

TRAITEMENT D'IMAGE

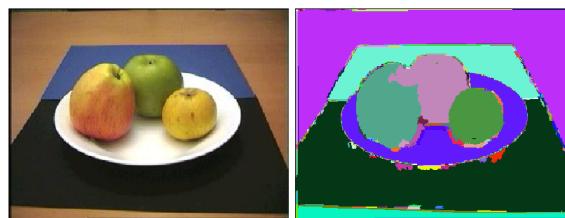
Segmentation d'image

Traitement d'image

- Segmentation :

Qu'est ce que la segmentation ? Et pourquoi segmenter ?

- La segmentation sert à partager l'image en plusieurs morceaux
 - chaque morceau correspond à un objet
- Pour reconnaître des objets dans une image, on a besoin de la segmenter afin de détecter ces objets => **reconnaissance d'objets.**



Traitement d'image

- Segmentation :

Principe de la segmentation

La segmentation peut se faire soit en détectant :

- les discontinuités : **contours**
 - les changements abruptes, frontières entre régions ...
- les zones homogènes : **régions**
 - mêmes couleurs, textures, intensités ...

La segmentation est le découpage d'une image en différentes **régions** et/ou **contours**

Un contours fermé est équivalent à une région.

Traitement d'image

- Segmentation :

Approches

- **Approches pixels** : seuillage

- **Approches régions** :

- rechercher les zones homogènes dans l'image

- **Approches contours** :

- rechercher les discontinuités entre les régions dans l'image

- **Approches duales ou hybrides** (contours et régions)

SEGMENTATION D'IMAGE

Approche pixels: seuillage, k-moyennes

Traitement d'image

- Segmentation : Principe de base
seuillage

Le seuillage peut être :

- **global** : un seuil pour toute l'image
- **local** : un seuil pour chaque portion de l'image
- **adaptatif** : un seuil s'ajustant selon les parties de l'image

Seuillage de base (2 classes) :

$$\begin{array}{ll} \text{si } I(\text{pixel}) > \text{seuil} & \text{alors } I(\text{pixel}) = 1 \\ \text{si } I(\text{pixel}) \leq \text{seuil} & \text{alors } I(\text{pixel}) = 0 \end{array}$$

L'image résultat du seuillage est une image binaire

Problème : choix de la valeur du seuil

Traitements de base

- Seuillage:

Exemple image « saturn.tif » :



Image seuillée ou binarisée

Image d'origine

Traitement d'image

- Segmentation : **principe de base**

Seuillage

Multi-seuils

- n seuils pour séparer l'image en $n+1$ classes :
 - Si $valeur(pixel) < seuil_1$
 - alors $valeur(pixel) = 0$
 - Si $valeur(pixel) \geq seuil_1 \text{ } \&\& \text{ } valeur(pixel) < seuil_2$
 - alors $valeur(pixel) = 1$
 - ...
 - Si $valeur(pixel) \geq seuil_n$
 - alors $valeur(pixel) = n$

Problèmes : le nombre et les valeurs des seuils !

Traitement d'image

- Segmentation : **principe de base**

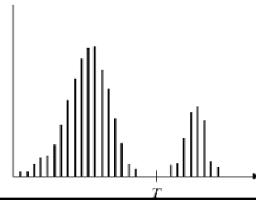
Seuillage

Comment trouver le bon seuil ?

Le seuil est choisi soit manuellement ou calculé automatiquement.

Sa valeur peut être obtenue :

- Par tests
- En calculant la moyenne des intensités
- En calculant la médiane des intensités
- Comme la valeur qui balance les deux sections de l'histogramme
- Par des algorithmes automatiques tels que:
 - Otsu
 - Kittler
 - ...



Traitement d'image

- Segmentation : **principe de base**

Seuillage

Choix du seuil optimal

- Pour trouver un seuil, il existe plusieurs méthodes globales automatiques.
- Chaque méthode donne un seuil différent → quel est le bon seuil ?
- Il faut tester pour chaque nouvelle application

Traitement d'image

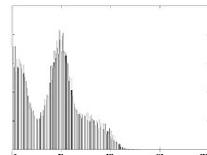
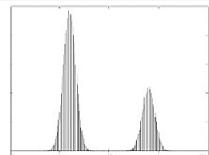
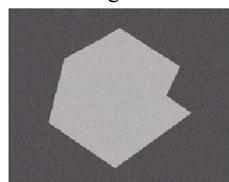
- Segmentation : **principe de base**

