

TI – Traitement d'Images

Semaine 10 : Détection de contours (2)

Olivier Losson

Master Informatique : <http://www.fil.univ-lille1.fr>
Spécialité IVI : <http://master-ivi.univ-lille1.fr>

Plan du cours

- **1 – Approches par dérivées secondes**
 - Justification de l'approche par dérivées secondes
 - Dérivée seconde directionnelle et Laplacien
 - Approximations discrètes
- **2 – Utilisation du Laplacien**
 - Réduction de la sensibilité au bruit
 - Comparaison des approches de 1^{er} et 2^{ème} ordre
 - Détection de contours multi-échelles
- **3 – Post-traitements sur les contours**
 - Généralités
 - Fermeture de contour
 - Codage
- **Sélection de références**

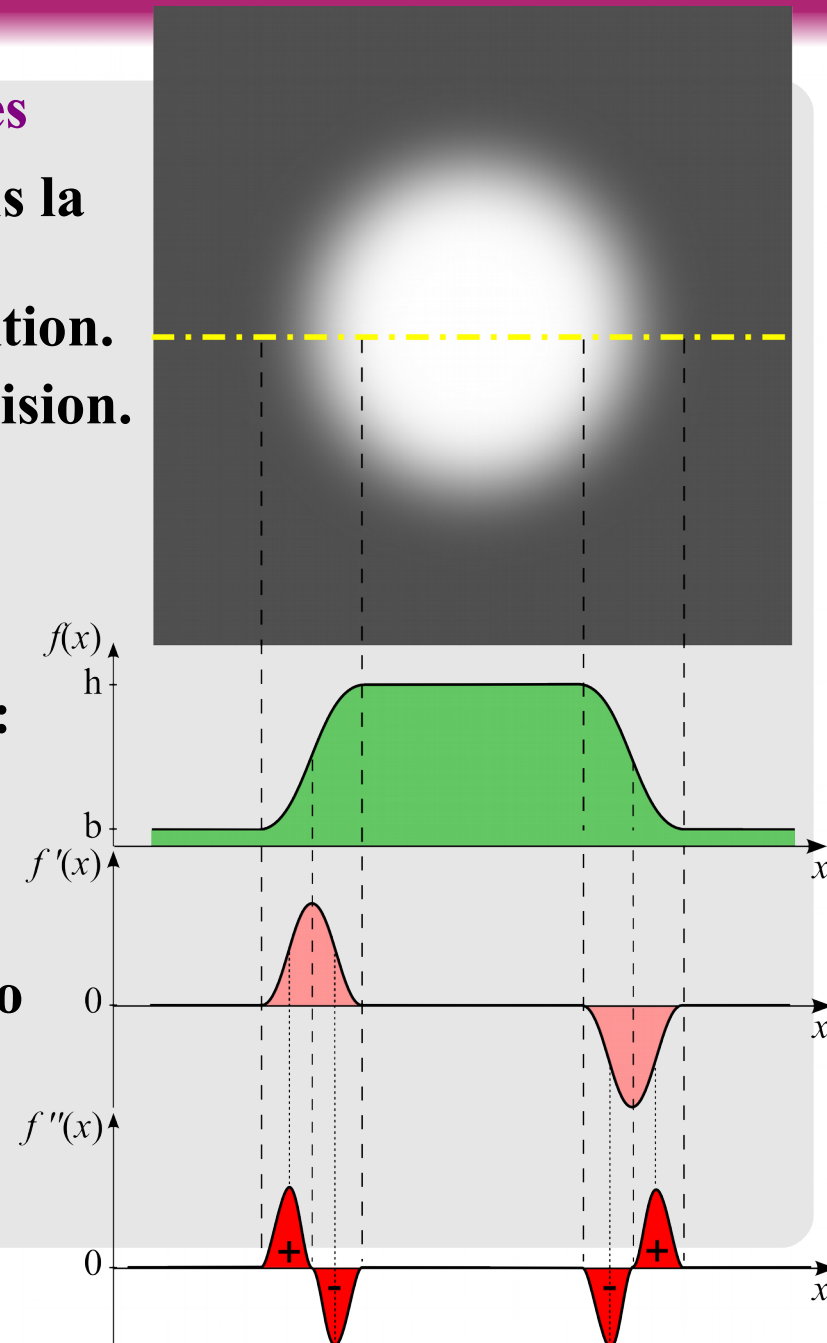
Justification de l'approche par dérivées secondes

- **Critique des approches par dérivées premières**

- Nécessitent une détection des maxima dans la direction du gradient, car l'épaisseur du « contour » dépend de la largeur de transition.
- Difficulté de localiser le contour avec précision.

- **Utilisation de la dérivée seconde**

- La dérivée seconde d'une fonction mesure sa courbure locale.
- Utilisation dans la détection des contours :
 - Aux points contours, la dérivée seconde est nulle.
 - Plus précisément, les points contours sont caractérisés par un **passage par zéro** (*ang.* « zero crossing ») de la dérivée seconde.



Dérivée seconde directionnelle et Laplacien (1/4)

• Calcul de la dérivée seconde directionnelle

→ Dérivée première directionnelle dans la direction α (*rappel*) :

$$f'_{\alpha} := \frac{df}{d\alpha} \approx \cos \alpha \frac{\partial f}{\partial x} + \sin \alpha \frac{\partial f}{\partial y} = \vec{\alpha} \cdot \vec{\nabla} f$$

→ Dérivée seconde directionnelle dans la direction α :

$$f''_{\alpha}(x, y) := \frac{d^2 f}{d\alpha^2}(x, y) := \lim_{h \rightarrow 0} \frac{f'_{\alpha}(x + h \cos \alpha, y + h \sin \alpha) - f'_{\alpha}(x, y)}{h}$$

Or, en appliquant la formule de Taylor,

$$f'_{\alpha}(x + h \cos \alpha, y + h \sin \alpha) = f'_{\alpha}(x, y) + h \cos \alpha \frac{\partial f'_{\alpha}}{\partial x}(x, y) + h \sin \alpha \frac{\partial f'_{\alpha}}{\partial y}(x, y) + O(h^2)$$

D'où

$$\begin{aligned} \frac{d^2 f}{d\alpha^2} &\approx \cos \alpha \frac{\partial}{\partial x} \left(\cos \alpha \frac{\partial f}{\partial x} + \sin \alpha \frac{\partial f}{\partial y} \right) + \sin \alpha \frac{\partial}{\partial y} \left(\cos \alpha \frac{\partial f}{\partial x} + \sin \alpha \frac{\partial f}{\partial y} \right) \\ &\approx \cos^2 \alpha \frac{\partial^2 f}{\partial x^2} + 2 \cos \alpha \sin \alpha \frac{\partial^2 f}{\partial x \partial y} + \sin^2 \alpha \frac{\partial^2 f}{\partial y^2} \end{aligned}$$

Dérivée seconde directionnelle et Laplacien (2/4)

- **Détection des points contours**

- On cherche les passages par zéro de la dérivée seconde directionnelle, soit α_0 tel que :
$$\cos^2 \alpha_0 \frac{\partial^2 f}{\partial x^2} + 2 \cos \alpha_0 \sin \alpha_0 \frac{\partial^2 f}{\partial x \partial y} + \sin^2 \alpha_0 \frac{\partial^2 f}{\partial y^2} = 0$$

- **Définition du Laplacien**

- Pour simplifier l'évaluation de la concavité locale de la fonction image f , exprimée par sa dérivée seconde, on fait appel à l'opérateur **Laplacien**, défini par :

$$\Delta f := \frac{\partial^2 f}{\partial \alpha^2} + \frac{\partial^2 f}{\partial \alpha_{\perp}^2}$$

où α et α_{\perp} sont 2 directions orthogonales quelconques.

