

Partie B — thème 5

Détection de contours



- ▶ variations « significatives » donnant des indications sur les objets ou structures de l'image
- ▶ définition intuitive : discontinuité de luminance ou de couleur
- ▶ comment caractériser un contour, sur quels critères ?
- ▶ comment passer d'une image à une image de contours ?

Partie B — thème 5

Détection de contours



- ▶ variations « significatives » donnant des indications sur les objets ou structures de l'image
- ▶ définition intuitive : discontinuité de luminance ou de couleur
- ▶ comment caractériser un contour, sur quels critères ?
- ▶ comment passer d'une image à une image de contours ?

Partie B — thème 5

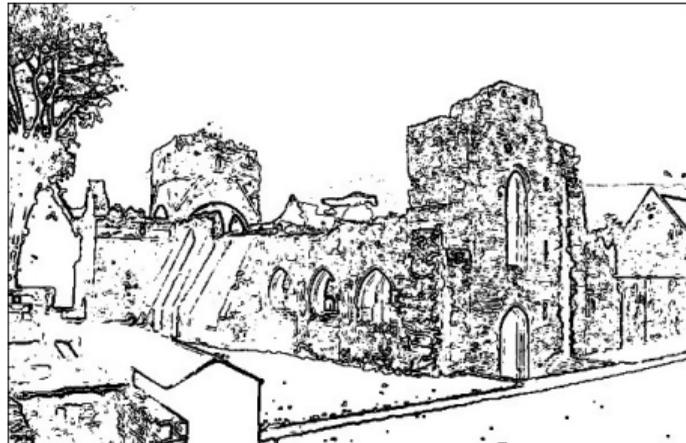
Détection de contours



- ▶ variations « significatives » donnant des indications sur les objets ou structures de l'image
- ▶ définition intuitive : discontinuité de luminance ou de couleur
- ▶ comment caractériser un contour, sur quels critères ?
- ▶ comment passer d'une image à une image de contours ?

Partie B — thème 5

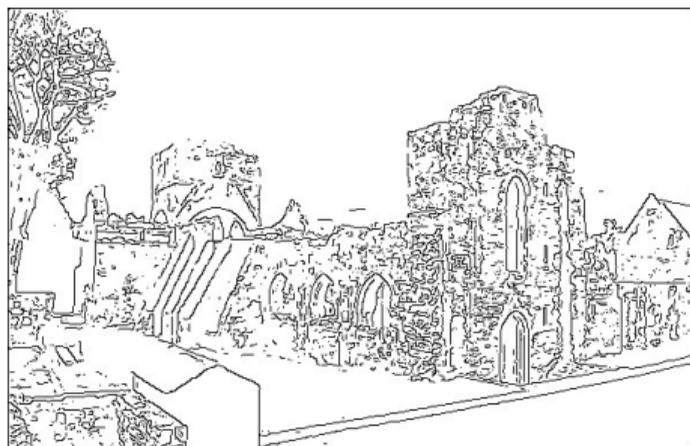
Détection de contours



- ▶ variations « significatives » donnant des indications sur les objets ou structures de l'image
- ▶ définition intuitive : discontinuité de luminance ou de couleur
- ▶ comment caractériser un contour, sur quels critères ?
- ▶ comment passer d'une image à une image de contours ?

Partie B — thème 5

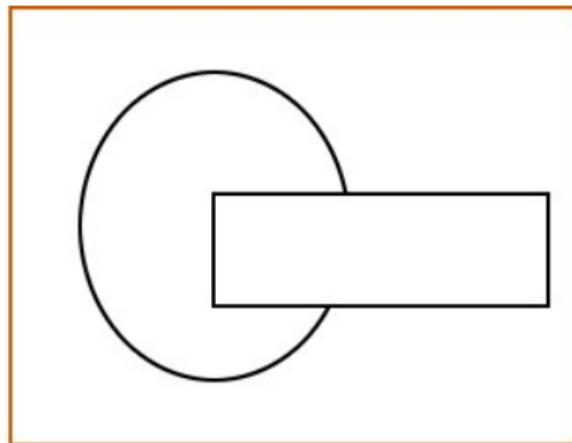
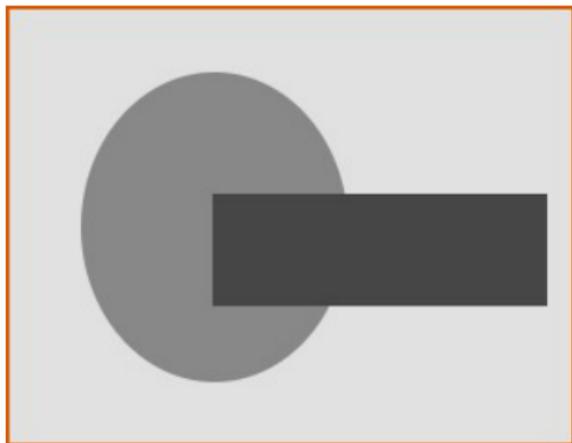
Détection de contours



- ▶ variations « significatives » donnant des indications sur les objets ou structures de l'image
- ▶ définition intuitive : discontinuité de luminance ou de couleur
- ▶ comment caractériser un contour, sur quels critères ?
- ▶ comment passer d'une image à une image de contours ?

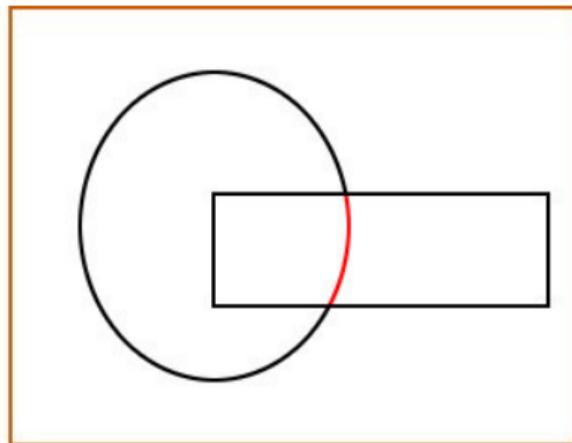
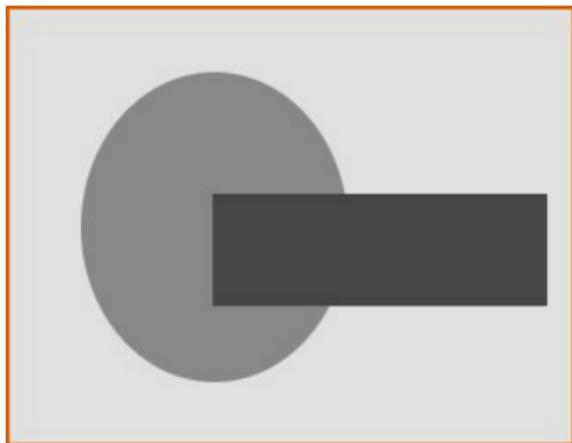
1. Caractérisation globale ou locale des contours

- ▶ discontinuités **globales** : frontières de régions
 - ▶ segmentation : cohérence, contours fermés
 - ▶ possibilité d'ajustement à un modèle cible
- ▶ discontinuités **locales** : contours de luminance
 - ▶ plus simple mais ne préserve pas la cohérence globale : dégradées, bruit
 - ▶ possibilité de prise en compte des textures
- ▶ détection de contours : méthodes **locales**
- ▶ compromis contours / texture : paramétrage de la **sensibilité** de la détection



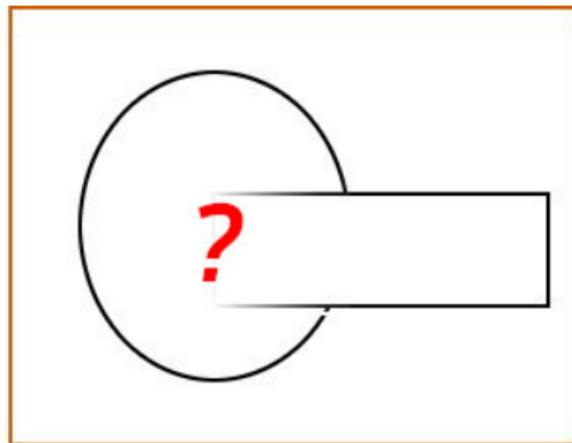
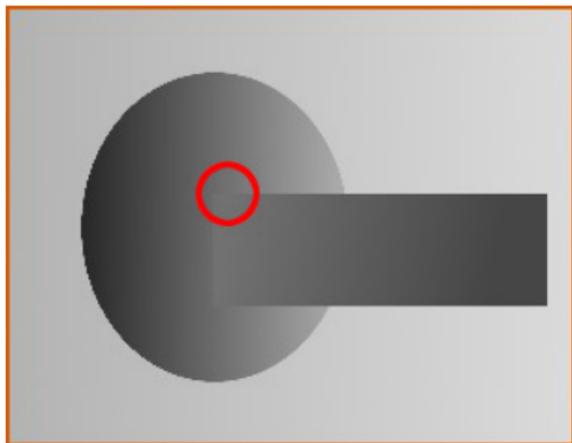
1. Caractérisation globale ou locale des contours

- ▶ discontinuités **globales** : frontières de régions
 - ▶ segmentation : cohérence, contours fermés
 - ▶ possibilité d'ajustement à un modèle cible
- ▶ discontinuités **locales** : contours de luminance
 - ▶ plus simple mais ne préserve pas la cohérence globale : dégradées, bruit
 - ▶ possibilité de prise en compte des textures
- ▶ détection de contours : méthodes **locales**
- ▶ compromis contours / texture : paramétrage de la **sensibilité** de la détection



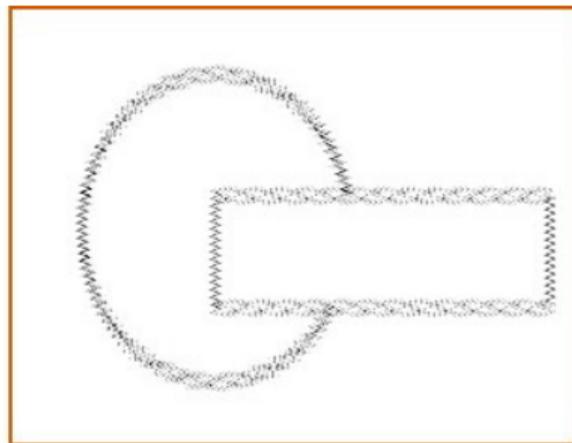
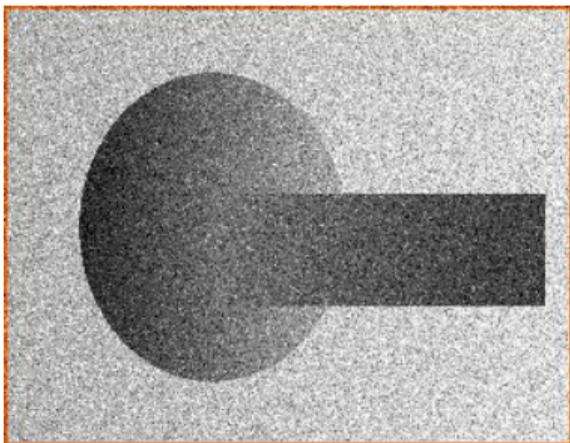
1. Caractérisation globale ou locale des contours

- ▶ discontinuités **globales** : frontières de régions
 - ▶ segmentation : cohérence, contours fermés
 - ▶ possibilité d'ajustement à un modèle cible
- ▶ discontinuités **locales** : contours de luminance
 - ▶ plus simple mais ne préserve pas la cohérence globale : dégradées, bruit
 - ▶ possibilité de prise en compte des textures
- ▶ détection de contours : méthodes **locales**
- ▶ compromis contours / texture : paramétrage de la **sensibilité** de la détection



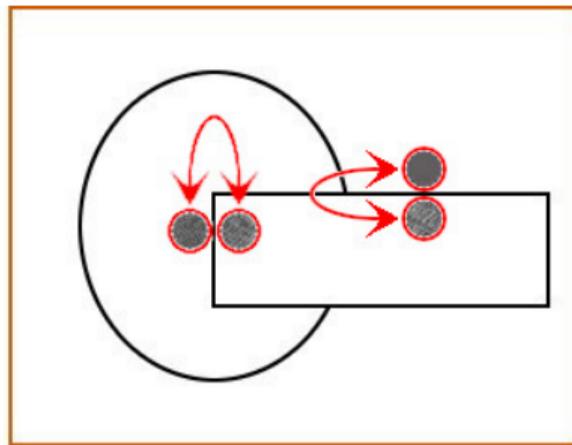
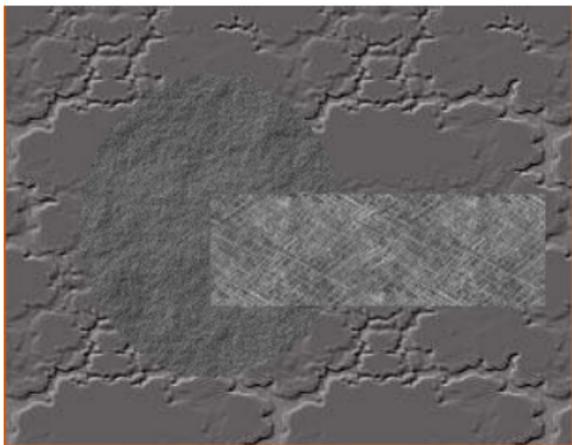
1. Caractérisation globale ou locale des contours

- ▶ discontinuités **globales** : frontières de régions
 - ▶ segmentation : cohérence, contours fermés
 - ▶ possibilité d'ajustement à un modèle cible
- ▶ discontinuités **locales** : contours de luminance
 - ▶ plus simple mais ne préserve pas la cohérence globale : dégradées, bruit
 - ▶ possibilité de prise en compte des textures
- ▶ détection de contours : méthodes **locales**
- ▶ compromis contours / texture : paramétrage de la **sensibilité** de la détection



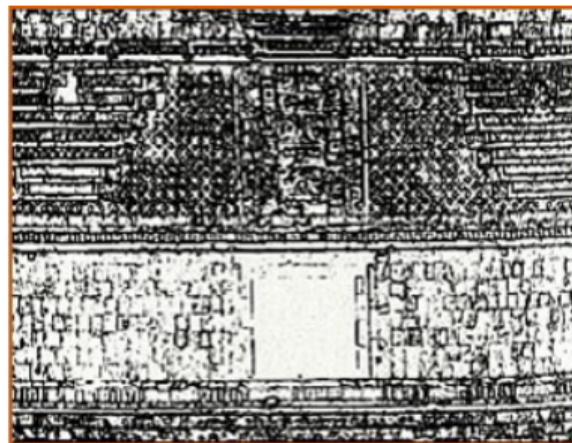
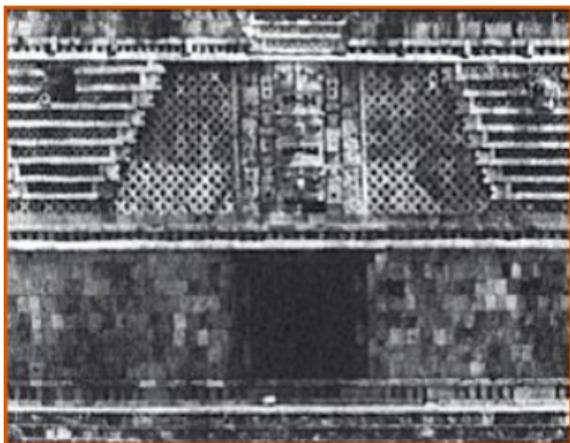
1. Caractérisation globale ou locale des contours