Seuillage

Seuillage global

Problème d'éclairage

Seuillage
global facile



Seuillage
global pas
évident



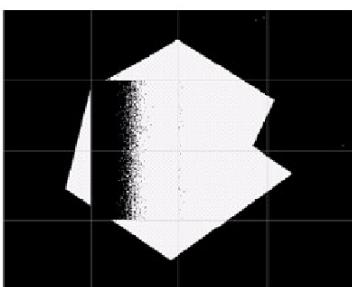
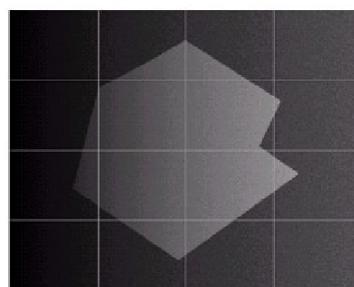
Traitement d'image

- Segmentation : **principe de base**

Seuillage

Seuillage local adaptatif

- On divise l'image en sous-images
- On seuille chaque sous-image indépendamment
- Les 4 sous images de coins ne sont pas traitées car $variance < 100$



Traitement d'image

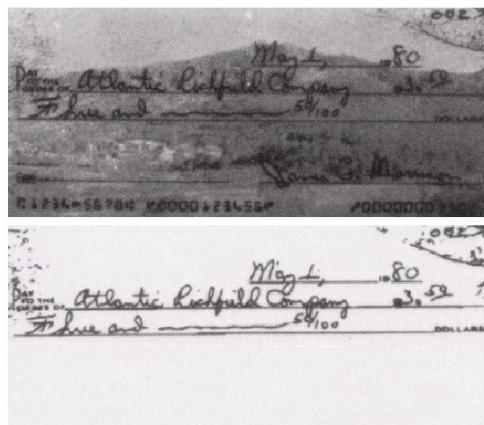
- Segmentation : **principe de base**

Seuillage

Seuillage local adaptatif

a
b

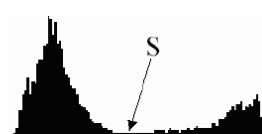
FIGURE 10.37
(a) Original
image. (b) Image
segmented by
local thresholding.
(Courtesy of IBM
Corporation.)



Traitement d'image

- Segmentation : **Principe de base**
seuillage

Avantages	Inconvénients
Universel Temps réel Simple	<ul style="list-style-type: none">Connaître le nombre de classes (et donc les seuils)Apparition de faux éléments (la composante spatiale n'est pas prise en compte)



2 seuils pour 3 classes

Traitement d'image

- Segmentation : principe de base
k-moyennes (k-means)

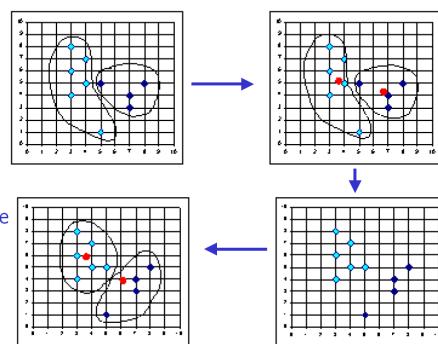
On veut diviser les points en k groupes (*clusters*)

- k est donné à l'avance (un paramètre de l'algorithme)
- On définit le centre d'un groupe comme la moyenne des éléments (pixels) du groupe

L'algorithme est composé de 4 étapes :

1. Partitionnement des données en k sous-ensembles (non vides)
2. Calcul des centres des groupes de la partition courante
3. Les données sont affectées au groupe dont le centre leur est le plus proche
4. Retour à l'étape 2

Arrêt lorsque les groupes sont \sim constants



Traitement d'image

- Segmentation : principe de base
k-moyennes (k-means)



Image source



K-moyennes sur l'intensité



K-moyennes sur la couleur

K-moyennes basé sur l'intensité (gris) ou sur la couleur

Traitement d'image

- Segmentation : principe de base

Approche pixels

- L'approche pixels effectue des opérations sur les pixels pour segmenter les régions.
- La méthode utilisant le seuillage est une méthode simple mais qui nécessite un nettoyage des résultats obtenus :
 - élimination des pixels isolés pour conserver que les blocs de pixels.
- Il existe d'autres méthodes de segmentation des régions conservant la connexité des régions.