- *Remarque* : le Laplacien est un opérateur **scalaire** et **isotrope** (invariant par rotation)

Dérivée seconde directionnelle et Laplacien (3/4)

Utilisation du Laplacien pour la détection des points contours

- Les passages par zéro de la dérivée seconde directionnelle f''_{α} coïncident généralement avec ceux du Laplacien ($\Delta f = 0$).

En particulier, avec $\alpha \equiv \text{Ox}$ et $\alpha_{\perp} \equiv \text{Oy}$: $P(x_0, y_0)$ est point contour si

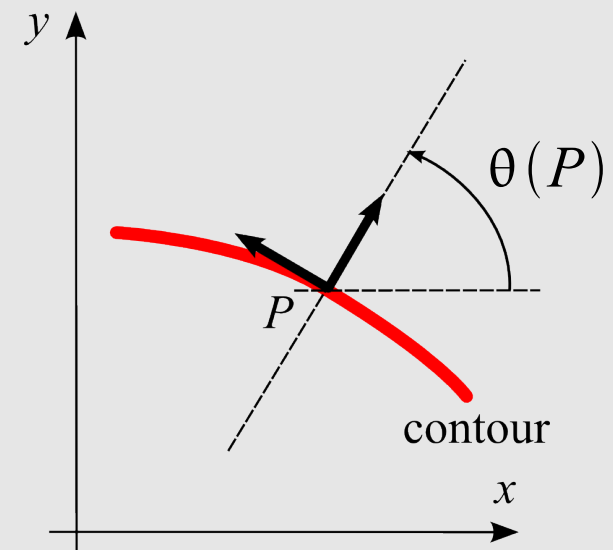
$$\Delta f(x_0, y_0) := \frac{\partial^2 f}{\partial x^2}(x_0, y_0) + \frac{\partial^2 f}{\partial y^2}(x_0, y_0) = 0$$

→ Correspondance avec le gradient :

- Cas particulier où $\alpha \equiv \theta$, direction du gradient (normale au contour) et $\alpha_{\perp} \equiv \theta_{\perp}$ (tangente).

$$\Delta f := \frac{\partial^2 f}{\partial \theta^2} + \frac{\partial^2 f}{\partial \theta_{\perp}^2}$$

- Dans les zones de faible courbure, $\frac{\partial^2 f}{\partial \theta_{\perp}^2} \approx 0$.
- Un passage par zéro du Laplacien correspond donc à une dérivée seconde directionnelle nulle dans la direction du gradient.

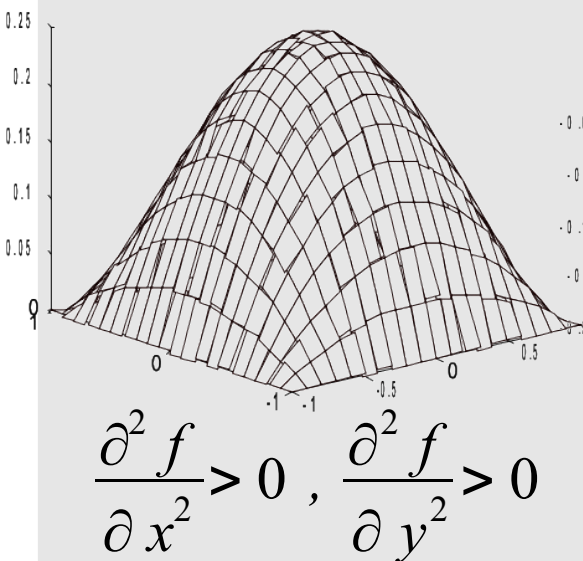


Dérivée seconde directionnelle et Laplacien (4/4)

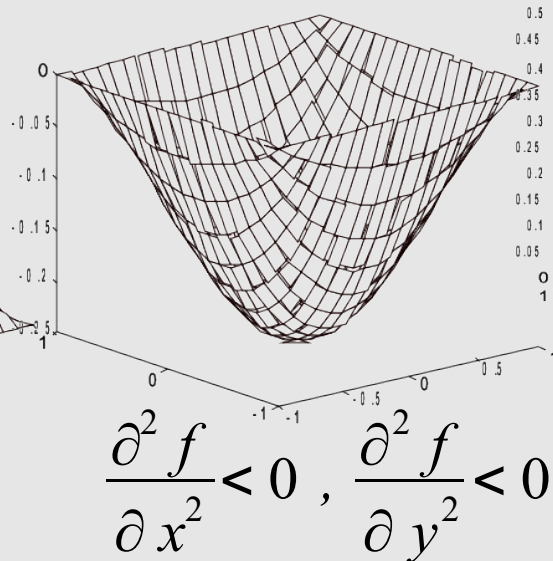
Interprétation du Laplacien.

- Pour $\alpha \equiv \text{Ox}$ et $\alpha_{\perp} \equiv \text{Oy}$, le Laplacien s'écrit $\Delta f = \vec{\nabla} f \cdot \vec{\nabla} f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$
- Sauf dans le cas d'un « point de selle », le Laplacien représente fidèlement la **concavité** de la surface au point considéré :

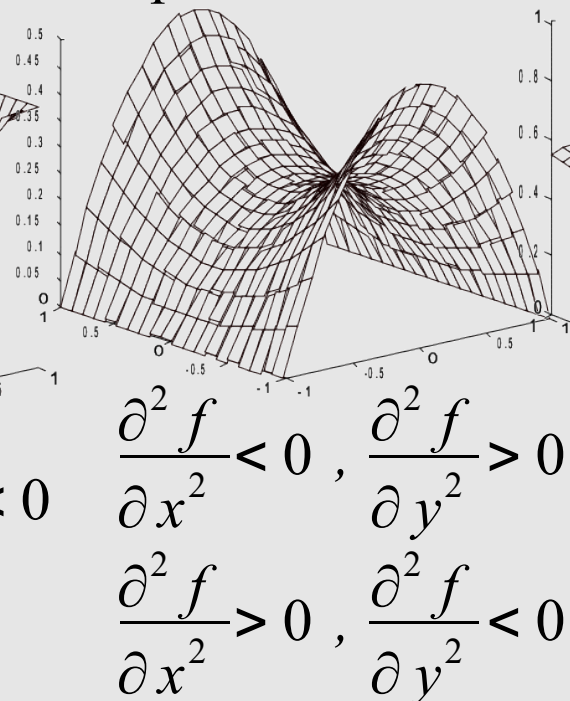
concavité positive



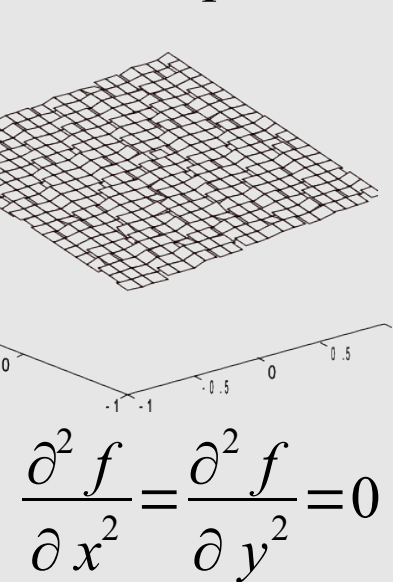
concavité négative



point de selle



surface plane



source : P. Bonnet

Approximations discrètes (1/2)

