm{I} \cong I * \Delta$$

2. Puis, on détecte les points où il y'a passage par zéro, autrement dit, les points où il y'a variation du signe de $\Delta \mathrm{I}$ par rapport aux points avoisinants

Remarque : Cet opérateur est extrêmement sensible au bruit, de ce fait son utilisation sans l'adjoindre à un lisseur (tel que pour Sobel) est obsolète

Détection de Contours utilisant le Laplacien :

Opérateur Sobel

Opérateur Laplacien

Détection de Contours utilisant le Laplacien :

Avantages du filtre du deuxième ordre par rapport au filtre du premier ordre :

- Les contours sont plus fins
- Meilleurs réponse pour les petits détails
- Double réponses par rapport aux contours
- Indépendant de l'orientation : un seul masque pour tous les contours

Détection de Contours utilisant le Laplacien :

<u>Opérateur Laplacian d'une Gaussienne LoG</u> (Laplacian of Gaussien)

- Idem qu'avec les opérateurs approximant le contour, la problématique de détection du bruit comme contour se pose également pour les opérateurs approximant le Laplacian.

- L'opérateur LoG vise
  1. d'abord à lisser l'image : par un filtre Gaussienn H_{Gauss}
  2. Puis détecter les contours : par le Laplacian Δ

- On effectuera donc : $\Delta \mathrm{I} \cong I * H_{Gauss} * \Delta = I * \underbrace{\Delta H_{Gauss}}_{\Delta * H_{Gauss}}$

- $\boldsymbol{\Delta H_{Gauss}}$ **est l'opérateur LoG**

Détection de Contours utilisant le Laplacien :

- **Opérateur Laplacian d'une Gaussienne LoG** **(Laplacian of Gaussien)**
  - Le calcul du Laplacien d'une gaussienne donne :

$$\Delta H_{Gauss} = \frac{4}{\sqrt{2\pi\sigma}}\left(\frac{x^2+y^2}{2\sigma^2} - 1\right) e^{\frac{x^2+y^2}{2\sigma^2}}$$

  - Pour σ=0.5, et
    - sur un voisinage 3×3, on obtient : $\Delta H_{Gauss} = \begin{bmatrix} 0.4038 & 0.8021 & 0.4038 \\ 0.8021 & -4.8233 & 0.8021 \\ 0.4038 & 0.8021 & 0.4038 \end{bmatrix}$
    - sur un voisinage 5×5, on obtient :

$$\Delta H_{Gauss} = \begin{bmatrix} 0.0448 & 0.0468 & 0.0564 & 0.0468 & 0.0448 \\ 0.0468 & 0.3167 & 0.7146 & 0.3167 & 0.0468 \\ 0.0564 & 0.7146 & -4.9048 & 0.7146 & 0.0564 \\ 0.0468 & 0.3167 & 0.7146 & 0.3167 & 0.0468 \\ 0.0448 & 0.0468 & 0.0564 & 0.0468 & 0.0448 \end{bmatrix}$$

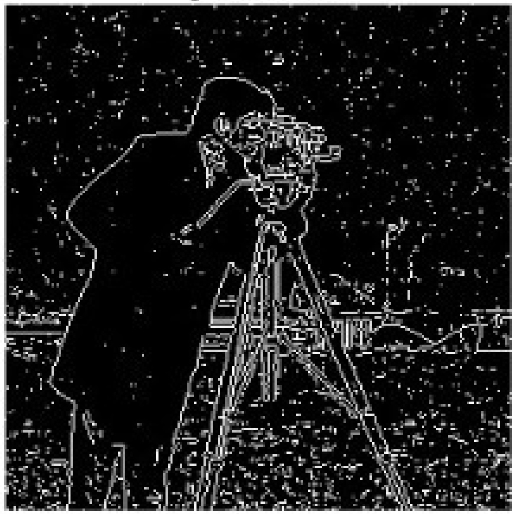
Détection de Contours utilisant le Laplacien :

LoG (sigma=1) *LoG* (sigma=2) *LoG* (sigma=3)

Détection de Contours utilisant le Laplacien :

Image bruitée par un bruit gaussien

prewitt



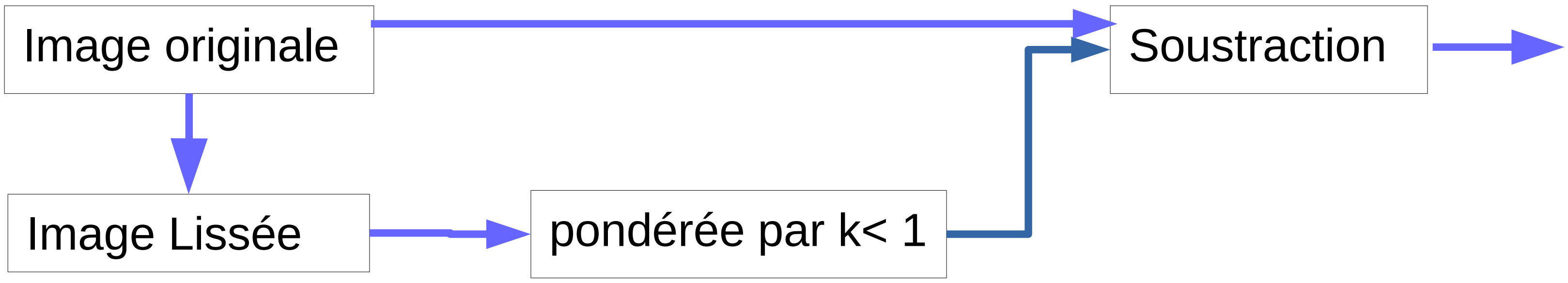
sobel

LoG (sigma=2.3)

Masquage Flou :

Le masquage Flou (Unsharp Masking USM) : est une technique d'accentuation des contours dans une image.

Elle est basée sur la combinaison de l'image originale et la version lissée de l'image.



Masquage Flou :

Supprimer l'image lissée par une Gaussien pour obtenir le masque de renforcement des contours :

$$M \leftarrow I - (I * \tilde{H}) = I - \tilde{I}$$

Ajouter le masque à l'image avec un poids :

$$\check{I} \leftarrow I + a \cdot M$$

Ensemble :

$$\check{I} \leftarrow I + a \cdot (I - \tilde{I}) = (1 + a) \cdot I - a \cdot \tilde{I}$$

Image accentuée = original + (original − lissée) × facteur.

Masquage Flou :

(a) Original

(b)

(d) $\sigma = 2.5$

(e)

(g) $\sigma = 10.0$

(h)

Exemple de masquage flou

Filtre de Laplace + l'image originale → Accentuation

0	-1	0
-1	5	-1
0	-1	0

$$\begin{bmatrix}0&0&0\\0&1&0\\0&0&0\end{bmatrix}+\left(\begin{bmatrix}0&0&0\\0&1&0\\0&0&0\end{bmatrix}-\begin{bmatrix}0&1&0\\1&1&1\\0&1&0\end{bmatrix}/5\right)5=\begin{bmatrix}0&-1&0\\-1&5&-1\\0&-1&0\end{bmatrix}$$

Image originale

Image accentué