SEGMENTATION D'IMAGE

Approche régions: Division-Fusion, croissance de régions

Traitement d'image

- Segmentation : principe de base

Division-Fusion

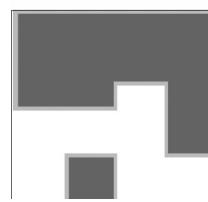
Etape de division (split)

- Diviser récursivement tout bloc non-homogène selon un prédictat défini
 - variance, max-min, ...
- La division d'un bloc donne 4 sous-blocs
- Les attributs de chaque sous-bloc sont recalculés



Etape de fusion (merge)

- Regrouper les blocs adjacents représentant des régions homogènes selon un prédictat défini



Traitement d'image

- Segmentation : principe de base

Division-Fusion (étape de division)

0 1 0 0 7 7 7 7
1 0 2 2 7 7 7 7
0 2 2 2 7 7 7 7
4 4 2 2 7 7 7 7
0 0 1 1 3 3 7 7
1 1 2 2 3 7 7 7
2 4 3 0 5 7 7 7
2 3 3 5 5 0 7 7

Image initiale

0 1 0 0 | 7 7 7 7
1 0 2 2 | 7 7 7 7
0 2 2 2 | 7 7 7 7
4 4 2 2 | 7 7 7 7
0 0 1 1 | 3 3 7 7
1 1 2 2 | 3 7 7 7
2 4 3 0 | 5 7 7 7
2 3 3 5 | 5 0 7 7

Division 1

0 1 0 0 | 7 7 7 7
1 0 2 2 | 7 7 7 7
0 2 2 2 | 7 7 7 7
4 4 2 2 | 7 7 7 7
0 0 1 1 | 3 3 7 7
1 1 2 2 | 3 7 7 7
2 4 3 0 | 5 7 7 7
2 3 3 5 | 5 0 7 7

Division 2

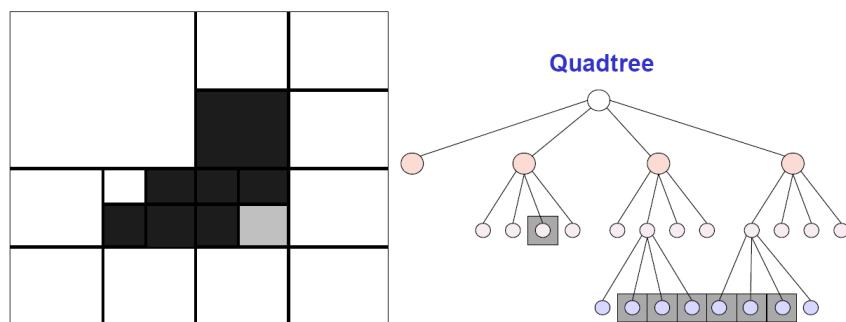
0 1 0 0 | 7 7 7 7
1 0 2 2 | 7 7 7 7
0 2 2 2 | 7 7 7 7
4 4 2 2 | 7 7 7 7
0 0 1 1 | 3 3 7 7
1 1 2 2 | 3 7 7 7
2 4 3 0 | 5 7 7 7
2 3 3 5 | 5 0 7 7

Division 3

*Homogénéité = critère sur la variance
(ou max-min ≤ 1)*

Traitement d'image

- Segmentation : principe de base
- Division-Fusion (étape de fusion)**



Traitement d'image

- Segmentation : principe de base
- Division-Fusion (exemple)**



Traitement d'image

- Segmentation : principe de base

Croissance de régions

- On débute avec un pixel, et on « ajouter » les pixels voisins qui répondent à un critère d'appartenance :
 - *Variance faible*
 - *Niveau de gris répondant un seuil*
 - ...
- Les pixels initiaux sont appelés « germes », « amorces » ou semences«
 - *Choix des pixels initiaux automatiques ou semi-automatiques*
- La région « grandit » à partir de son germe
 - *Besoin d'une critère (ou prédict) pour choisir les pixels à ajouter*

Traitement d'image

- Segmentation : principe de base

Croissance de régions

- On part d'un germe (*seed*) et on l'étend en ajoutant les pixels voisins qui satisfont le critère d'homogénéité
- Le germe peut être choisi soit par un humain, soit de manière automatique en évitant les zones de fort contraste (*gradient important*)



SEGMENTATION D'IMAGE

Approche contours: filtre de roberts, Sobel, Prewitt, Gradient, Laplacien

Traitement d'image

• Contour:

➤ Définition :

Un contour est une variation brusque d'intensité.



Par définition, un contour est la frontière qui sépare deux objets dans une image.