• Dérivées premières discrètes (*rappel*)

→ Masques de Roberts

$$\frac{\partial f}{\partial x}(x_0, y_0) \approx f(x_0+1, y_0) - f(x_0, y_0)$$

$$\begin{bmatrix} +1 & -1 \end{bmatrix}$$

ou $\approx f(x_0, y_0) - f(x_0-1, y_0)$

$$\begin{bmatrix} +1 & -1 \end{bmatrix}$$

→ Masques de Roberts2

$$\frac{\partial f}{\partial x}(x_0, y_0) \approx \frac{f(x_0+1, y_0) - f(x_0-1, y_0)}{2}$$

$$\frac{1}{2} \begin{bmatrix} +1 & 0 & -1 \end{bmatrix}$$

• Dérivées secondes discrètes

→ Meilleure approximation au centre : appliquer

$$\begin{bmatrix} +1 & -1 \end{bmatrix}, \text{ puis } \begin{bmatrix} +1 & -1 \end{bmatrix}$$

$$\frac{\partial^2 f}{\partial x^2}(x_0, y_0) \approx \frac{\partial f}{\partial x}(x_0+1, y_0) - \frac{\partial f}{\partial x}(x_0, y_0)$$

$$\approx [f(x_0+1, y_0) - f(x_0, y_0)] - [f(x_0, y_0) - f(x_0-1, y_0)]$$

$$\approx f(x_0+1, y_0) - 2f(x_0, y_0) + f(x_0-1, y_0)$$

→ Masque associé à la dérivée seconde selon x :

$$\begin{bmatrix} +1 & -2 & +1 \end{bmatrix}$$

Approximations discrètes (2/2)

• Dérivées secondes discrètes (suite)

→ De même, $\frac{\partial^2 f}{\partial y^2}(x_0, y_0) \approx f(x_0, y_0 + 1) - 2f(x_0, y_0) + f(x_0, y_0 - 1)$

→ D'où l'approximation discrète du Laplacien :

$$\Delta f(x_0, y_0) \approx f(x_0 + 1, y_0) + f(x_0 - 1, y_0) + f(x_0, y_0 + 1) + f(x_0, y_0 - 1) - 4f(x_0, y_0)$$

masque de convolution :

0	+1	0
+1	-4	+1
0	+1	0

+1
-2
+1

→ Autres approximations possibles du Laplacien :

+1	0	+1
0	-4	0
+1	0	+1

+1	+1	+1
+1	-8	+1
+1	+1	+1

+1	+2	+1
+2	-12	+2
+1	+2	+1

+1	+4	+1
+4	-20	+4
+1	+4	+1

Introduction

• Prérequis pour l'utilisation du Laplacien

- Les points contours correspondent aux passages par 0 de Δf .
- Problème : l'approximation de Δf est fortement bruitée (cf. diapos suivantes).
- On va donc :
 - détecter les points où Δf change de signe plutôt que ceux où $\Delta f(x,y)=0$;
 - imposer un seuil
 - et/ou pré-lisser l'image pour en réduire le bruit avant de calculer le Laplacien.

• Exemple d'algorithme

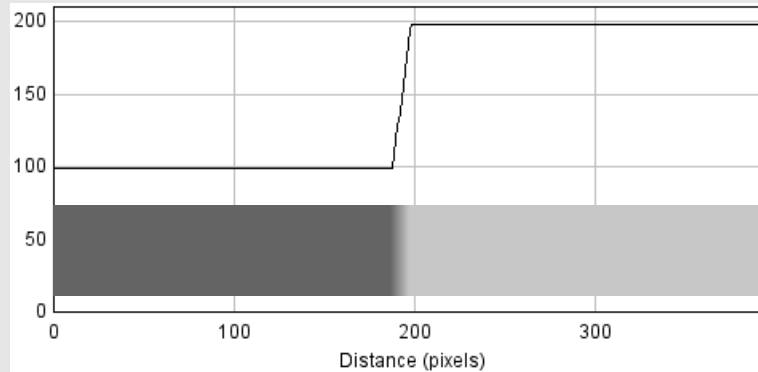
- Principe : seuil sur les valeurs locales minimum et maximum du Laplacien
- Paramètres d'entrée : Laplacien de l'image Δf , seuil S_Δ .
- En chaque pixel,
 - considérer un voisinage (ex. 3x3) centré ;
 - calculer $m := \min(\Delta f)$ et $M := \max(\Delta f)$ dans ce voisinage ;
 - le pixel est considéré comme point contour si $m < -S_\Delta$ et $M > S_\Delta$.

Autre critère possible : $m < 0$ et $M > 0$ et $M-m > S_\Delta$.

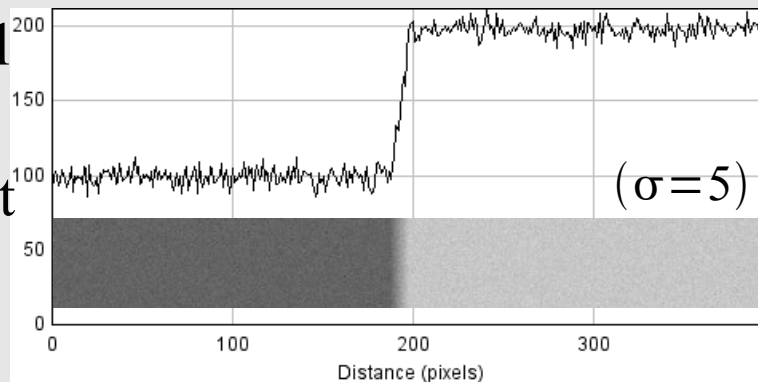
Réduction de la sensibilité au bruit (1/4)