- ▶ discontinuités **globales** : frontières de régions
 - ▶ segmentation : cohérence, contours fermés
 - ▶ possibilité d'ajustement à un modèle cible
- ▶ discontinuités **locales** : contours de luminance
 - ▶ plus simple mais ne préserve pas la cohérence globale : dégradées, bruit
 - ▶ possibilité de prise en compte des textures
- ▶ détection de contours : méthodes **locales**
- ▶ compromis contours / texture : paramétrage de la **sensibilité** de la détection



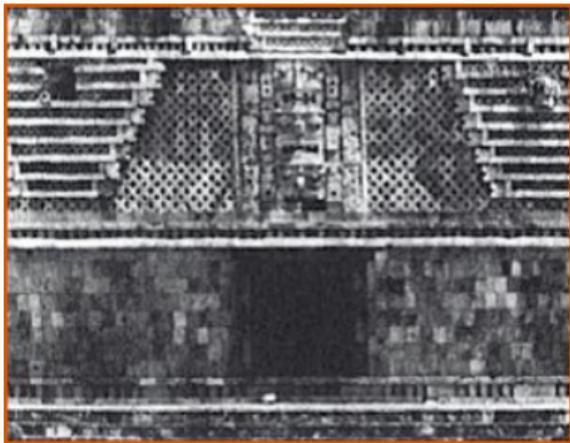
1. Caractérisation globale ou locale des contours

- ▶ discontinuités **globales** : frontières de régions
 - ▶ segmentation : cohérence, contours fermés
 - ▶ possibilité d'ajustement à un modèle cible
- ▶ discontinuités **locales** : contours de luminance
 - ▶ plus simple mais ne préserve pas la cohérence globale : dégradées, bruit
 - ▶ possibilité de prise en compte des textures
- ▶ détection de contours : méthodes **locales**
- ▶ compromis contours / texture : paramétrage de la **sensibilité** de la détection



1. Caractérisation globale ou locale des contours

- ▶ discontinuités **globales** : frontières de régions
 - ▶ segmentation : cohérence, contours fermés
 - ▶ possibilité d'ajustement à un modèle cible
- ▶ discontinuités **locales** : contours de luminance
 - ▶ plus simple mais ne préserve pas la cohérence globale : dégradées, bruit
 - ▶ possibilité de prise en compte des textures
- ▶ détection de contours : méthodes **locales**
- ▶ compromis contours / texture : paramétrage de la **sensibilité** de la détection

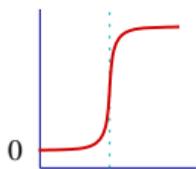


2. Détection des contours et méthodes dérivatives

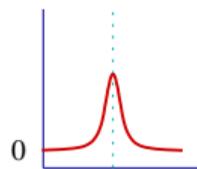
2.1. Caractérisation des contours selon une direction



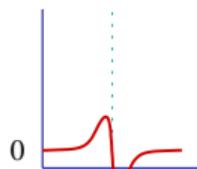
*contour vertical,
caractérisation
horizontale*



*modélisation de
la transition*



dérivée première



dérivée seconde

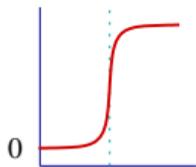
- ▶ domaine continu : modélisation par une augmentation ou diminution d'amplitude
- ▶ caractéristiques du contour : importance de la variation (pente), coordonnées du point médian
- ▶ localisation dérivative du contour
 - ▶ ordre 1 : extrémums de la dérivée première
 - ▶ ordre 2 : passage par zéro de la dérivée seconde

2. Détection des contours et méthodes dérivatives

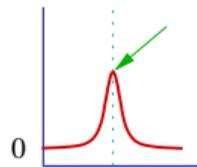
2.1. Caractérisation des contours selon une direction



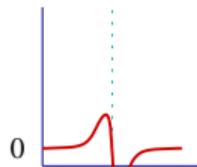
*contour vertical,
caractérisation
horizontale*



*modélisation de
la transition*



dérivée première



dérivée seconde

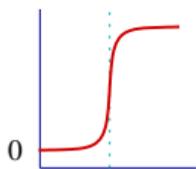
- ▶ domaine continu : modélisation par une augmentation ou diminution d'amplitude
- ▶ caractéristiques du contour : importance de la variation (pente), coordonnées du point médian
- ▶ localisation dérivative du contour
 - ▶ *ordre 1 : extrêmes de la dérivée première*
 - ▶ *ordre 2 : passage par zéro de la dérivée seconde*

2. Détection des contours et méthodes dérivatives

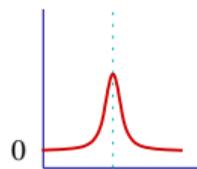
2.1. Caractérisation des contours selon une direction



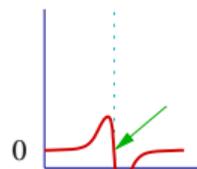
*contour vertical,
caractérisation
horizontale*



*modélisation de
la transition*



dérivée première



dérivée seconde

- ▶ domaine continu : modélisation par une augmentation ou diminution d'amplitude
- ▶ caractéristiques du contour : importance de la variation (pente), coordonnée du point médian
- ▶ localisation dérivative du contour
 - ▶ ordre 1 : extrêmes de la dérivée première
 - ▶ ordre 2 : passage par zéro de la dérivée seconde

2.2. Caractérisation des contours en dimension 2

o Principe

- ▶ déterminer l'emplacement des transitions significatives dans toutes les directions
- ▶ produire une image différentielle représentative des transitions spatiale

o Ordre 1 : caractérisation par le vecteur gradient de la fonction image

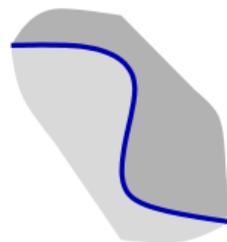
$$I(x, y) = \begin{pmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{pmatrix}$$

analogie avec les images topographiques

niveaux de gris \leftrightarrow altitudes

transitions \leftrightarrow dénivelés

contours \leftrightarrow courbes de niveau



direction du gradient

normale au contour (pente)

$$\Phi = \arctan \left(\frac{\partial I}{\partial y} / \frac{\partial I}{\partial x} \right)$$

norme du gradient

importance du contour (dénivelé)

$$|\nabla I(i, j)| = \sqrt{\left(\frac{\partial I}{\partial i}\right)^2 + \left(\frac{\partial I}{\partial j}\right)^2}$$

2.2. Caractérisation des contours en dimension 2

o Principe

- ▶ déterminer l'emplacement des transitions significatives dans toutes les directions
- ▶ produire une image différentielle représentative des transitions spatiale

o Ordre 1 : caractérisation par le vecteur gradient de la fonction image

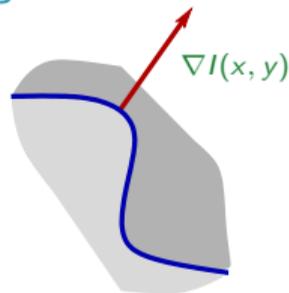
$$I(x, y) = \begin{pmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{pmatrix}$$

analogie avec les images topographiques

niveaux de gris \leftrightarrow altitudes

transitions \leftrightarrow dénivelés

contours \leftrightarrow courbes de niveau



direction du gradient

normale au contour (pente)

$$\Phi = \arctan \left(\frac{\partial I}{\partial y} / \frac{\partial I}{\partial x} \right)$$

norme du gradient

importance du contour (dénivelé)

$$|\nabla I(i, j)| = \sqrt{\left(\frac{\partial I}{\partial i}\right)^2 + \left(\frac{\partial I}{\partial j}\right)^2}$$

2.2. Caractérisation des contours en dimension 2

o Principe

- ▶ déterminer l'emplacement des transitions significatives dans toutes les directions
- ▶ produire une image différentielle représentative des transitions spatiale

o Ordre 1 : caractérisation par le vecteur gradient de la fonction image

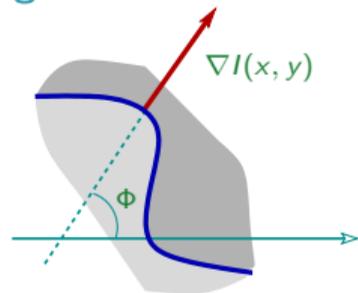
$$I(x, y) = \begin{pmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{pmatrix}$$

analogie avec les images topographiques

niveaux de gris \leftrightarrow altitudes

transitions \leftrightarrow dénivelés

contours \leftrightarrow courbes de niveau



direction du gradient

normale au contour (pente)

$$\Phi = \arctan \left(\frac{\partial I}{\partial y} / \frac{\partial I}{\partial x} \right)$$

norme du gradient

importance du contour (dénivelé)

$$|\nabla I(i, j)| = \sqrt{\left(\frac{\partial I}{\partial i}\right)^2 + \left(\frac{\partial I}{\partial j}\right)^2}$$

○ **Ordre 2 : utilisation du Laplacien (approximation de la dérivée seconde)**

- ▶ expression de la dérivée seconde de la fonction image $I(x, y)$:

$$\frac{\partial^2 I}{\partial \vec{N}^2} = \frac{\partial^2 I}{\partial x^2} \cos^2 \Phi + 2 \frac{\partial^2 I}{\partial x \partial y} \cos \Phi \sin \Phi + \frac{\partial^2 I}{\partial y^2} \sin^2 \Phi$$

- ▶ approximation par l'opérateur laplacien

$$\Delta I(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

- ▶ mêmes passages par zéro que la dérivée seconde
- ▶ simple à mettre en œuvre
- ▶ isotrope (invariant par rotation)

○ **Ordre 2 : utilisation du Laplacien (approximation de la dérivée seconde)**

- ▶ expression de la dérivée seconde de la fonction image $I(x, y)$:

$$\frac{\partial^2 I}{\partial \vec{N}^2} = \frac{\partial^2 I}{\partial x^2} \cos^2 \Phi + 2 \frac{\partial^2 I}{\partial x \partial y} \cos \Phi \sin \Phi + \frac{\partial^2 I}{\partial y^2} \sin^2 \Phi$$

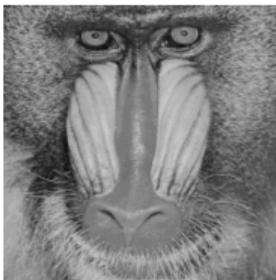
- ▶ approximation par l'opérateur laplacien

$$\Delta I(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

- ▶ mêmes passages par zéro que la dérivée seconde
- ▶ simple à mettre en œuvre
- ▶ isotrope (invariant par rotation)

2.3. Principe des méthodes dérivatives d'ordre 1 ou 2

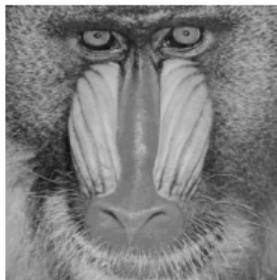
- ▶ différenciation spatiale du premier ordre
 - ▶ calcul du **gradient**
 - ▶ recherche des maximums de la norme du gradient
- ▶ différenciation spatiale du second ordre
 - ▶ calcul du **laplacien**
 - ▶ recherche des passages par zéro



- ▶ construction de l'image des contours d'une image monochrome 2D
 - ▶ images différentielles : calcul des **dérivées partielles discrètes**
 - ▶ localisation précise des contours : **seuillage**

2.3. Principe des méthodes dérivatives d'ordre 1 ou 2

- ▶ différenciation spatiale du premier ordre
 - ▶ calcul du **gradient**
 - ▶ recherche des maximums de la norme du gradient
- ▶ différenciation spatiale du second ordre
 - ▶ calcul du **laplacien**
 - ▶ recherche des passages par zéro



- ▶ construction de l'image des contours d'une image monochrome 2D
 - ▶ images différentielles : calcul des **dérivées partielles discrètes**
 - ▶ localisation précise des contours : **seuillage**

3. Calcul des dérivées partielles d'une image discrètes

3.1. Formulation de la dérivation partielle discrète

- ▶ définition continue :

$$I : D \subset \mathbb{R}^2 \rightarrow [0, 1] \quad (\text{fonction image})$$

$$\frac{\partial I(x, y)}{\partial x} = \lim_{h \rightarrow 0} \frac{I(x + h, y) - I(x, y)}{h}$$

$$\frac{\partial I(x, y)}{\partial y} = \lim_{h \rightarrow 0} \frac{I(x, y + h) - I(x, y)}{h}$$

- ▶ en discret : $h \geq 1 \Rightarrow$ approximation par différence finie ($h = 1$)

$$\frac{\partial I(i, j)}{\partial j} = I(i, j + 1) - I(i, j)$$

$$\frac{\partial I(i, j)}{\partial i} = I(i + 1, j) - I(i, j)$$

3. Calcul des dérivées partielles d'une image discrètes

3.1. Formulation de la dérivation partielle discrète

- ▶ définition **continue** :

$$I : D \subset \mathbb{R}^2 \rightarrow [0, 1] \quad (\text{fonction image})$$

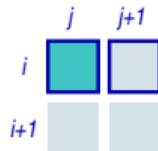
$$\frac{\partial I(x, y)}{\partial x} = \lim_{h \rightarrow 0} \frac{I(x + h, y) - I(x, y)}{h}$$

$$\frac{\partial I(x, y)}{\partial y} = \lim_{h \rightarrow 0} \frac{I(x, y + h) - I(x, y)}{h}$$

- ▶ en discret : $h \geq 1 \Rightarrow$ approximation par **différence finie** ($h = 1$)

$$\frac{\partial I(i, j)}{\partial j} = I(i, j + 1) - I(i, j)$$

$$\frac{\partial I(i, j)}{\partial i} = I(i + 1, j) - I(i, j)$$



3. Calcul des dérivées partielles d'une image discrètes

3.1. Formulation de la dérivation partielle discrète

- ▶ définition continue :

$$I : D \subset \mathbb{R}^2 \rightarrow [0, 1] \quad (\text{fonction image})$$

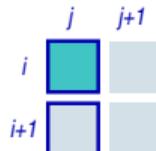
$$\frac{\partial I(x, y)}{\partial x} = \lim_{h \rightarrow 0} \frac{I(x + h, y) - I(x, y)}{h}$$

$$\frac{\partial I(x, y)}{\partial y} = \lim_{h \rightarrow 0} \frac{I(x, y + h) - I(x, y)}{h}$$