- ✓ Une discontinuité de l'image

Dans notre cas, nous détecterons toutes les lignes marquant des changements d'intensité :

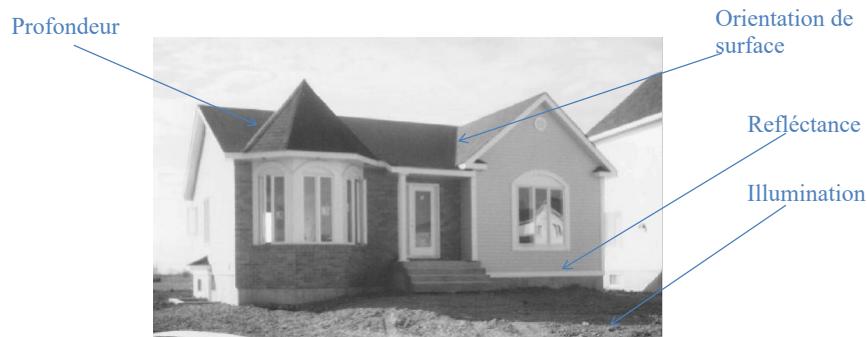
- ✓ Pas seulement les contours !
- ✓ Abus de langage sur la notion de contours !

Traitement d'image

- **Contour:**

➤ Définition :

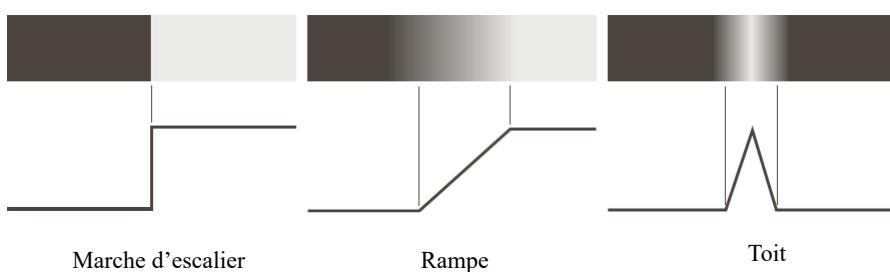
Exemples de détection des discontinuités :



Traitement d'image

- **Contour:**

➤ Type de contours :

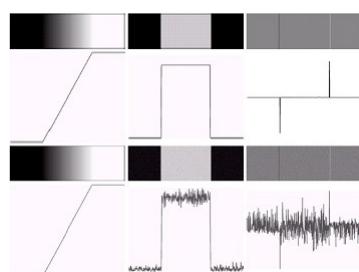


Traitement d'image

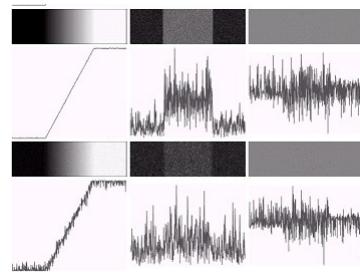
- **Contour:**

- **Type de contours :**

Et si le contour est bruité ...



Un peu ..



Beaucoup ..

Traitement d'image

- **Détection de contours:**

- **Définition :**

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

Si les structures extraites sont simples à manipuler (courbes fines, régulières, stables...), elles peuvent être utiles pour la mise en correspondance d'images (robotique, indexation,...).



Image en niveau de gris

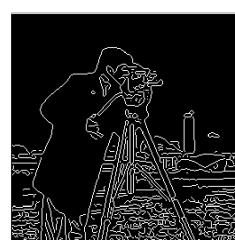


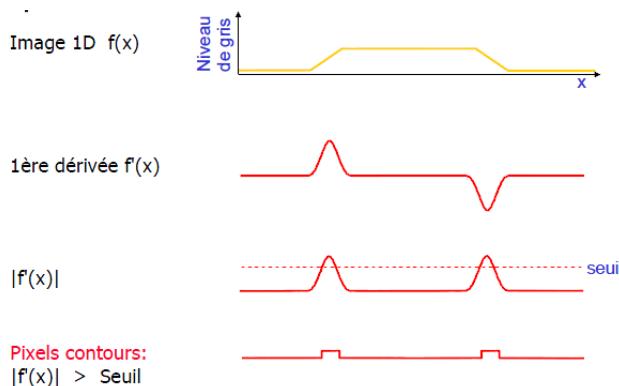
Image contour

Traitement d'image

- **Détection de contours:**

- **Méthode de détection :**

Pour détecter les contours dans une image, la première dérivée de l'image est l'opérateur de base.