Effet de la dérivation sur le bruit

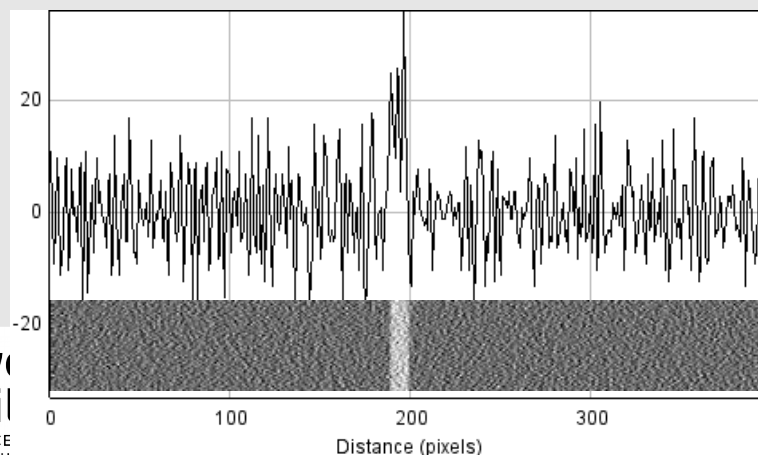
signal
idéal \tilde{f}



signal réel
 $f = \tilde{f} + b$
où b bruit
gaussien

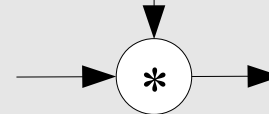
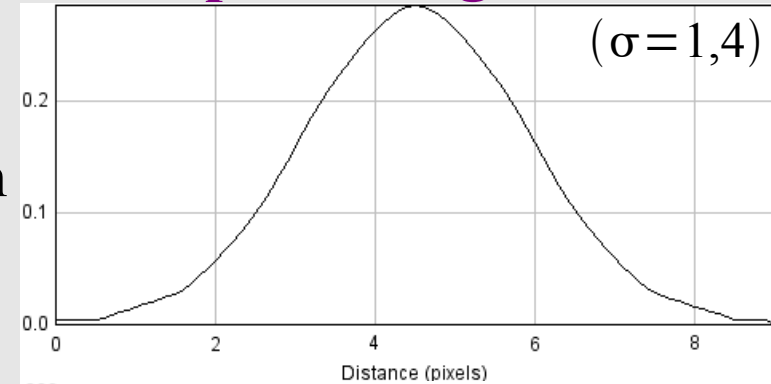


dérivée
 $\frac{df}{dx}$

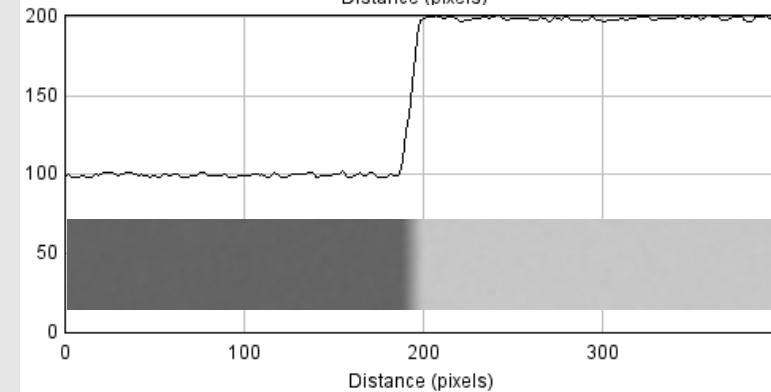


Solution : pré-filtrage

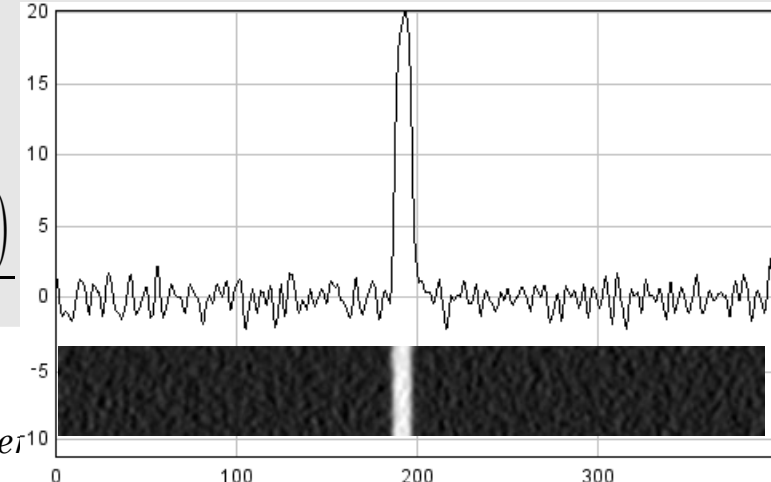
noyau
gaussien
 g



$f * g$



$\frac{d(f * g)}{dx}$



Réduction de la sensibilité au bruit (2/4)

• Opérateur LoG (*Laplacian of Gaussian*)

- Du fait de la double dérivation, le Laplacien est très sensible au bruit.
- Nécessité de lisser l'image en la pré-filtrant avec un noyau gaussien

$$g_{\sigma}(x, y) := \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

avant d'utiliser le Laplacien pour détecter les points contours.

- Possibilité de réaliser ces deux opérations en une seule :

$$\Delta\left[(g_{\sigma} * f)\right] = \underbrace{(\Delta g_{\sigma})}_{\text{LoG}_{\sigma}} * f$$

Justification :

$$\begin{aligned} \frac{\partial}{\partial x}(h * f)(x, y) &= \frac{\partial}{\partial x} \iint_{\mathcal{V}} h(x-u, y-v) f(u, v) du dv \\ &= \iint_{\mathcal{V}} \frac{\partial h}{\partial x}(x-u, y-v) f(u, v) du dv \\ &= \left(\frac{\partial h}{\partial x} * f\right)(x, y) \end{aligned}$$

Réduction de la sensibilité au bruit (3/4)

• Expression et masque de l'opérateur LoG

$$\frac{\partial g_{\sigma}}{\partial x}(x, y) = -\frac{x}{2\pi\sigma^4} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

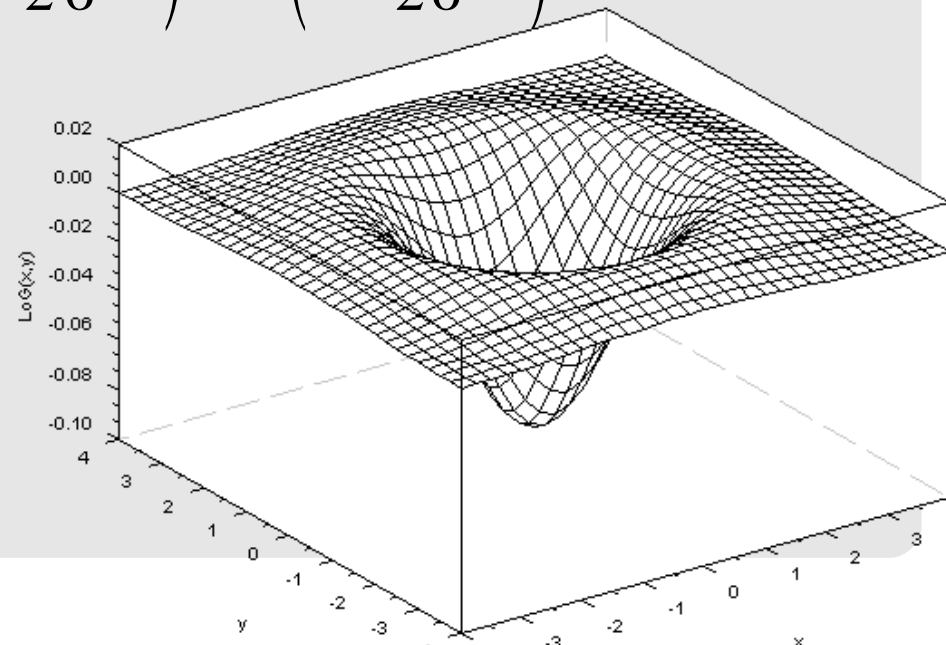
$$\frac{\partial^2 g_{\sigma}}{\partial x^2}(x, y) = -\frac{1}{2\pi\sigma^4} \left(1 - \frac{x^2}{\sigma^2}\right) \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

d'où

$$\text{LoG}_{\sigma}(x, y) := \Delta g_{\sigma}(x, y) = -\frac{1}{\pi\sigma^4} \left(1 - \frac{x^2+y^2}{2\sigma^2}\right) \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