- ▶ en discret : $h \geq 1 \Rightarrow$ approximation par différence finie ($h = 1$)

$$\frac{\partial I(i, j)}{\partial j} = I(i, j + 1) - I(i, j)$$

$$\frac{\partial I(i, j)}{\partial i} = I(i + 1, j) - I(i, j)$$



- ▶ le calcul des DP discrètes se traduit par un produit de convolution :

$$\frac{\partial I}{\partial i} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} * I$$

$$\frac{\partial I}{\partial j} = \begin{bmatrix} -1 & 1 \end{bmatrix} * I$$

- ▶ on peut également utiliser les noyaux : $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$ et $\begin{bmatrix} 1 & -1 \end{bmatrix}$

- ▶ la dérivée partielle d'une image possède des valeurs positives et négatives :

$$\frac{\partial I}{\partial i}, \frac{\partial I}{\partial j} : [0..255] \longrightarrow [-255..255]$$

ce n'est donc pas une image directement affichable

- ▶ le calcul des DP discrètes se traduit par un produit de convolution :

$$\frac{\partial I}{\partial i} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} * I$$

$$\frac{\partial I}{\partial j} = \begin{bmatrix} -1 & 1 \end{bmatrix} * I$$

- ▶ on peut également utiliser les noyaux : $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$ et $\begin{bmatrix} 1 & -1 \end{bmatrix}$

- ▶ la dérivée partielle d'une image possède des valeurs positives et négatives :

$$\frac{\partial I}{\partial i}, \frac{\partial I}{\partial j} : [0..255] \longrightarrow [-255..255]$$

ce n'est donc pas une image directement affichable

- ▶ le calcul des DP discrètes se traduit par un produit de convolution :

$$\frac{\partial I}{\partial i} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} * I$$
$$\frac{\partial I}{\partial j} = \begin{bmatrix} -1 & 1 \end{bmatrix} * I$$

- ▶ on peut également utiliser les noyaux : $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$ et $\begin{bmatrix} 1 & -1 \end{bmatrix}$
- ▶ la dérivée partielle d'une image possède des valeurs positives et négatives :

$$\frac{\partial I}{\partial i}, \frac{\partial I}{\partial j} : [0..255] \longrightarrow [-255..255]$$

ce n'est donc pas une image directement affichable

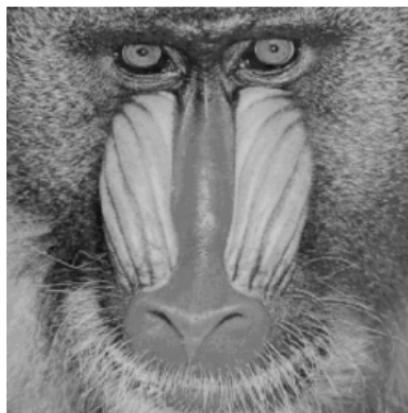
- ▶ deux stratégies pour ramener les valeurs de la DP dans la dynamique de l'image

$$[-255..255] \longrightarrow [0..255]$$

- ▶ $\left| \frac{\partial I(i,j)}{\partial \cdot} \right|$ (valeur absolue) : perte de l'orientation de la dérivée
- ▶ $\frac{1}{2} \frac{\partial I(i,j)}{\partial \cdot} + 127$ (normalisation par rapport à un gris moyen) : réduction de la dynamique mais conservation de l'orientation

valeur absolue

normalisation

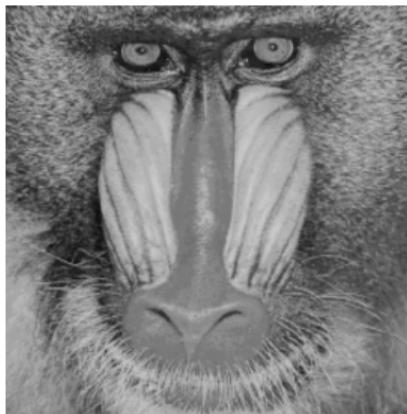


I

$$\left| \begin{bmatrix} -1 & 1 \end{bmatrix} * I \right|$$

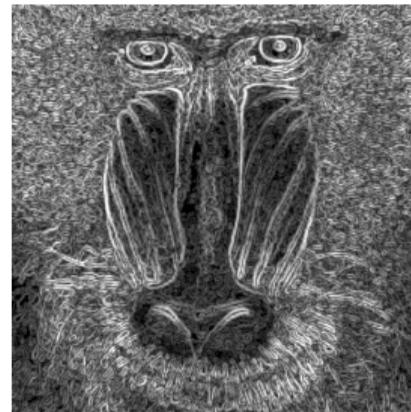
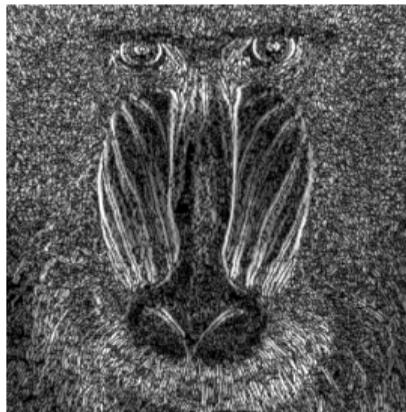
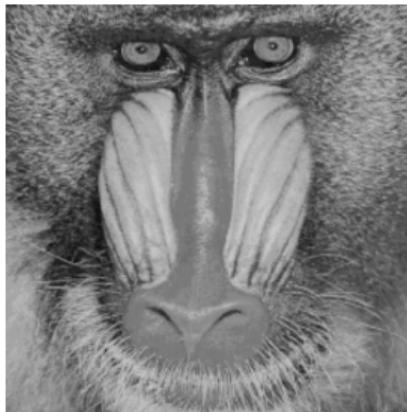
$$\begin{bmatrix} -\frac{1}{2} & \frac{1}{2} \end{bmatrix} * I + 127$$

★ Visualisation des dérivées partielles : valeur absolue et maximum



A	$\max\left(\left \frac{\partial I}{\partial i}\right , \left \frac{\partial I}{\partial j}\right \right)$	$\left \frac{\partial I}{\partial i}\right $	$\left \frac{\partial I}{\partial j}\right $
B	$\left \frac{\partial I}{\partial j}\right $	$\left \frac{\partial I}{\partial i}\right $	$\max\left(\left \frac{\partial I}{\partial i}\right , \left \frac{\partial I}{\partial j}\right \right)$
C	$\left \frac{\partial I}{\partial i}\right $	$\left \frac{\partial I}{\partial j}\right $	$\max\left(\left \frac{\partial I}{\partial i}\right , \left \frac{\partial I}{\partial j}\right \right)$
D	$\max\left(\left \frac{\partial I}{\partial i}\right , \left \frac{\partial I}{\partial j}\right \right)$	$\left \frac{\partial I}{\partial j}\right $	$\left \frac{\partial I}{\partial i}\right $

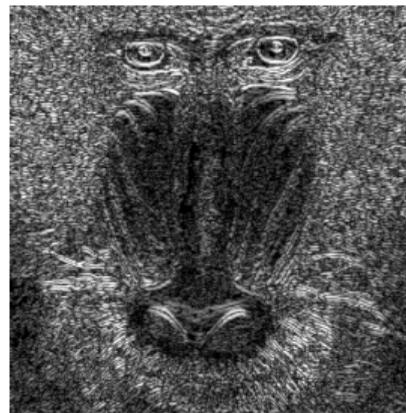
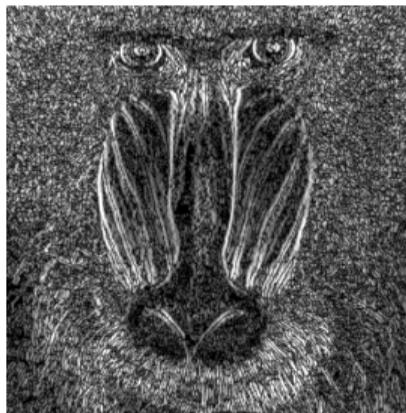
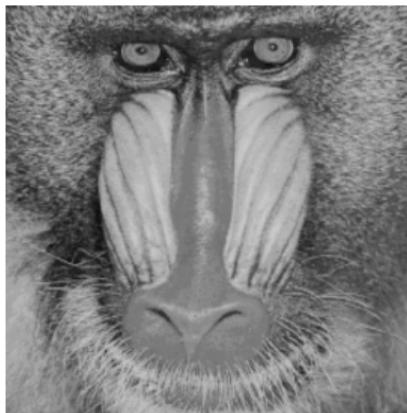
★ Visualisation des dérivées partielles : valeur absolue et maximum



- A
- B
- C
- D

$\max\left(\left \frac{\partial I}{\partial i}\right , \left \frac{\partial I}{\partial j}\right \right)$	$\left \frac{\partial I}{\partial i}\right $	$\left \frac{\partial I}{\partial j}\right $
$\left \frac{\partial I}{\partial j}\right $	$\left \frac{\partial I}{\partial i}\right $	$\max\left(\left \frac{\partial I}{\partial i}\right , \left \frac{\partial I}{\partial j}\right \right)$
$\left \frac{\partial I}{\partial i}\right $	$\left \frac{\partial I}{\partial j}\right $	$\max\left(\left \frac{\partial I}{\partial i}\right , \left \frac{\partial I}{\partial j}\right \right)$
$\max\left(\left \frac{\partial I}{\partial i}\right , \left \frac{\partial I}{\partial j}\right \right)$	$\left \frac{\partial I}{\partial j}\right $	$\left \frac{\partial I}{\partial i}\right $

★ Visualisation des dérivées partielles : valeur absolue et maximum



A

$$\max\left(\left|\frac{\partial I}{\partial i}\right|, \left|\frac{\partial I}{\partial j}\right|\right)$$

$$\left|\frac{\partial I}{\partial i}\right|$$

$$\left|\frac{\partial I}{\partial j}\right|$$

B

$$\left|\frac{\partial I}{\partial j}\right|$$

$$\left|\frac{\partial I}{\partial i}\right|$$

$$\max\left(\left|\frac{\partial I}{\partial i}\right|, \left|\frac{\partial I}{\partial j}\right|\right)$$

C

$$\left|\frac{\partial I}{\partial i}\right|$$

$$\left|\frac{\partial I}{\partial j}\right|$$

$$\max\left(\left|\frac{\partial I}{\partial i}\right|, \left|\frac{\partial I}{\partial j}\right|\right)$$

D

$$\max\left(\left|\frac{\partial I}{\partial i}\right|, \left|\frac{\partial I}{\partial j}\right|\right)$$

$$\left|\frac{\partial I}{\partial j}\right|$$

$$\left|\frac{\partial I}{\partial i}\right|$$

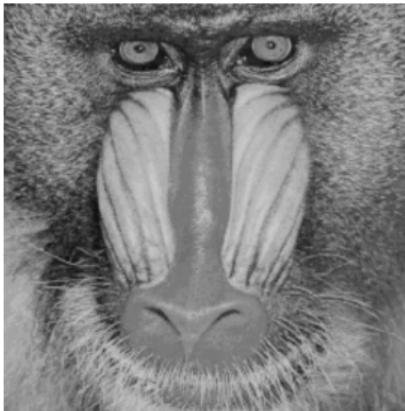
3.2. Opérateurs dérivatifs dédiés à la détection de contours

- Amélioration de la localisation du contour

- ▶ calcul de la différence **autour** du point à dériver

$$\frac{\partial}{\partial i} : \begin{pmatrix} \boxed{-1} \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} -1 \\ \boxed{0} \\ 1 \end{pmatrix}$$

$$\frac{\partial}{\partial j} : \begin{pmatrix} \boxed{-1} & 1 \end{pmatrix} \rightarrow \begin{pmatrix} -1 & \boxed{0} & 1 \end{pmatrix}$$



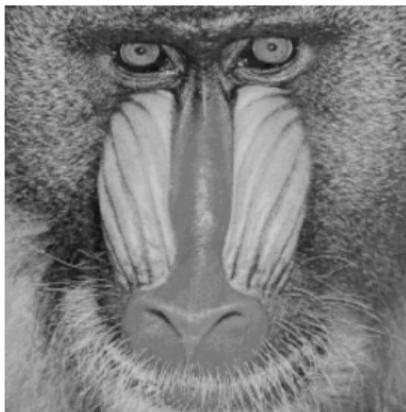
3.2. Opérateurs dérivatifs dédiés à la détection de contours

- Amélioration de la localisation du contour

- ▶ calcul de la différence **autour** du point à dériver

$$\frac{\partial}{\partial i} : \begin{pmatrix} \boxed{-1} \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} -1 \\ \boxed{0} \\ 1 \end{pmatrix}$$

$$\frac{\partial}{\partial j} : \begin{pmatrix} \boxed{-1} & 1 \end{pmatrix} \rightarrow \begin{pmatrix} -1 & \boxed{0} & 1 \end{pmatrix}$$



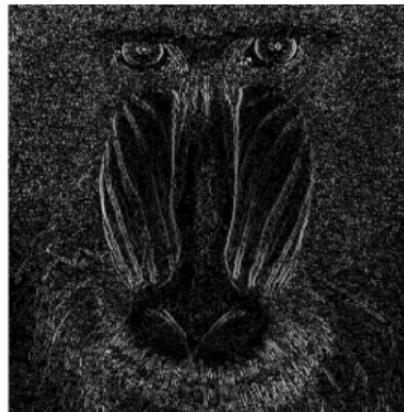
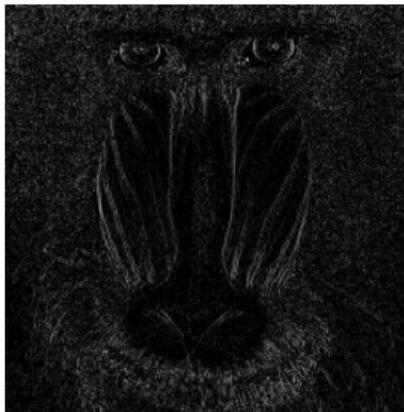
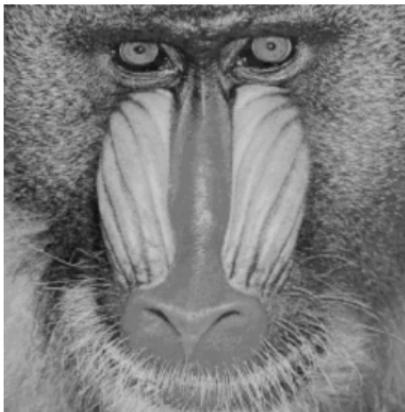
3.2. Opérateurs dérivatifs dédiés à la détection de contours