→ **Approx.
discrète
de taille
9x9
pour
 $\sigma=1,4$**

0	+1	+1	+2	+2	+2	+1	+1	0
+1	+2	+4	+5	+5	+5	+4	+2	+1
+1	+4	+5	+3	0	+3	+5	+4	+1
+2	+5	+3	-12	-24	-12	+3	+5	+2
+2	+5	0	-24	-40	-24	0	+5	+2
+2	+5	+3	-12	-24	-12	+3	+5	+2
+1	+4	+5	+3	0	+3	+5	+4	+1
+1	+2	+4	+5	+5	+5	+4	+2	+1
0	+1	+1	+2	+2	+2	+1	+1	0



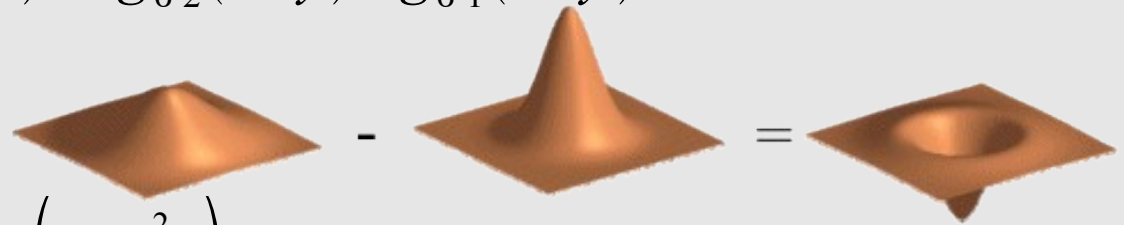
Réduction de la sensibilité au bruit (4/4)

• Approximation de LoG par différence de gaussiennes (DoG)

- Le filtre LoG peut être approché par la différence de deux gaussiennes d'écart-types proches :

$$\text{LoG}_{\sigma_1}(x, y) \approx \text{DoG}_{\sigma_1, \sigma_2}(x, y) := g_{\sigma_2}(x, y) - g_{\sigma_1}(x, y)$$

avec $\sigma_2 = \sigma_1 + \delta$, $\delta \ll \sigma_1$



- Démonstration en 1D :

$$\text{Considérons } f_x(\sigma) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

dont le développement de Taylor en σ_1 est $f_x(\sigma_1 + \delta) = f_x(\sigma_1) + \delta \frac{d f_x}{d \sigma}(\sigma_1) + O(\delta^2)$

$$\begin{aligned} \text{Alors } \text{DoG}_{\sigma_1, \sigma_2}(x) &= g_{\sigma_1 + \delta}(x) - g_{\sigma_1}(x) = f_x(\sigma_1 + \delta) - f_x(\sigma_1) \\ &\approx \delta \frac{d f_x}{d \sigma}(\sigma_1) = \frac{\delta}{\sqrt{2\pi}} \left(-\frac{1}{\sigma_1^2} - \frac{1}{\sigma_1} \cdot \left(-\frac{x^2}{\sigma_1^3} \right) \right) \exp\left(-\frac{x^2}{2\sigma_1^2}\right) \\ &\approx -\frac{\delta}{\sigma_1^2 \sqrt{2\pi}} \left(1 - \frac{x^2}{\sigma_1^2} \right) \exp\left(-\frac{x^2}{2\sigma_1^2}\right) = \delta \sigma_1 \frac{d^2 g_{\sigma_1}}{dx^2}(x) \end{aligned}$$

Utilisation de l'opérateur LoG

• Détecteur de contours de Marr-Hildreth

→ Principe : localiser les contours aux passages par 0 de $\text{LoG}_\sigma * f$.

→ Justifications

- Nécessité de lisser l'image avant d'appliquer le Laplacien.
- Le lissage gaussien réalise le meilleur compromis entre détection et localisation.

→ Algorithme

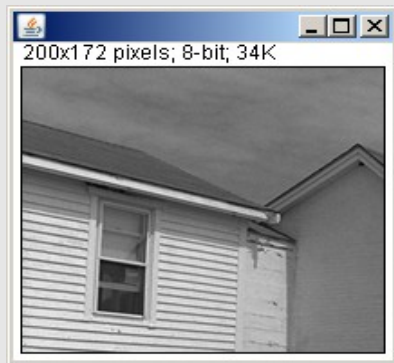
- Convolution de l'image avec LoG_σ (approché par un masque de taille $n \times n$).
Autre possibilité : utiliser $\text{DoG}_{\sigma_1, \sigma_2}$ (avantage : filtres gaussiens séparables).
- Détection des passages par 0 de l'image résultante.
- (Éventuellement :) Seuillage des passages par 0.
 - Avantage : prise en compte des seuls passages par 0 significatifs.
 - Inconvénient : on perd la propriété de fermeture présentée par les lignes de passages par 0 du Laplacien.

→ Limites :

- pour σ élevé, localisation médiocre et points d'intérêt saillants « perdus ».

Approche multi-échelles (1/2)

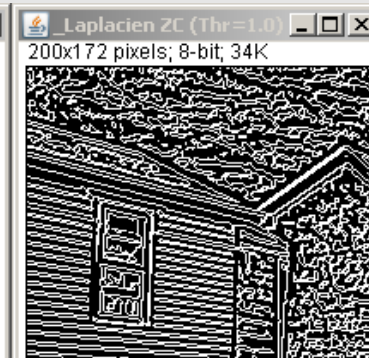
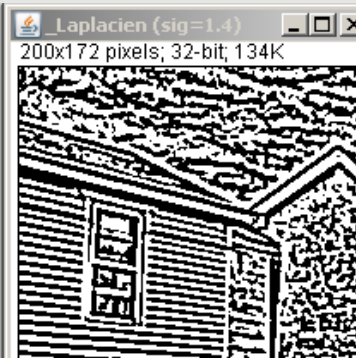
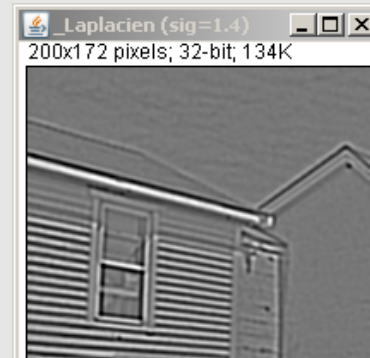
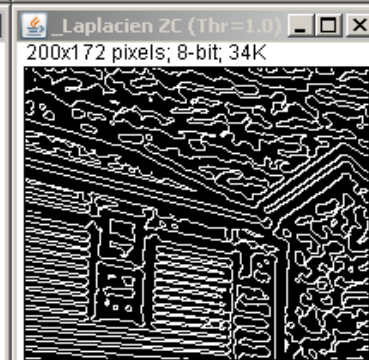
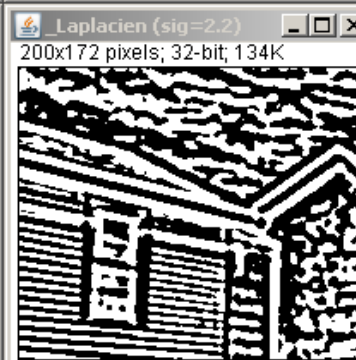
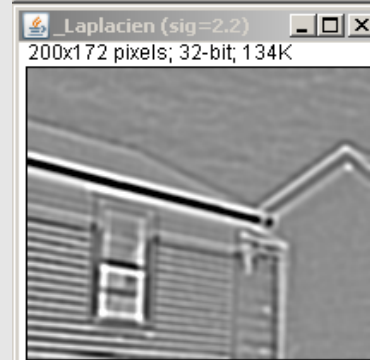
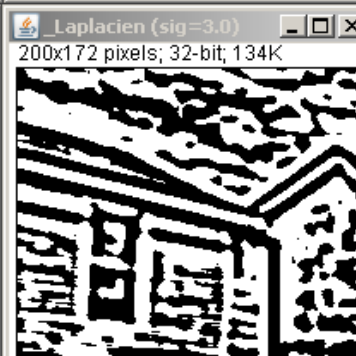
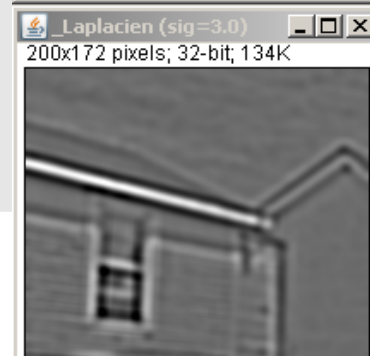
- Principe** : Détecter des contours plus ou moins significatifs, en pré-lissant plus ou moins fortement l'image grâce au filtre gaussien (σ est l'« échelle »).



Laplacien (32 bits)

Polarité

Passages par 0

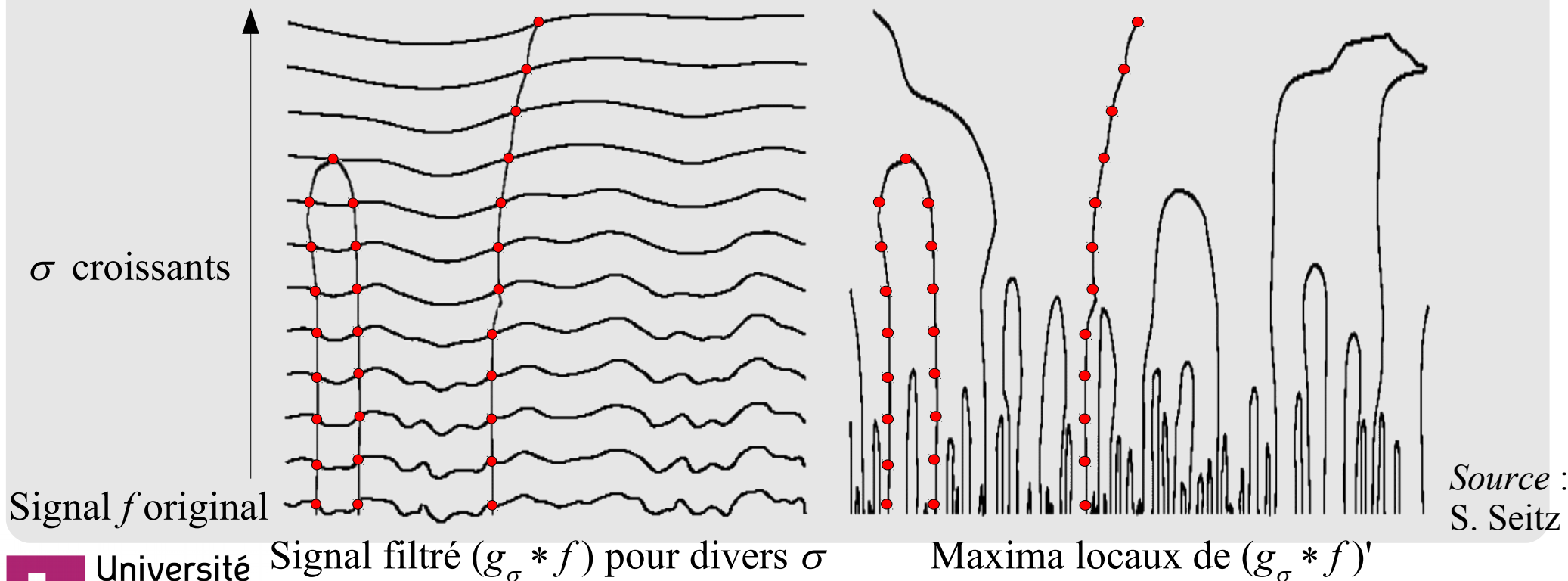
 $\sigma = 1,4$  $\sigma = 2,2$  $\sigma = 3,0$

On obtient différents
niveaux de détails
pour les objets

Approche multi-échelles (2/2)

Avantages et limites de l'approche

- 👍 deux contours peuvent « fusionner » lorsque σ augmente
- 👍 un contour ne peut *pas* être divisé lorsque σ augmente
- 👍 s'applique aussi avec le gradient (\Rightarrow détecteur de Canny)
- 🙅 la position du contour peut varier lorsque σ augmente (médiocre localisation)



Comparaison Gradient / Laplacien (1/3)

• Résumé des approches vues pour la détection des points contours

➔ Gradient (*ex. Canny*)

- Calcul des dérivées partielles (lissées)
- Calcul de la norme et de la direction du gradient
- Suppression des non-maxima locaux de la norme du gradient dans sa direction
- Seuillage par hystérésis des maxima locaux de la norme du gradient

➔ Laplacien

- (lissage de l'image)
 - Calcul du Laplacien
 - Détection des passages par 0 du Laplacien
 - Seuillage des passages par 0 du Laplacien
- } LoG, DoG

• Approches combinées Laplacien-gradient

➔ Principe : utilisation de $\|\nabla G\|$ pour seuiller les passages par 0 de Δf .

- Calcul du Laplacien (après lissage)
- Calcul de la norme du gradient
- Détection des passages par 0 de Δ (\Rightarrow masque I_z) à partir de l'image de polarité I_p
- Application du masque binaire I_z à l'image de la norme du gradient
- Seuillage de l'image résultante (simple, par hystérésis, ...)

Comparaison Gradient / Laplacien (2/3)

• Gradient

→ Avantages

- Fournit l'orientation du contour
- Bonne localisation malgré le lissage
- La suppression des non-maxima locaux fournit des contours fins

→ Inconvénients

- Assez sensible au bruit
⇒ nécessite un lissage
- Le seuillage fournit des contours non fermés.