- Amélioration de la localisation du contour

- ▶ calcul de la différence **autour** du point à dériver

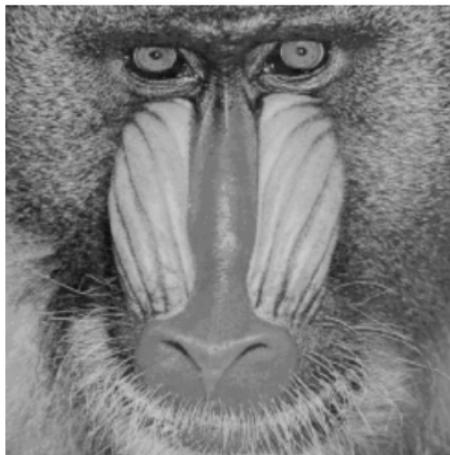
$$\frac{\partial}{\partial i} : \begin{pmatrix} \boxed{-1} \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} -1 \\ \boxed{0} \\ 1 \end{pmatrix}$$

$$\frac{\partial}{\partial j} : \begin{pmatrix} \boxed{-1} & 1 \end{pmatrix} \rightarrow \begin{pmatrix} -1 & \boxed{0} & 1 \end{pmatrix}$$

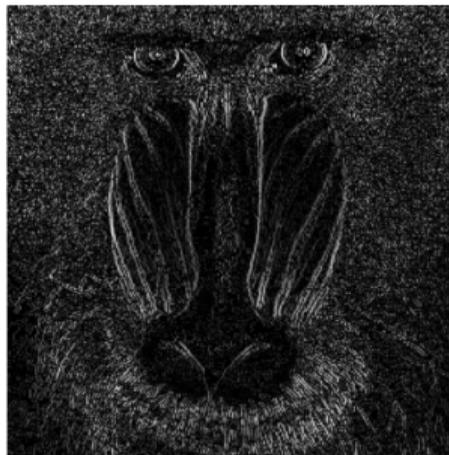


- Réduction de la sensibilité à la texture et au bruit

- ▶ filtrage **passé-bas** avant dérivation



I

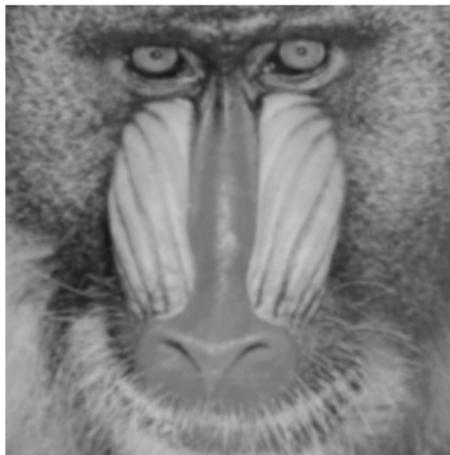


$\frac{\partial I}{\partial j}$

- ▶ réduction des détails et épaississement des contours
- ▶ construction de **dérivateurs contenant le filtrage passé-bas** : convolution d'une dérivation dans une direction avec un lissage dans la direction orthogonale

- Réduction de la sensibilité à la texture et au bruit

- ▶ filtrage **passé-bas** avant dérivation



$$G_{\sigma} * I$$



$$\frac{\partial G_{\sigma} * I}{\partial j}$$

- ▶ réduction des détails et épaissement des contours
- ▶ construction de **dérivateurs contenant le filtrage passé-bas** : convolution d'une dérivation dans une direction avec un lissage dans la direction orthogonale

★ Composition de dérivation et lissage

- ▶ On précalcule la composition (dérivation \circ lissage) en effectuant la convolution des deux opérateurs :

$$\begin{pmatrix} -1 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

- ▶ Quel est l'opérateur résultant ?

A	B	C	D
$\begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$	$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix}$

★ Composition de dérivation et lissage

- ▶ On précalcule la composition (dérivation \circ lissage) en effectuant la convolution des deux opérateurs :

$$\begin{pmatrix} -1 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

- ▶ Quel est l'opérateur résultant ?

A	B	C	D
$\begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$	$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix}$

Attention : on applique le **symétrique** du noyau

o Opérateurs de Prewitt et de Sobel

- ▶ composition entre une dérivation et un lissage dans la direction orthogonale

- ▶ dérivateurs : $\begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$ et $\begin{pmatrix} -1 & 0 & 1 \end{pmatrix}$

- ▶ lissage avec $\begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$: opérateur de Prewitt

$$\frac{\partial}{\partial i} = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \quad \frac{\partial}{\partial j} = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

- ▶ lissage avec $\begin{pmatrix} 1 & 2 & 1 \end{pmatrix}$: opérateur de Sobel

$$\frac{\partial}{\partial i} = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad \frac{\partial}{\partial j} = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

o Opérateurs de Prewitt et de Sobel

- ▶ composition entre une dérivation et un lissage dans la direction orthogonale

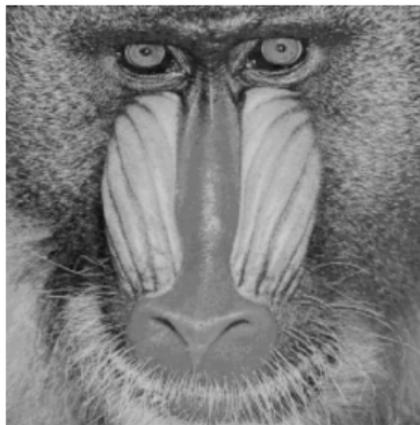
- ▶ dérivateurs : $\begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$ et $\begin{pmatrix} -1 & 0 & 1 \end{pmatrix}$

- ▶ lissage avec $\begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$: opérateur de Prewitt

$$\frac{\partial}{\partial i} = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \quad \frac{\partial}{\partial j} = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

- ▶ lissage avec $\begin{pmatrix} 1 & 2 & 1 \end{pmatrix}$: opérateur de Sobel

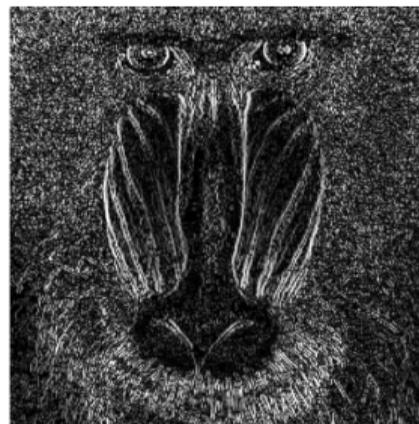
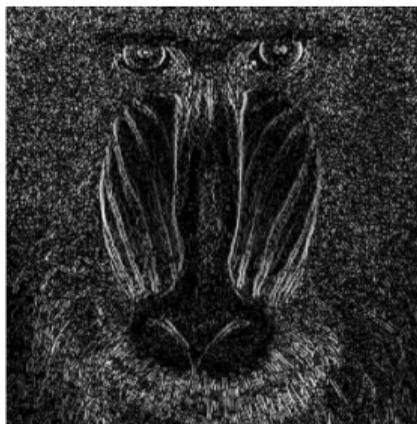
$$\frac{\partial}{\partial i} = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \quad \frac{\partial}{\partial j} = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$



$$\frac{\partial l}{\partial i}$$



$$\frac{\partial l}{\partial j}$$



Prewitt

Sobel

3.3. Filtrage optimal de Canny / Deriche

○ Critères de Canny [1983]

- ▶ contour 1D modélisé par une fonction de Heaviside $H(t)$ noyée dans un bruit gaussien

$$H(t) = \begin{cases} 0 & \text{si } t < 0 \\ 1 & \text{si } t \geq 0 \end{cases}$$



$$I(t) = A \times H(t) + \nu(t)$$

- ▶ trois critères pour évaluer la performance d'un détecteur

1. *détection* : manquer le moins possible de contours
2. *localisation* : aussi proche que possible du contour véritable
3. *unicité* : fournir une seule localisation par contour

- ▶ recherche d'un filtre Φ t.q. le contour corresponde à $\max(\Phi * I)$

- ▶ solution approximable par la dérivée d'une gaussienne : $\begin{pmatrix} -1 & 0 & 1 \end{pmatrix} * G_\sigma$
- ▶ justification formelle des opérateurs de Prewitt et Sobel
- ▶ algorithme de détection de contours de Canny (§5.5)

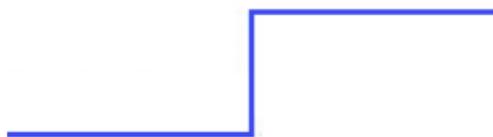
3.3. Filtrage optimal de Canny / Deriche

o Critères de Canny [1983]

- ▶ contour 1D modélisé par une fonction de Heaviside $H(t)$ noyée dans un bruit gaussien

$$H(t) = \begin{cases} 0 & \text{si } t < 0 \\ 1 & \text{si } t \geq 0 \end{cases}$$

$$I(t) = A \times H(t) + \nu(t)$$



- ▶ trois critères pour évaluer la performance d'un détecteur

1. *détection* : manquer le moins possible de contours
2. *localisation* : aussi proche que possible du contour véritable
3. *unicité* : fournir une seule localisation par contour

- ▶ recherche d'un filtre Φ t.q. le contour corresponde à $\max(\Phi * I)$

- ▶ solution approximable par la dérivée d'une gaussienne : $\begin{pmatrix} -1 & 0 & 1 \end{pmatrix} * G_\sigma$
- ▶ justification formelle des opérateurs de Prewitt et Sobel
- ▶ algorithme de détection de contours de Canny (§5.5)

3.3. Filtrage optimal de Canny / Deriche

○ Critères de Canny [1983]

- ▶ contour 1D modélisé par une fonction de Heaviside $H(t)$ noyée dans un bruit gaussien

$$H(t) = \begin{cases} 0 & \text{si } t < 0 \\ 1 & \text{si } t \geq 0 \end{cases}$$

$$I(t) = A \times H(t) + \nu(t)$$



- ▶ trois critères pour évaluer la performance d'un détecteur

1. *détection* : manquer le moins possible de contours
2. *localisation* : aussi proche que possible du contour véritable
3. *unicité* : fournir une seule localisation par contour

- ▶ recherche d'un filtre Φ t.q. le contour corresponde à $\max(\Phi * I)$

- ▶ solution approximable par la dérivée d'une gaussienne : $\begin{pmatrix} -1 & 0 & 1 \end{pmatrix} * G_\sigma$
- ▶ justification formelle des opérateurs de Prewitt et Sobel
- ▶ algorithme de détection de contours de Canny (§5.5)

3.3. Filtrage optimal de Canny / Deriche

o Critères de Canny [1983]

- ▶ contour 1D modélisé par une fonction de Heaviside $H(t)$ noyée dans un bruit gaussien

$$H(t) = \begin{cases} 0 & \text{si } t < 0 \\ 1 & \text{si } t \geq 0 \end{cases}$$

$$I(t) = A \times H(t) + \nu(t)$$



- ▶ trois critères pour évaluer la performance d'un détecteur

1. *détection* : manquer le moins possible de contours
2. *localisation* : aussi proche que possible du contour véritable
3. *unicité* : fournir une seule localisation par contour

- ▶ recherche d'un filtre Φ t.q. le contour corresponde à $\max(\Phi * I)$

- ▶ solution approximable par la dérivée d'une gaussienne : $(-1 \ 0 \ 1) * G_\sigma$
- ▶ justification formelle des opérateurs de Prewitt et Sobel
- ▶ algorithme de détection de contours de Canny (§5.5)

3.3. Filtrage optimal de Canny / Deriche

○ Critères de Canny [1983]

- ▶ contour 1D modélisé par une fonction de Heaviside $H(t)$ noyée dans un bruit gaussien

$$H(t) = \begin{cases} 0 & \text{si } t < 0 \\ 1 & \text{si } t \geq 0 \end{cases}$$

$$I(t) = A \times H(t) + \nu(t)$$



- ▶ trois critères pour évaluer la performance d'un détecteur

1. *détection* : manquer le moins possible de contours
2. *localisation* : aussi proche que possible du contour véritable
3. *unicité* : fournir une seule localisation par contour

- ▶ recherche d'un filtre Φ t.q. le contour corresponde à $\max(\Phi * I)$

- ▶ solution approximable par la dérivée d'une gaussienne : $(-1 \ 0 \ 1) * G_\sigma$
- ▶ justification formelle des opérateurs de Prewitt et Sobel
- ▶ algorithme de détection de contours de Canny (§5.5)

○ Filtre de Deriche [1987]

- ▶ solution exacte à l'équation de Canny
- ▶ opérateurs de dérivation D et de lissage L (intégration de D)

$$D(t) = -c t e^{-\alpha|t|}$$
$$L(t) = k(\alpha|t| + 1)e^{-\alpha|t|}$$

- ▶ opérateurs séparables
- ▶ convolution de la dérivation et du lissage dans des directions orthogonales :

$$\frac{\partial I}{\partial i} = D_i * (L_j * I)$$
$$\frac{\partial I}{\partial j} = D_j * (L_i * I)$$

- ▶ contrôle du lissage au moyen du paramètre α
- ▶ support infini mais implémentation récursive (voir mise en œuvre annexe A)

○ Filtre de Deriche [1987]

- ▶ solution exacte à l'équation de Canny
- ▶ opérateurs de dérivation D et de lissage L (intégration de D)

$$D(t) = -c t e^{-\alpha|t|}$$
$$L(t) = k(\alpha|t| + 1)e^{-\alpha|t|}$$

- ▶ opérateurs séparables
- ▶ convolution de la dérivation et du lissage dans des directions orthogonales :

$$\frac{\partial I}{\partial i} = D_i * (L_j * I)$$
$$\frac{\partial I}{\partial j} = D_j * (L_i * I)$$

- ▶ contrôle du lissage au moyen du paramètre α
- ▶ support infini mais implémentation récursive (voir mise en œuvre annexe A)

○ Filtre de Deriche [1987]

- ▶ solution exacte à l'équation de Canny
- ▶ opérateurs de dérivation D et de lissage L (intégration de D)

$$D(t) = -c t e^{-\alpha|t|}$$
$$L(t) = k(\alpha|t| + 1)e^{-\alpha|t|}$$

- ▶ opérateurs séparables
- ▶ convolution de la dérivation et du lissage dans des directions orthogonales :

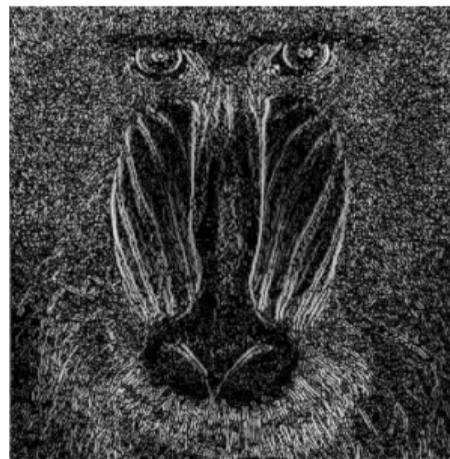
$$\frac{\partial I}{\partial i} = D_i * (L_j * I)$$
$$\frac{\partial I}{\partial j} = D_j * (L_i * I)$$