• Les deux approches

→  nécessitent l'ajustement de paramètres (σ et S_Δ , S_b et S_h)

→  permettent une interpolation subpixelique

• Laplacien

→ Avantages

- Proche du système visuel humain
- La détection des passages par 0 fournit des réseaux de lignes **fermées**

→ Inconvénients

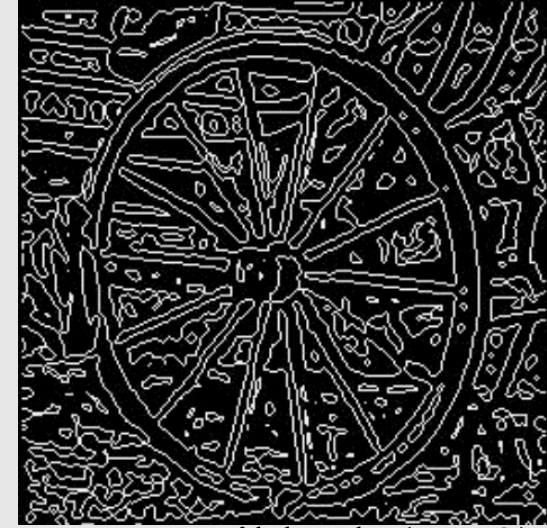
- Grande sensibilité au bruit
⇒ nécessite un lissage fort
⇒ affecte la localisation
- Pas d'info sur l'orientation du contour
- Le seuillage des passages par 0 crée des lacunes (« ouvertures ») dans les contours

Comparaison Gradient / Laplacien (3/3)

- Exemples de détection des points contours



Canny



Marr-Hildreth ($\sigma=2$)

- Conclusion

- Pas d'opérateur parfait pour détecter les contours
- On obtient en pratique des contours incomplets (ouverts)
 - **détection** incorrecte : pixels superflus, pixels manquants
 - **localisation** incorrecte : erreurs dans la position des points contours, l'orientation
- La détection des points contours n'est que la première étape dans la chaîne de segmentation

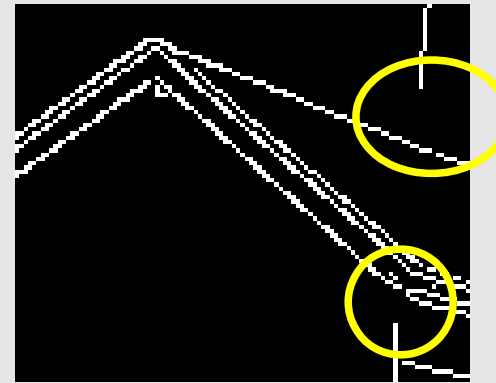
Généralités sur les post-traitements (1/2)

- **La détection des points contours fournit**

- une carte binaire des *points* contours ;
- souvent, des contours ouverts (composantes de l'image non séparées en objets topologiquement distincts).

Causes :

- bruit
- faible contraste
- occultations



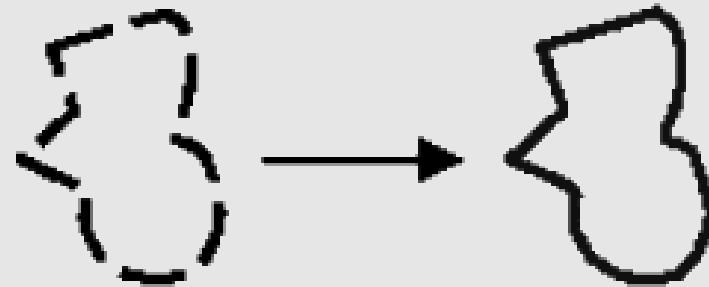
- **Nécessité de fermer les contours**

- **Pour obtenir des régions fermées**
 - définies comme des composantes connexes maximales n'incluant pas de points contours ;
 - interprétables comme projections des objets de la scène.
- **Très important pour l'exploitation en segmentation d'image.**

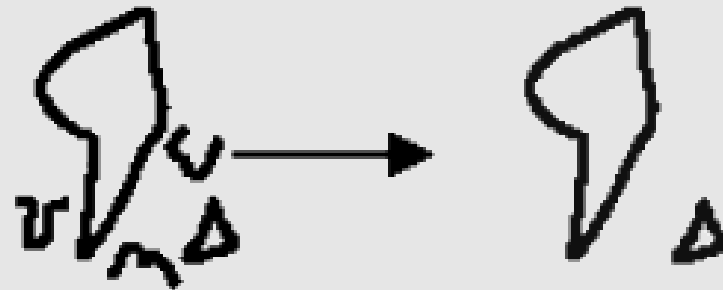
Généralités sur les post-traitements (2/2)

- **Fermeture des contours par extrapolation**

Ajout de points non détectés

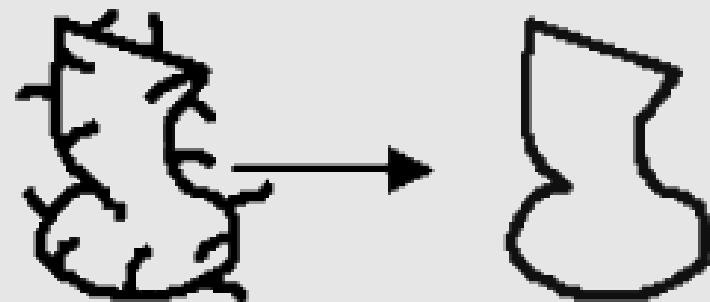


- **Suppression des contours non fermés / non significatifs**



- **Suppression des « branches pendantes » des contours fermés (ébarbulation)**

Rejet de points détectés par erreur



- **Codage ou chaînage des contours**

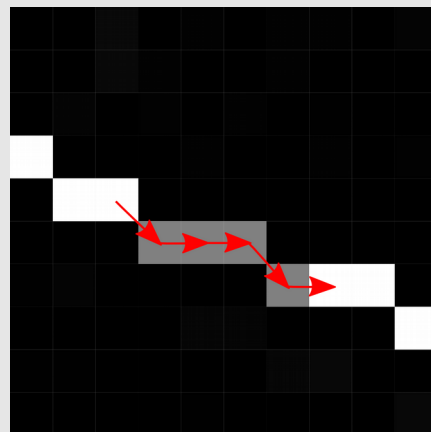
Fermeture des contours (1/4)

• Principe

- ➔ Prolonger les extrémités des chaînes de points contours vers d'autres chaînes de points contours.

• Techniques d'exploration de graphes

- ➔ Principe : exploration de l'espace des chemins solutions,
 - chaque pixel étant un nœud du graphe,
 - chaque connexion d'un pixel à l'un de ses voisins étant un arc du graphe.



- Point contour
- ➔ Chemin solution pour la fermeture
- Point contour de fermeture

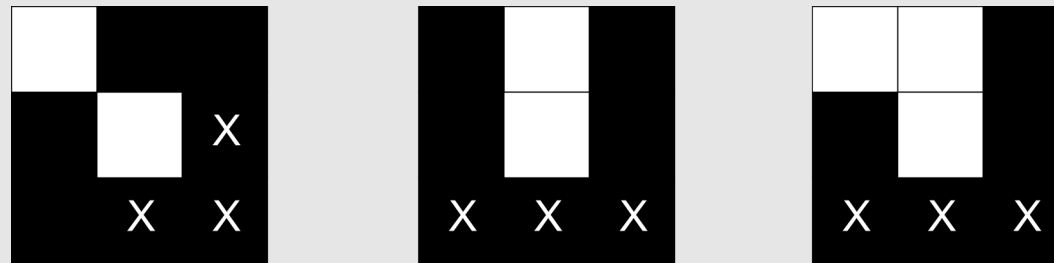
- ➔ **Avantage : méthodes aux bases solides, fournissant les meilleurs résultats.**
- ➔ **Inconvénient : coût élevé (explosion combinatoire)**

Fermeture des contours (2/4)

• Exploration de graphes : détails (1/2)

→ Approche générale

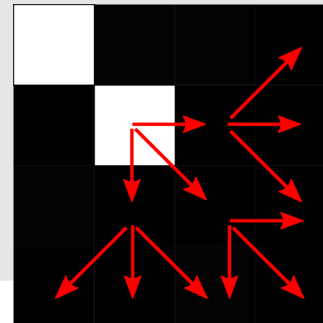
- recherche des extrémités des composantes connexes de points contours
- sélection de points candidats à la prolongation du contour (**X**)



- prolongation suivant critère (minimum d'une fonction de coût)

→ Implémentation simpliste

- fonction de coût : inverse de la norme du gradient
- mais la sélection sur un niveau atteint rapidement ses limites :
⇒ utiliser plusieurs niveaux, ex. 2 :
- le coût sur n niveaux est la somme du coût en chaque pixel candidat.