- ▶ contrôle du lissage au moyen du paramètre α
- ▶ support infini mais implémentation récursive (voir mise en œuvre annexe A)

$$\frac{\partial I}{\partial i}$$



$$\frac{\partial I}{\partial j}$$



Sobel

Deriche avec $\alpha = 1$

Deriche avec $\alpha = 3$

4. Calcul du laplacien discret

4.1. Calcul des dérivées seconde discrètes

- ▶ le laplacien est la somme des dérivées secondes

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

- ▶ calcul des dérivées secondes discrètes

$$\begin{aligned} \frac{\partial}{\partial j} \left(\frac{\partial}{\partial j} I \right) &= \begin{pmatrix} -1 & 1 \end{pmatrix} *_j \left(\begin{pmatrix} -1 & 1 \end{pmatrix} *_j I \right) \\ &= \left(\begin{pmatrix} -1 & 1 \end{pmatrix} *_j \begin{pmatrix} -1 & 1 \end{pmatrix} \right) *_j I \\ &= \begin{pmatrix} 1 & -2 & 1 \end{pmatrix} *_j I \end{aligned}$$

$$\frac{\partial}{\partial i} \left(\frac{\partial}{\partial i} I \right) = \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix} *_i I$$

4. Calcul du laplacien discret

4.1. Calcul des dérivées seconde discrètes

- ▶ le laplacien est la somme des dérivées secondes

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

- ▶ calcul des dérivées secondes discrètes

$$\begin{aligned} \frac{\partial}{\partial j} \left(\frac{\partial}{\partial j} I \right) &= \begin{pmatrix} -1 & 1 \end{pmatrix} *_j \left(\begin{pmatrix} -1 & 1 \end{pmatrix} *_j I \right) \\ &= \left(\begin{pmatrix} -1 & 1 \end{pmatrix} *_j \begin{pmatrix} -1 & 1 \end{pmatrix} \right) *_j I \\ &= \begin{pmatrix} 1 & -2 & 1 \end{pmatrix} *_j I \end{aligned}$$

$$\frac{\partial}{\partial i} \left(\frac{\partial}{\partial i} I \right) = \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix} *_i I$$

4. Calcul du laplacien discret

4.1. Calcul des dérivées seconde discrètes

- ▶ le laplacien est la somme des dérivées secondes

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

- ▶ calcul des dérivées secondes discrètes

$$\begin{aligned} \frac{\partial}{\partial j} \left(\frac{\partial}{\partial j} I \right) &= \begin{pmatrix} -1 & 1 \end{pmatrix} *_j \left(\begin{pmatrix} -1 & 1 \end{pmatrix} *_j I \right) \\ &= \left(\begin{pmatrix} -1 & 1 \end{pmatrix} *_j \begin{pmatrix} -1 & 1 \end{pmatrix} \right) *_j I \\ &= \begin{pmatrix} 1 & -2 & 1 \end{pmatrix} *_j I \end{aligned}$$

$$\frac{\partial}{\partial i} \left(\frac{\partial}{\partial i} I \right) = \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix} *_i I$$

4. Calcul du laplacien discret

4.1. Calcul des dérivées seconde discrètes

- ▶ le laplacien est la somme des dérivées secondes

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

- ▶ calcul des dérivées secondes discrètes

$$\begin{aligned} \frac{\partial}{\partial j} \left(\frac{\partial}{\partial j} I \right) &= \begin{pmatrix} -1 & 1 \end{pmatrix} *_j \left(\begin{pmatrix} -1 & 1 \end{pmatrix} *_j I \right) \\ &= \left(\begin{pmatrix} -1 & 1 \end{pmatrix} *_j \begin{pmatrix} -1 & 1 \end{pmatrix} \right) *_j I \\ &= \begin{pmatrix} 1 & -2 & 1 \end{pmatrix} *_j I \end{aligned}$$

$$\frac{\partial}{\partial i} \left(\frac{\partial}{\partial i} I \right) = \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix} *_i I$$

4.2. Opérateur laplacien discret

- ▶ formulation discrète de l'opérateur laplacien :

$$\Delta I(i,j) = (1 \quad -2 \quad 1) *_j I + \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix} *_i I$$

$$= \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} * I$$

$$= I(j-1, j) + I(j+1, j) + I(i, i-1) + I(i, i+1) - 4I(i, j)$$

- ▶ alternatives

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

4.2. Opérateur laplacien discret

- ▶ formulation discrète de l'opérateur laplacien :

$$\begin{aligned}\Delta I(i,j) &= (1 \quad -2 \quad 1) *_j I + \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix} *_i I \\ &= \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} * I \\ &= I(j-1, j) + I(j+1, j) + I(i, i-1) + I(i, i+1) - 4I(i, j)\end{aligned}$$

- ▶ alternatives

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

4.2. Opérateur laplacien discret

- ▶ formulation discrète de l'opérateur laplacien :

$$\begin{aligned}\Delta I(i,j) &= (1 \quad -2 \quad 1) *_j I + \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix} *_i I \\ &= \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix} * I \\ &= I(j-1, j) + I(j+1, j) + I(i, i-1) + I(i, i+1) - 4I(i, j)\end{aligned}$$

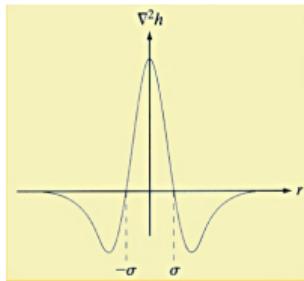
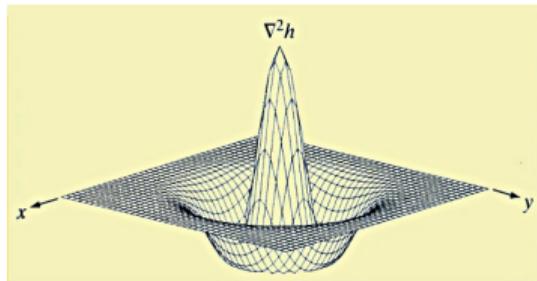
- ▶ alternatives

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

4.3. Combinaison avec un filtrage passe-bas

- ▶ l'opérateur laplacien est très sensible au bruit
- ▶ pour la détection de contours : combinaison avec un filtrage gaussien
- ▶ ΔG_σ : *laplacian of gaussian* (filtre LOG ou chapeau mexicain [Marr et Hildreth, 1980])

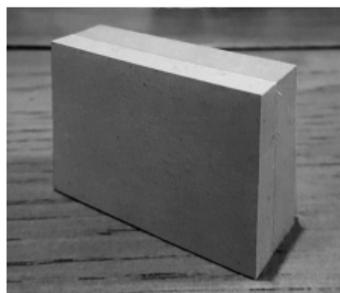
$$\Delta G_\sigma = \frac{\partial^2 G_\sigma}{\partial x^2} + \frac{\partial^2 G_\sigma}{\partial y^2} = 1 - \frac{x^2 + y^2}{2\sigma^2} \exp^{-\frac{x^2+y^2}{2\sigma^2}}$$



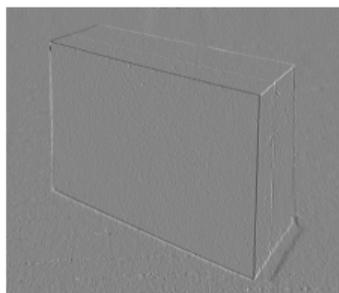
0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

- ▶ approximable par une différence $G_{\sigma_1} - G_{\sigma_2}$ de deux gaussiennes (DOG), avec $\frac{\sigma_2}{\sigma_1} \simeq 1.6$ (voir annexe B)

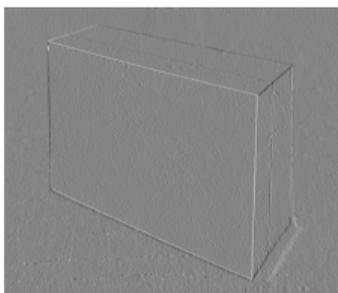
★ Dérivées partielles d'image monochrome



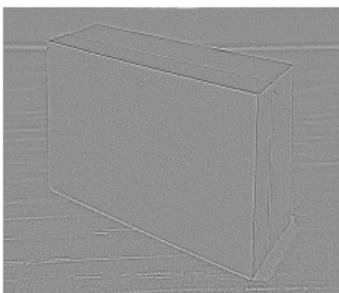
I



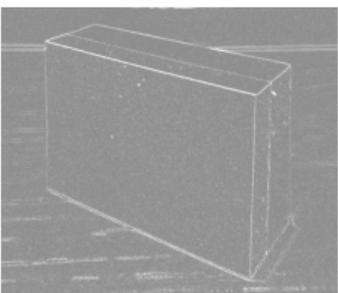
I_1



I_2



I_3

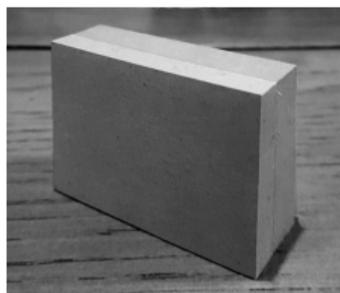


I_4

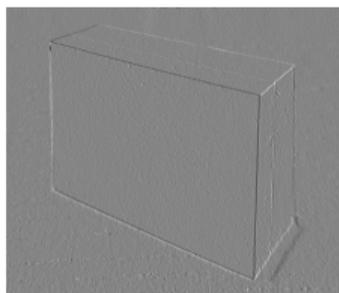
La notation $\frac{\partial I}{\partial j_+}$ (resp. $\frac{\partial I}{\partial j_-}$) représente la dérivée partielle calculée en faisant la différence entre le point suivant (resp. le point précédent) et le point courant.

	I_1	I_2	I_3	I_4
A	$\frac{\partial I}{\partial j_+}$	$\frac{\partial I}{\partial j_-}$	$\ \nabla I\ $	ΔI
B	$\frac{\partial I}{\partial j_+}$	$\frac{\partial I}{\partial j_-}$	ΔI	$\ \nabla I\ $
C	$\frac{\partial I}{\partial j_-}$	$\frac{\partial I}{\partial j_+}$	$\ \nabla I\ $	ΔI
D	$\frac{\partial I}{\partial j_-}$	$\frac{\partial I}{\partial j_+}$	ΔI	$\ \nabla I\ $

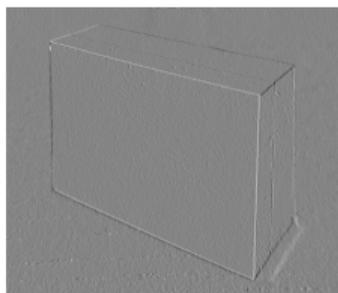
★ Dérivées partielles d'image monochrome



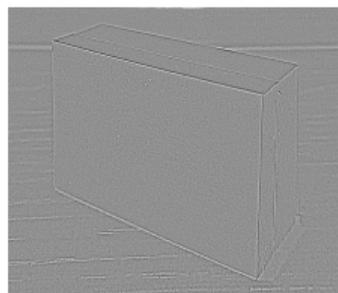
I



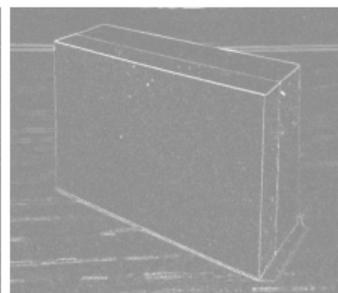
I_1



I_2



I_3



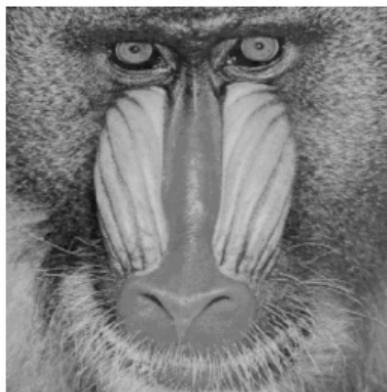
I_4

La notation $\frac{\partial I}{\partial j_+}$ (resp. $\frac{\partial I}{\partial j_-}$) représente la dérivée partielle calculée en faisant la différence entre le point suivant (resp. le point précédent) et le point courant.

	I_1	I_2	I_3	I_4
A	$\frac{\partial I}{\partial j_+}$	$\frac{\partial I}{\partial j_-}$	$\ \nabla I\ $	ΔI
B	$\frac{\partial I}{\partial j_+}$	$\frac{\partial I}{\partial j_-}$	ΔI	$\ \nabla I\ $
C	$\frac{\partial I}{\partial j_-}$	$\frac{\partial I}{\partial j_+}$	$\ \nabla I\ $	ΔI
D	$\frac{\partial I}{\partial j_-}$	$\frac{\partial I}{\partial j_+}$	ΔI	$\ \nabla I\ $

5. De l'image différentielle à l'image de contours

5.1. Localisation des contours



Image



Image différentielle



Image des contours

- ▶ image différentielle : maximums de la norme gradient, zéros du laplacien / LOG
- ▶ image des contours : seuillage

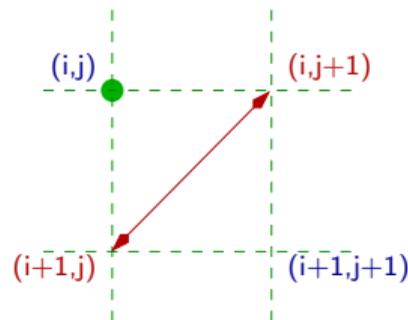
5.2. Méthode historique : croix de Roberts

- ▶ premier opérateur différentiel [1965]
- ▶ calcul d'un gradient à partir de différences finies diagonales :

$$D_1 = I(i, j + 1) - I(i + 1, j)$$

$$D_2 = I(i, j) - I(i + 1, j + 1)$$

- ▶ norme euclidienne ($D = \sqrt{D_1^2 + D_2^2}$)
- ▶ seuillage global



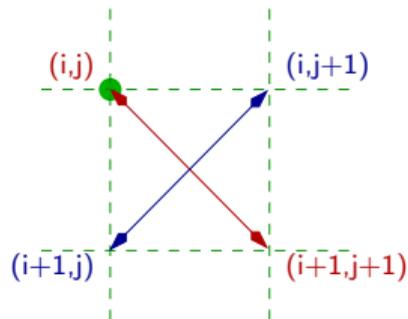
5.2. Méthode historique : croix de Roberts

- ▶ premier opérateur différentiel [1965]
- ▶ calcul d'un gradient à partir de différences finies diagonales :

$$D_1 = I(i, j + 1) - I(i + 1, j)$$

$$D_2 = I(i, j) - I(i + 1, j + 1)$$

- ▶ norme euclidienne ($D = \sqrt{D_1^2 + D_2^2}$)
- ▶ seuillage global



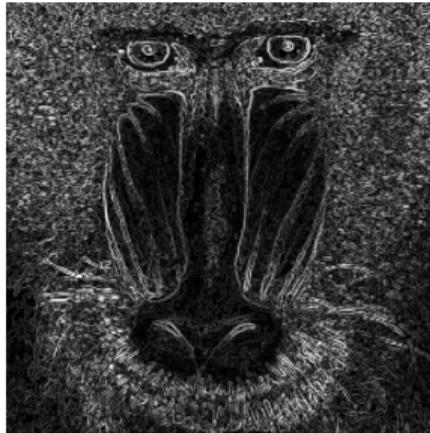
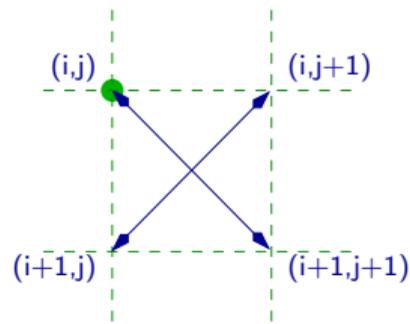
5.2. Méthode historique : croix de Roberts

- ▶ premier opérateur différentiel [1965]
- ▶ calcul d'un gradient à partir de différences finies diagonales :

$$D_1 = I(i, j + 1) - I(i + 1, j)$$

$$D_2 = I(i, j) - I(i + 1, j + 1)$$

- ▶ norme euclidienne ($D = \sqrt{D_1^2 + D_2^2}$)
- ▶ seuillage global



D



$Th_1(D)$



$TH_2(D)$

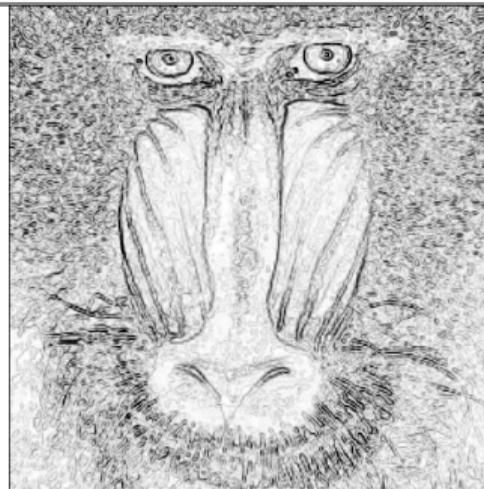
5.3. Maximums de la norme du gradient

○ Calcul de norme du gradient

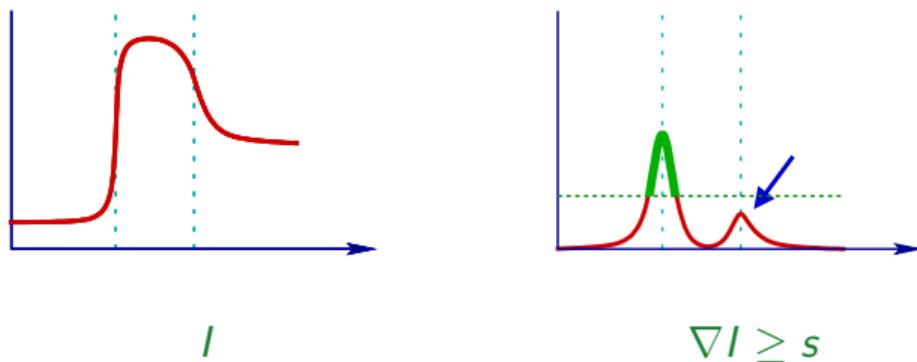
- trois possibilités — définition générale de la norme dans \mathbb{R}^2 : $(|x|^p + |y|^p)^{\frac{1}{p}}$

valeur absolue (<i>norme-1</i>)	euclidienne (<i>norme-2</i>)	maximum (<i>norme-∞</i>)
$ \frac{\partial I}{\partial i} + \frac{\partial I}{\partial j} $	$\sqrt{\frac{\partial I^2}{\partial i^2} + \frac{\partial I^2}{\partial j^2}}$	$\max(\frac{\partial I}{\partial i} , \frac{\partial I}{\partial j})$

exemple (avec inversion)



○ Recherche des maximums par seuillage global



- ▶ très simple à mettre en œuvre, mais compromis entre
 - ▶ le critère de **détection** du maximum possible de contours
 - ▶ le critère d'**unicité** de la localisation par contour
- ▶ seuil élevés : contours plus fins mais contours manquants (perte de maximums locaux)
- ▶ seuil bas : contours épais (+ bruit / texture)

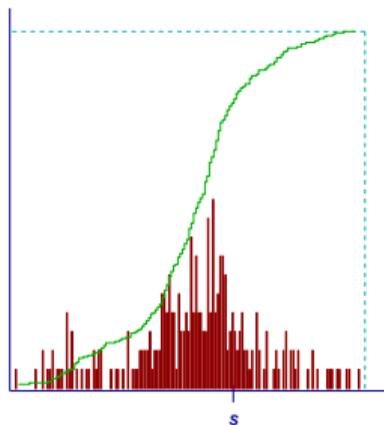
- Plusieurs possibilités pour guider le choix du seuil

- ▶ seuillage relativement à la **moyenne** ou à la **médiane**

$$TH_I(i) = \begin{cases} 0 & \text{si } i < \mu(I) \\ 1 & \text{sinon} \end{cases}$$

- ▶ relativement à un **pourcentage** de valeurs à éliminer : histogramme cumulé

$$\bar{h}(i) = \sum_{j=0}^i h(j)$$



le plus petit s t.q. $\bar{h}(s) > 0.8$

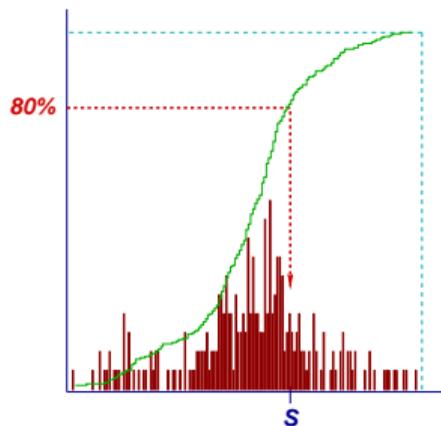
○ Plusieurs possibilités pour guider le choix du seuil

- ▶ seuillage relativement à la **moyenne** ou à la **médiane**

$$TH_I(i) = \begin{cases} 0 & \text{si } i < \mu(I) \\ 1 & \text{sinon} \end{cases}$$

- ▶ relativement à un **pourcentage** de valeurs à éliminer : histogramme cumulé

$$\bar{h}(i) = \sum_{j=0}^i h(j)$$



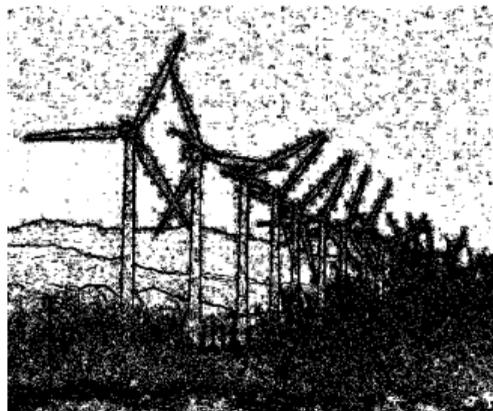
le plus petit s t.q. $\bar{h}(s) > 0.8$

Exemple avec le dérivateur
de Sobel

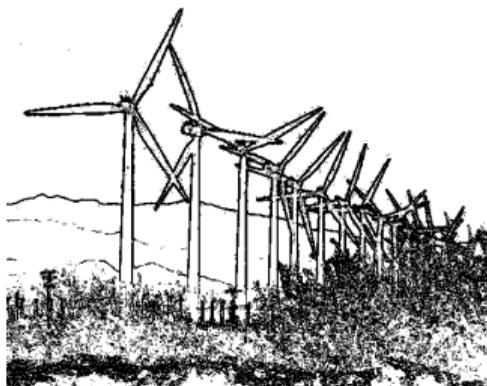


/

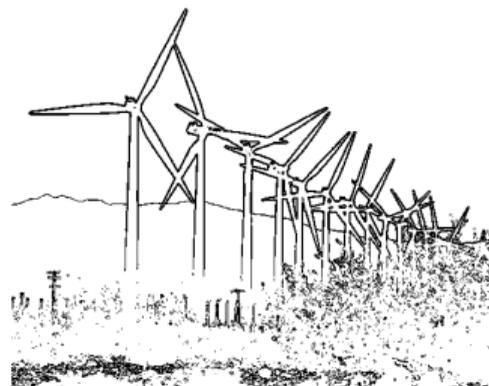
Image (inversée) de $||\nabla I||$



Seuillage à la médiane (50%)



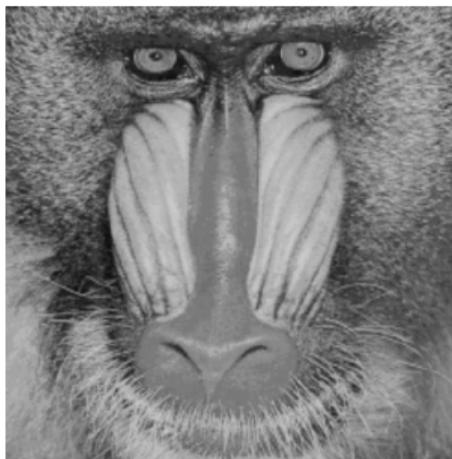
Seuillage à la moyenne



Seuillage à 90%

- ▶ exemple avec les dérivateurs de Sobel et de Deriche

Sobel



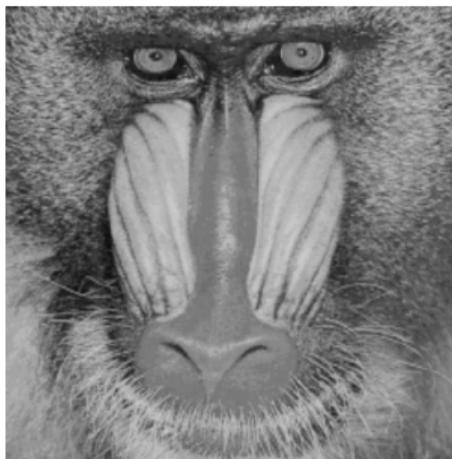
Deriche



norme du gradient

- ▶ exemple avec les dérivateurs de Sobel et de Deriche

Sobel

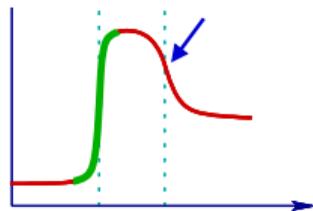
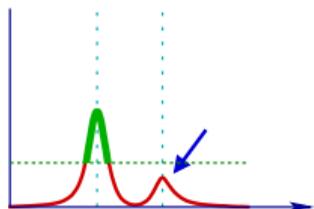


Deriche

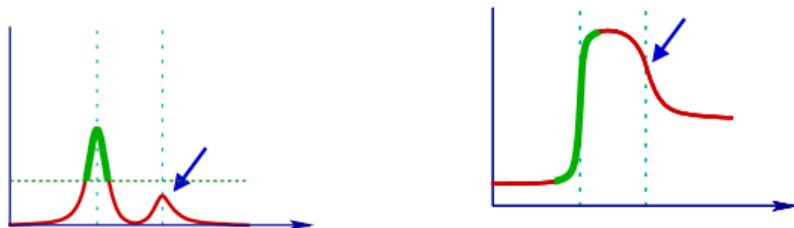


seuillage à la moyenne

- Recherche des maximums locaux

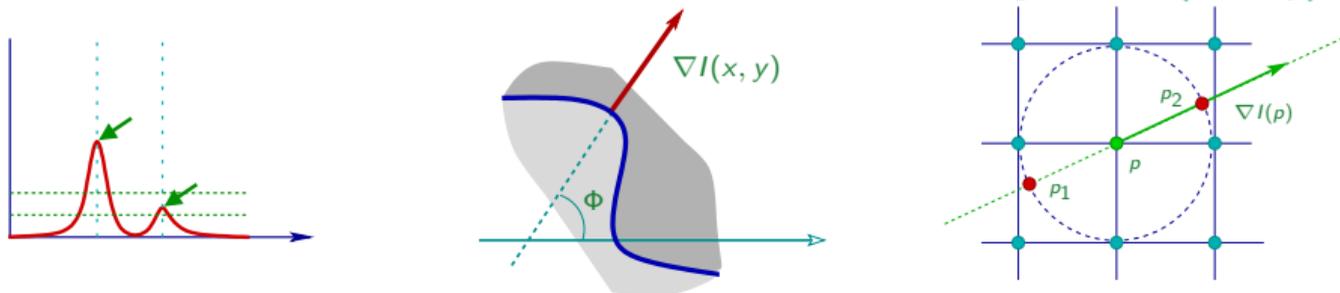


○ Recherche des maximums locaux dans la direction du vecteur gradient (Canny)



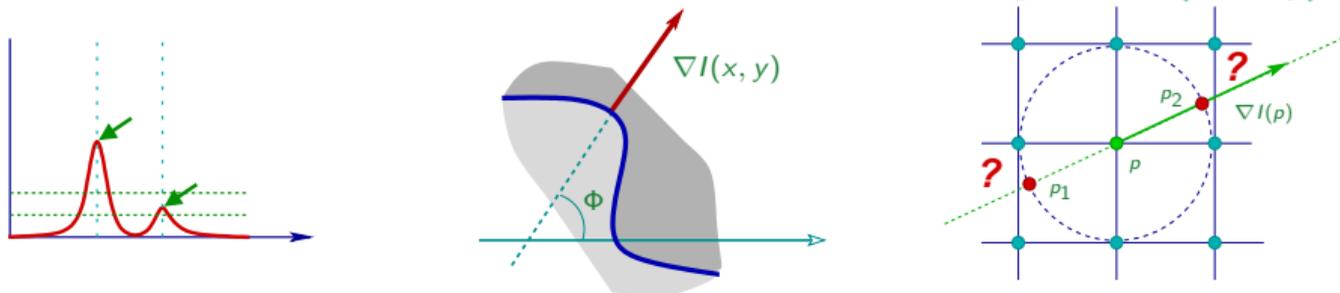
- ▶ p_1 et p_2 : à distance 1 de p dans la direction de $\nabla I(p) = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right) = \left(\frac{\partial I}{\partial j}, -\frac{\partial I}{\partial i}\right)$
- ▶ p est un **maximum local** si $\|\nabla I(p)\| > \|\nabla I(p_1)\|$ et $\|\nabla I(p)\| > \|\nabla I(p_2)\|$
- ▶ estimation de $\|\nabla I(p_1)\|$ et $\|\nabla I(p_2)\|$ par **interpolation** ou **approximation**