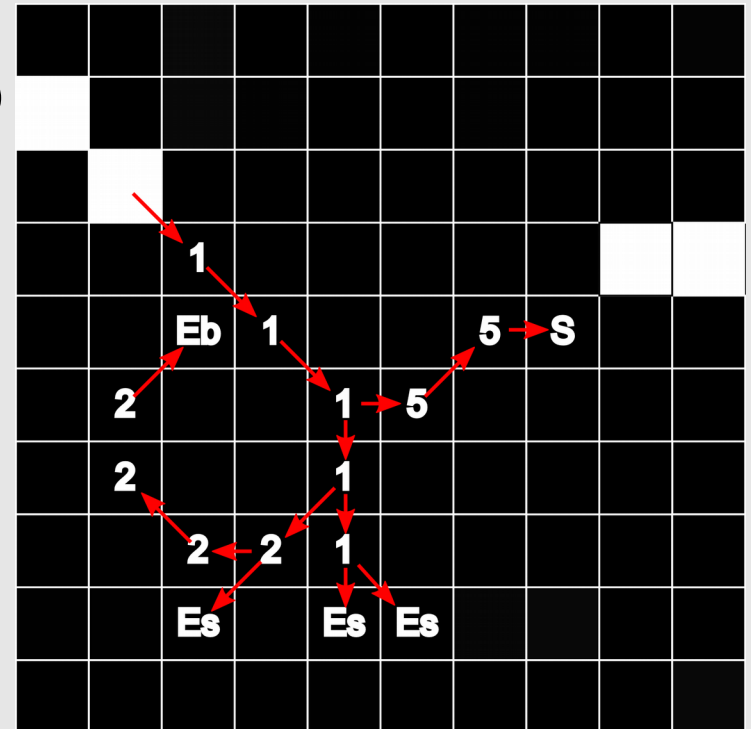
			4
		11	8
	20	19	12
10	8	9	25

Fermeture des contours (3/4)

• Exploration de graphes : détails (2/2)

➔ Implémentation avec retour sur trace (*ang.* « backtracking »)

- Utilise un arbre de recherche dont les nœuds sont les bifurcations possibles dans la chaîne de fermeture, et dont les branches sont les points contours candidats.
- À chaque itération (niveau), on garde tous les candidats auxquels la norme du gradient est supérieure à un seuil, et on les classe par coût croissant.
- Échec d'un chemin si
 - aucun candidat de norme $>$ au seuil (**Es**)
 - création d'une chaîne bouclée (**Eb**)
 ⇒ remontée au nœud précédent
- Fin de recherche si
 - rencontre d'un point contour (succès **S**)
 - tous les chemins mènent à un échec (fermeture impossible)
 - (nombre max. d'itérations atteint)



Source : J. Desachy

Fermeture des contours (4/4)

- **Autres méthodes**

- **Dilatation \mathcal{D} / Réduction topologique \mathcal{R}**



- Contraintes de \mathcal{R} : préserve la topologie et les points contour initiaux
 - cf. détails sur <http://dpt-info.u-strasbg.fr/~cronse/TIDOC/ATG/fermcont.html>

- **Méthodes neuromimétiques (réseaux de neurones)**

- **Les contours actifs (*ang.* « snakes ») évitent l'étape de fermeture**

- **Courbe 2D déformable qui épouse progressivement les contours des objets**
 - cf. http://imagejdocu.tudor.lu/doku.php?id=plugin:segmentation:active_contour:start

Codage/chaînage des contours

- **Principe**

- Décrire un contour au moyen d'une structure de donnée.

- **Exemple simple : codage de Freeman**

- On code les changements de direction d'un pixel de contour à son voisin.

- La séquence des codes locaux constitue le codage du contour. *Ex. à partir de P :*
 $\{ 6, 0, 7, 0, 2, 0, 1, 1, 2, 2, 4, 2, 4, 5, 6, 4, 6, 4, 5 \}$

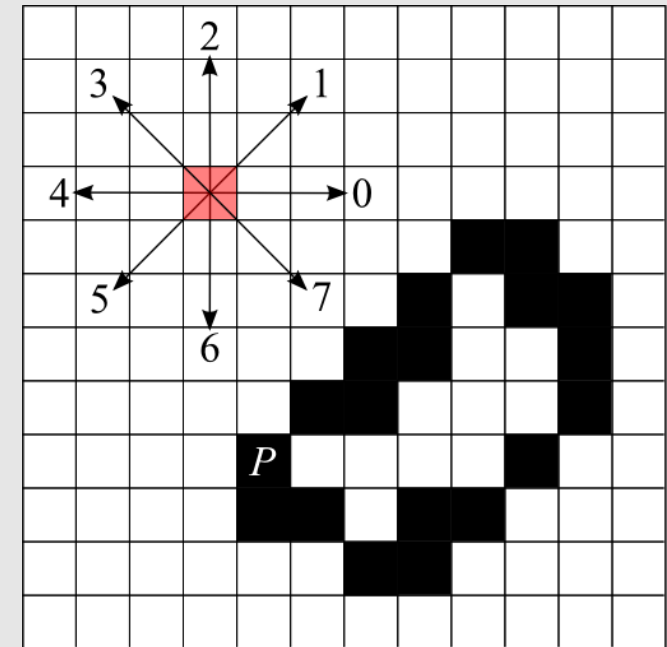
- Réduction de la chaîne de contour

- **Autres approches possibles**

- Approches *globales* : recherche des contours complets \neq seulement points.

- Exemple : transformée de **Hough** (détection de contours continus en classes de formes : lignes, mais aussi cercles et ellipses).

⇒ **cf. cours de Reconnaissance de Formes !**



Sélection de références

• Livre

- W. Burger, M.J. Burge, *Digital Image Processing – An Algorithmic Introduction Using Java*, Springer 2008.

• Sites web

- Cours de J.-H. Thomas (Université du Maine)

<http://www.optique-ingenieur.org/fr/cours/>

[OPI_fr_M04_C05/co/Grain_OPI_fr_M04_C05.html](http://www.optique-ingenieur.org/fr/cours/OPI_fr_M04_C05/co/Grain_OPI_fr_M04_C05.html)

- Image Processing Learning ressources – Explore with Java

Ex. filtre LoG : <http://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm>

- Using Laplacian for Edge Detection (R. Wang)

<http://fourier.eng.hmc.edu/e161/lectures/gradient/node9.html>

- Image Filtering & Edge Detection (N. Vasconcelos)

<http://www.svcl.ucsd.edu/courses/ece161c/handouts/EdgesAndInterpolation.pdf>