○ Recherche des maximums locaux dans la direction du vecteur gradient (Canny)

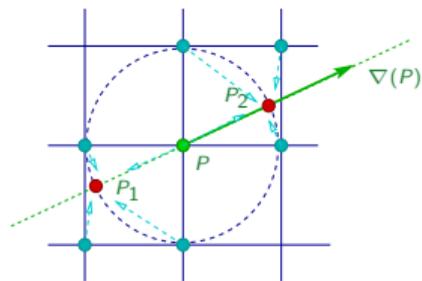


- ▶ p_1 et p_2 : à distance 1 de p dans la direction de $\nabla I(p) = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right) = \left(\frac{\partial I}{\partial j}, -\frac{\partial I}{\partial i} \right)$
- ▶ p est un **maximum local** si $\|\nabla I(p)\| > \|\nabla I(p_1)\|$ et $\|\nabla I(p)\| > \|\nabla I(p_2)\|$
- ▶ estimation de $\|\nabla I(p_1)\|$ et $\|\nabla I(p_2)\|$ par **interpolation** ou **approximation**

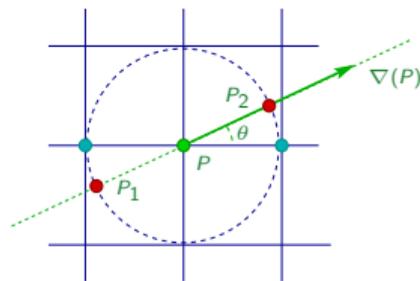
○ Recherche des maximums locaux dans la direction du vecteur gradient (Canny)



- ▶ p_1 et p_2 : à distance 1 de p dans la direction de $\nabla I(p) = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right) = \left(\frac{\partial I}{\partial j}, -\frac{\partial I}{\partial i} \right)$
- ▶ p est un **maximum local** si $\|\nabla I(p)\| > \|\nabla I(p_1)\|$ et $\|\nabla I(p)\| > \|\nabla I(p_2)\|$
- ▶ estimation de $\|\nabla I(p_1)\|$ et $\|\nabla I(p_2)\|$ par **interpolation** ou **approximation**

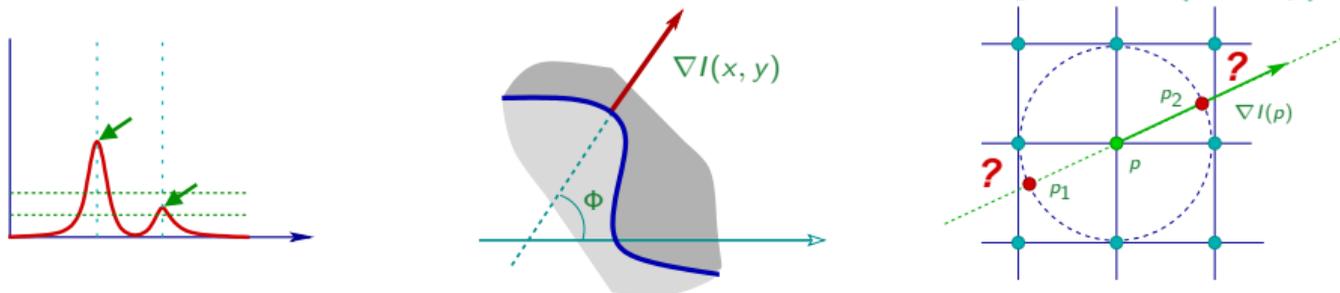


interpolation bilinéaire

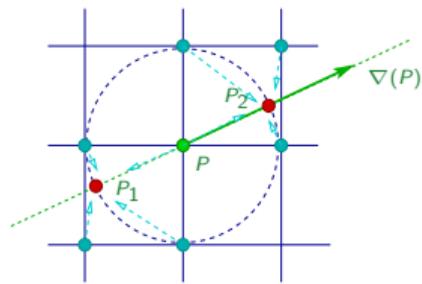


point entier le plus proche

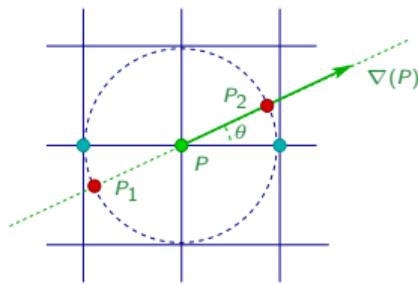
○ Recherche des maximums locaux dans la direction du vecteur gradient (Canny)



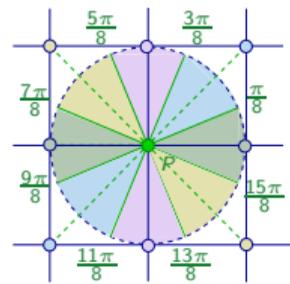
- ▶ p_1 et p_2 : à distance 1 de p dans la direction de $\nabla I(p) = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right) = \left(\frac{\partial I}{\partial j}, -\frac{\partial I}{\partial i} \right)$
- ▶ p est un **maximum local** si $\|\nabla I(p)\| > \|\nabla I(p_1)\|$ et $\|\nabla I(p)\| > \|\nabla I(p_2)\|$
- ▶ estimation de $\|\nabla I(p_1)\|$ et $\|\nabla I(p_2)\|$ par **interpolation** ou **approximation**



interpolation bilinéaire



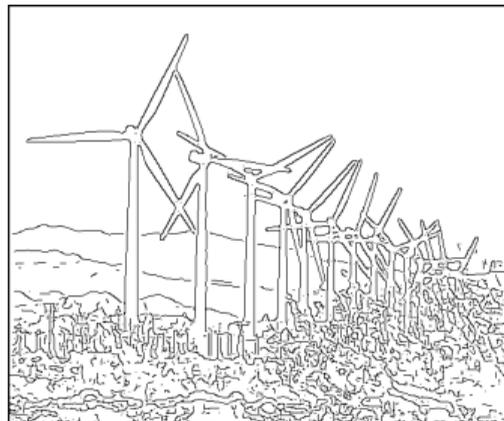
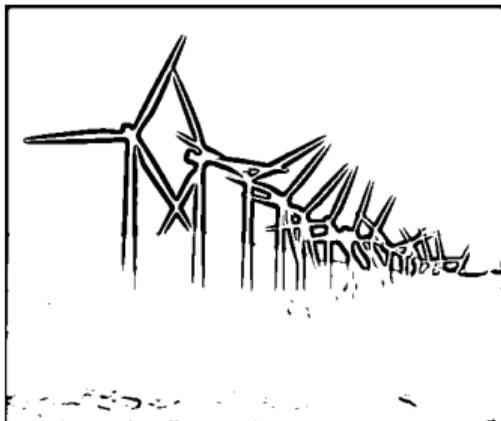
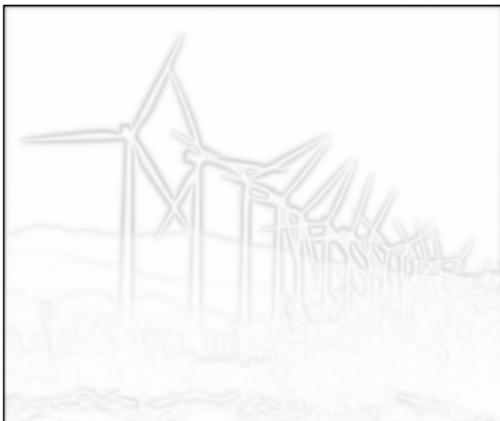
point entier le plus proche



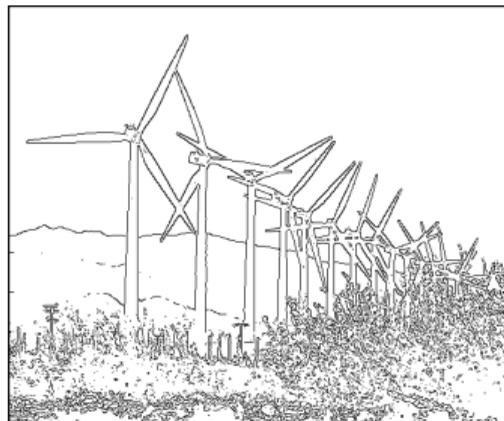
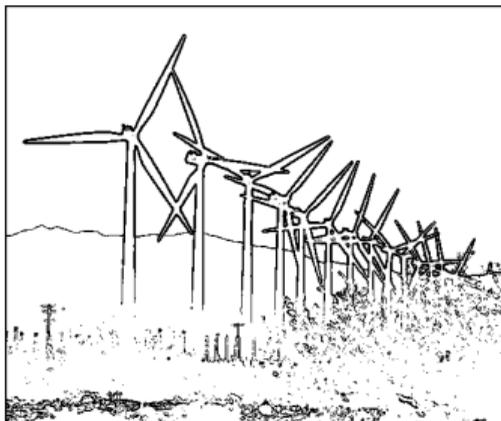
$$\theta = \text{atan2}\left(-\frac{\partial I}{\partial i}, \frac{\partial I}{\partial j}\right)$$

▶ exemple :

$\alpha = 0.7$



$\alpha = 2.2$

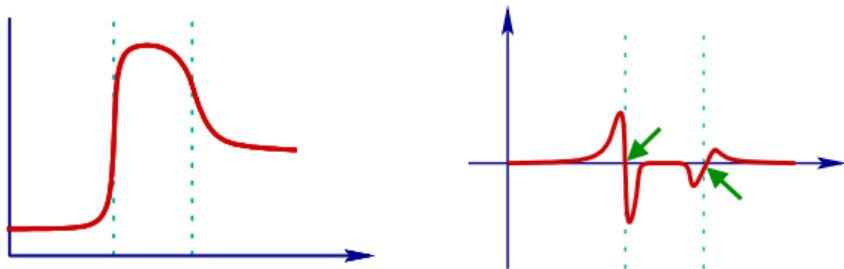


norme du gradient de Deriche

seuillage global à 90%

maxim. locaux seuillés à 90%

5.4. Passages par zéro du laplacien

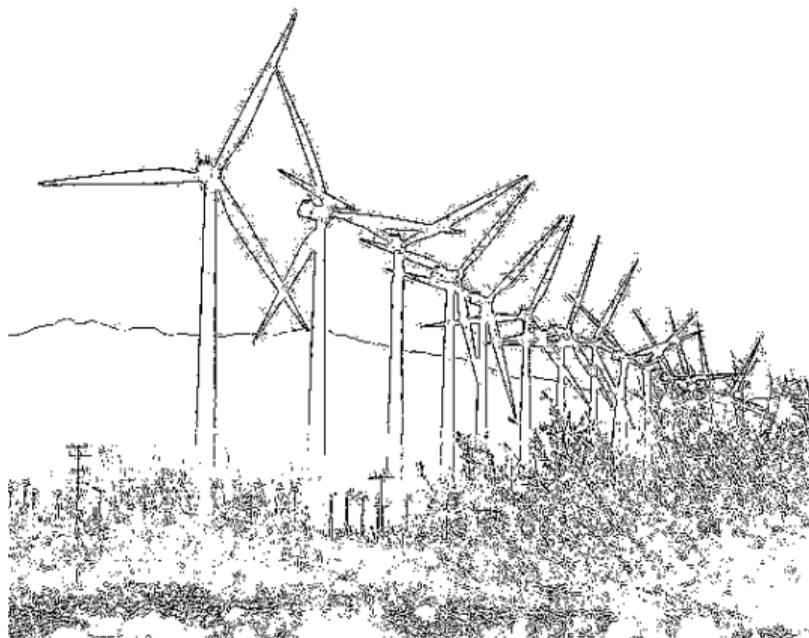


- ▶ un zéro correspond à un changement de signe horizontal ou vertical :

$$\begin{array}{cccccccc} \cdot & \overset{+}{\square} & \cdot & + & \overset{\cdot}{\square} & \cdot & \cdot & \overset{\cdot}{\square} & + & \cdot & \overset{\cdot}{\square} & \cdot & \cdot & \overset{-}{\square} & \cdot & - & \overset{\cdot}{\square} & \cdot & \cdot & \overset{\cdot}{\square} & - & \cdot & \overset{\cdot}{\square} & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \end{array}$$

- ▶ possible de tester si le signe du produit est négatif
- ▶ très sensible au bruit → LOG / DOG + seuillage

► exemple :



Seuillage des zéros du laplacien à 90%



Seuillage des zéros du LOG à 90%

5.5. Algorithme de Canny

○ Étapes de l'algorithme

- ▶ solution à l'optimisation des trois critères : détection, localisation et unicité
- ▶ approximation de l'opérateur optimal par une dérivée de gaussienne (localisation)
 1. *lissage de l'image avec un filtre gaussien*
 2. *calcul des dérivées partielles de l'image avec*

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

- ▶ amélioration : utilisation de l'opérateur optimal de Deriche
- ▶ localisation des points de contours
 3. *sélection des maximums locaux (unicité)*
 4. *seuillage par hystérésis (détection)*

○ **Seuillage par hystérésis**

▶ compromis entre

- ▶ *conservation des contours (seuil bas)*
- ▶ *suppression du bruit (seuil haut)*

▶ utilisation conjointe de **deux niveaux** de seuil

S_H : *seuil haut*

S_B : *seuil bas*

un pixel P est retenu si

$$i_P \geq S_H$$

ou $S_H \geq i_P \geq S_B$, et P est connexe à un pixel retenu

▶ mise en œuvre

1. *sélection des points $\geq S_H$*
2. *répéter jusqu'à convergence (éventuellement, un nombre max de fois) :*
rajout des points $\geq S_B$ adjacents à un point retenu

▶ utilisable pour n'importe quel seuillage

○ Seuillage par hystérésis

▶ compromis entre

- ▶ *conservation des contours (seuil bas)*
- ▶ *suppression du bruit (seuil haut)*

▶ utilisation conjointe de deux niveaux de seuil

S_H : seuil haut

S_B : seuil bas

un pixel P est retenu si

$$i_P \geq S_H$$

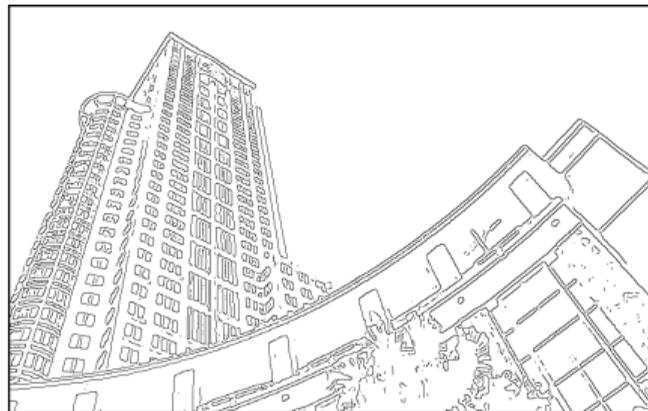
ou $S_H \geq i_P \geq S_B$, et P est connexe à un pixel retenu

▶ mise en œuvre

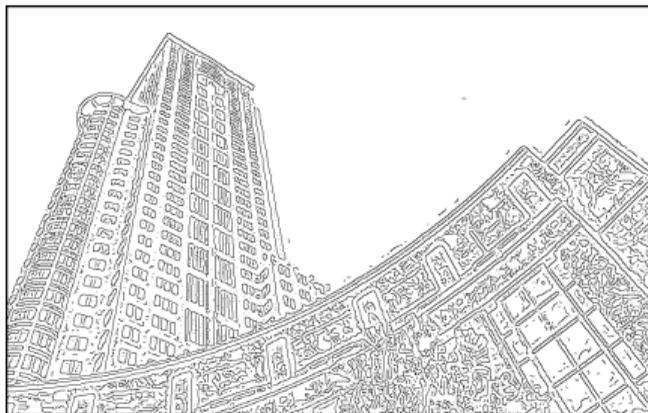
1. *sélection des points $\geq S_H$*
2. *répéter jusqu'à convergence (éventuellement. un nombre max de fois) :*
rajout des points $\geq S_B$ adjacents à un point retenu

▶ utilisable pour n'importe quel seuillage

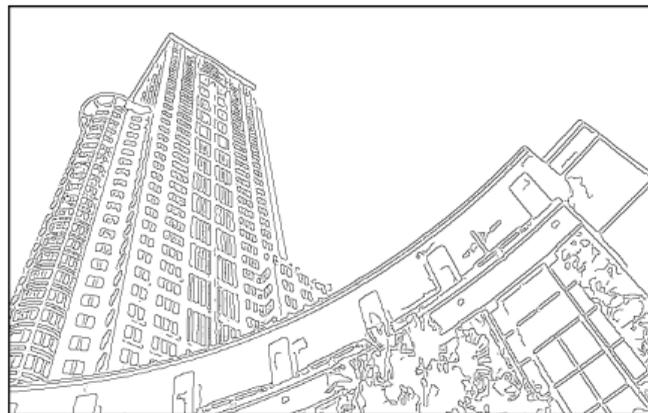
► seuillage par hystérésis des maximums locaux d'un gradient de Deriche ($\alpha = 0.9$)



seuil haut

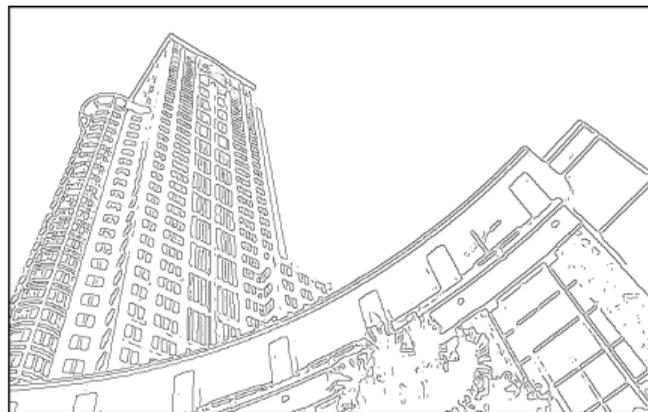


seuil bas

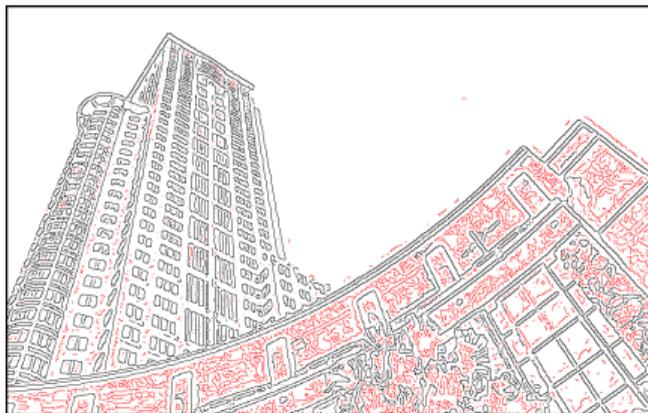


hystérésis

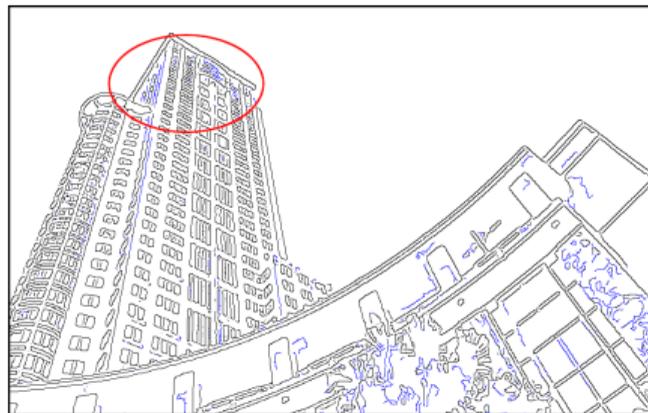
► seuillage par hystérésis des maximums locaux d'un gradient de Deriche ($\alpha = 0.9$)



seuil haut

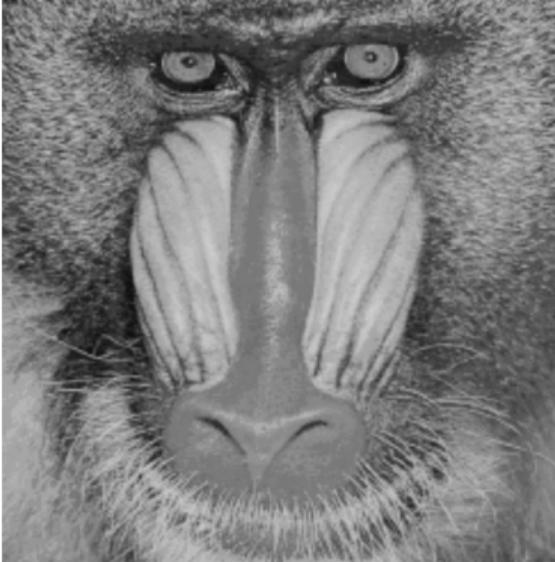


seuil bas



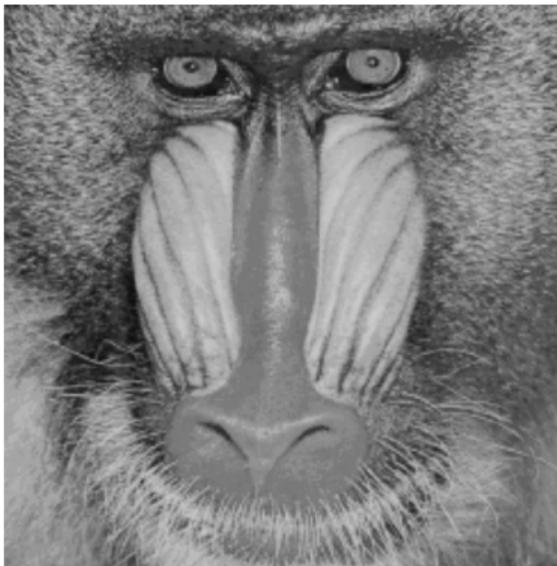
hystérésis

- Exemple de résultats de l'algorithme de Canny
 - ▶ avec l'opérateur optimal de Deriche



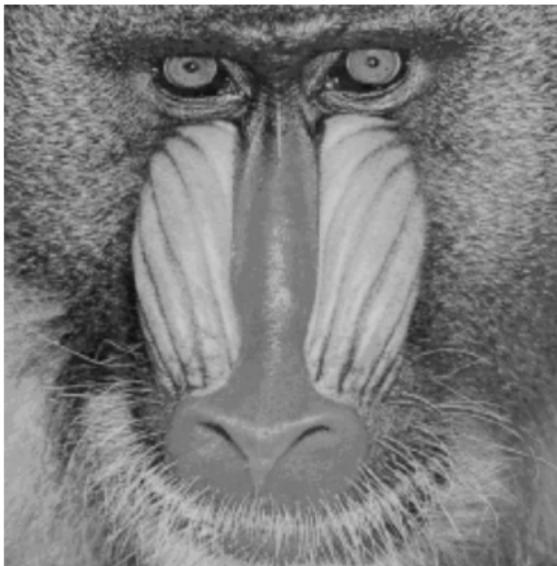
$\alpha = 3.0$
 $T_h = 90$
 $T_l = 40$

- Exemple de résultats de l'algorithme de Canny
 - ▶ avec l'opérateur optimal de Deriche



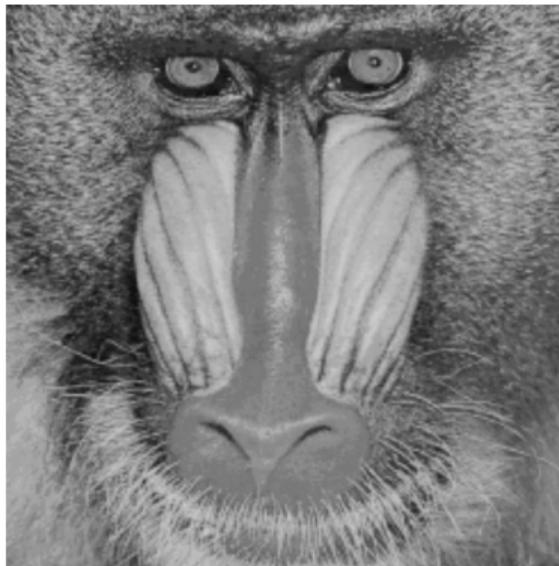
$$\begin{aligned}\alpha &= 1.4 \\ T_h &= 60 \\ T_l &= 30\end{aligned}$$

- Exemple de résultats de l'algorithme de Canny
 - ▶ avec l'opérateur optimal de Deriche



$$\begin{aligned}\alpha &= 1.0 \\ T_h &= 60 \\ T_l &= 8\end{aligned}$$

- Exemple de résultats de l'algorithme de Canny
 - ▶ avec l'opérateur optimal de Deriche



$$\begin{aligned}\alpha &= 0.5 \\ T_h &= 20 \\ T_l &= 4\end{aligned}$$

► méthode de Canny standard



Image convertie en monochrome et normalisée entre 0 et 1



$$\begin{aligned}\sigma &= 0.5 \\ T_h &= 0.17 \\ T_l &= 0.002\end{aligned}$$



$$\begin{aligned}\sigma &= 2.5 \\ T_h &= 0.0214 \\ T_l &= 10^{-5}\end{aligned}$$

ANNEXE A

Implémentation récursive du filtre de Deriche

- ▶ filtre de Deriche :

$$D(t) = -c t e^{-\alpha|t|}$$

$$L(t) = k(\alpha|t| + 1)e^{-\alpha|t|}$$

- ▶ implémentation récursive du filtre de dérivation D

$$I^+(t) = I(t-1) + 2e^{-\alpha}I^+(t-1) - e^{-2\alpha}I^+(t-2)$$

$$I^-(t) = I(t+1) + 2e^{-\alpha}I^-(t+1) - e^{-2\alpha}I^-(t+2)$$

$$I^D(t) = (1 - e^{-\alpha})^2 (I^+(t) - I^-(t))$$

- ▶ implémentation récursive filtre de lissage L

$$I^+(t) = a_0I(t) + a_1I(t-1) + b_1I^+(t-1) + b_2I^+(t-2)$$

$$I^-(t) = a_2I(t+1) + a_3I(t+2) + b_1I^-(t+1) + b_2I^-(t+2)$$

$$I^L(t) = I^-(t) + I^+(t)$$

avec :

$$\left\{ \begin{array}{l} k = \frac{(1-e^{-\alpha})^2}{1+2\alpha e^{-\alpha}-e^{-2\alpha}} \\ b_1 = 2e^{-\alpha} \\ b_2 = -e^{-2\alpha} \end{array} \right. \quad \left\{ \begin{array}{l} a_0 = k \\ a_1 = k(\alpha-1)e^{-\alpha} \\ a_2 = k(\alpha+1)e^{-\alpha} \\ a_3 = -ke^{-2\alpha} \end{array} \right.$$

- ▶ filtre de Deriche :

$$D(t) = -c t e^{-\alpha|t|}$$

$$L(t) = k(\alpha|t| + 1)e^{-\alpha|t|}$$

- ▶ implémentation récursive du filtre de dérivation D

$$I^+(t) = I(t-1) + 2e^{-\alpha}I^+(t-1) - e^{-2\alpha}I^+(t-2)$$

$$I^-(t) = I(t+1) + 2e^{-\alpha}I^-(t+1) - e^{-2\alpha}I^-(t+2)$$

$$I^D(t) = (1 - e^{-\alpha})^2 (I^+(t) - I^-(t))$$

- ▶ implémentation récursive filtre de lissage L

$$I^+(t) = a_0I(t) + a_1I(t-1) + b_1I^+(t-1) + b_2I^+(t-2)$$

$$I^-(t) = a_2I(t+1) + a_3I(t+2) + b_1I^-(t+1) + b_2I^-(t+2)$$

$$I^L(t) = I^-(t) + I^+(t)$$

avec :

$$\left\{ \begin{array}{l} k = \frac{(1-e^{-\alpha})^2}{1+2\alpha e^{-\alpha}-e^{-2\alpha}} \\ b_1 = 2e^{-\alpha} \\ b_2 = -e^{-2\alpha} \end{array} \right. \quad \left\{ \begin{array}{l} a_0 = k \\ a_1 = k(\alpha - 1)e^{-\alpha} \\ a_2 = k(\alpha + 1)e^{-\alpha} \\ a_3 = -ke^{-2\alpha} \end{array} \right.$$

- ▶ application de ces filtres dans les deux directions de l'image
- ▶ par exemple, lissage en ligne :

$$I^+(i,j) = a_0 I(i,j) + a_1 I(i,j-1) + b_1 I^+(i,j-1) + b_2 I^+(i,j-2)$$

$$I^-(i,j) = a_2 I(i,j+1) + a_3 I(i,j+2) + b_1 I^-(i,j+1) + b_2 I^-(i,j+2)$$

$$L_i(i,j) = I^-(i,j) + I^+(i,j)$$

ANNEXE B

Approximation du Laplacien de gaussienne (LOG)

- **Approximation du LOG par différence de gaussiennes (DOG)**

- ▶ utilisation de deux gaussiennes de variance proche :

$$\frac{\sigma}{\sigma'}$$
 de l'ordre de 1.6

- ▶ exemple avec

- ▶ $\sigma = 1$

$$g_1 = \frac{1}{2\pi} \exp\left(\frac{-x^2}{2}\right)$$

- ▶ $\sigma = 0.6$:

$$g_2 = \frac{1}{2\pi \times 0.6^2} \exp\left(\frac{-x^2}{2 \times 0.6^2}\right)$$

- ▶ $g = g_2 - g_1$

- ▶ bénéficie de l'impémentation **réursive** du filtre gaussien