Traitement d'image

- **Détection de contours:**

- **Méthode de détection :**

La première dérivée de l'image s'exprime analytiquement par :

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

On fait en général une approximation simple de la dérivée discrète :

$$\begin{array}{|c|c|} \hline -1 & 1 \\ \hline \end{array} \quad \begin{array}{|c|} \hline -1 \\ \hline 1 \\ \hline \end{array} \quad \text{ou} \quad \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline \end{array} \quad \begin{array}{|c|c|c|} \hline -1 \\ \hline 0 \\ \hline 1 \\ \hline \end{array}$$

Traitement d'image

• Détection de contours:

➤ Méthode de détection :

La première approximation de la dérivée d'une image discrète est celle faite par Roberts (1965) → **filtre de Roberts**

Le filtre de Roberts est réalisé en convoluant l'image par deux masques de convolution. Chaque masque correspond à une direction de la dérivée :

$$\begin{array}{c} \begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix} \quad \begin{matrix} 1 & 0 \\ 0 & -1 \end{matrix} \\ \searrow \quad \swarrow \\ G_x \quad G_y \end{array}$$

Les masques de Roberts sont les suivants :

Roberts: $\begin{matrix} 1 & 0 \\ 0 & -1 \end{matrix}$ $\begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix}$

Traitement d'image

• Détection de contours:

➤ Méthode de détection :

Il existe d'autres filtres pour détecter les contours dans une image. Dans ces filtres on effectue d'autres opérations additionnelles à la dérivée.

Exemples :

Roberts: $\begin{matrix} 1 & 0 \\ 0 & -1 \end{matrix}$ $\begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix}$

Avec les filtres de Prewitt et de Sobel on fait **lissage de l'image + dérivée de l'image**

Prewitt: $\begin{matrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{matrix}$ $\begin{matrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{matrix}$

Sobel: $\begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix}$ $\begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$

Traitement d'image

- Détection de contours:

➤ Méthode de détection : Lissage + Dérivée

Le lissage permet une détection de contours moins sensible au bruit.

Filtre de Prewitt : Moyenneur + Dérivée

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

Filtre de Sobel : Gaussienne + Dérivée

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

Traitement d'image

- Détection de contours:

➤ Méthode de détection : Gradient d'une image

Il est à noter que l'image est une fonction 2D. On parle alors de Gradient d'image → **dérivée selon x + dérivée selon y**

On obtient alors un vecteur avec une norme et une direction.

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

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

Direction : Direction du gradient le plus fort en chaque pixel

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

Traitement d'image

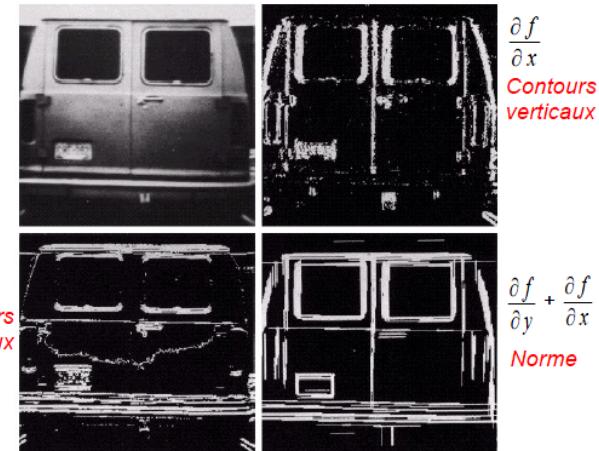
- Détection de contours:

➤ Méthode de détection :

a
b
c
d

FIGURE 10.16

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



Source : Gonzalez and Woods. Digital Image Processing. Prentice-Hall, 2002.

17

Traitement d'image

- Détection de contours:

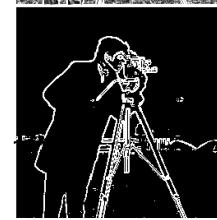
➤ Méthode de détection : Seuillage du Gradient

Sobel avec seuillage
seuil=25



Sobel sans
seuillage

Sobel avec seuillage
seuil=60



Traitement d'image

- **Détection de contours:**

➤ **Méthode de détection :**

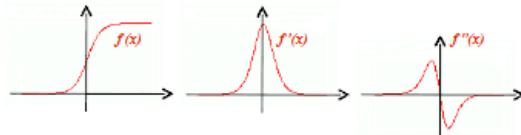
La première dérivée n'est pas la seule méthode pour détecter les contours.

La dérivée seconde peut aussi les détecter. Pour une image (2D) on parle alors de Laplacien.

L'opérateur Laplacien est défini de la façon suivante :

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

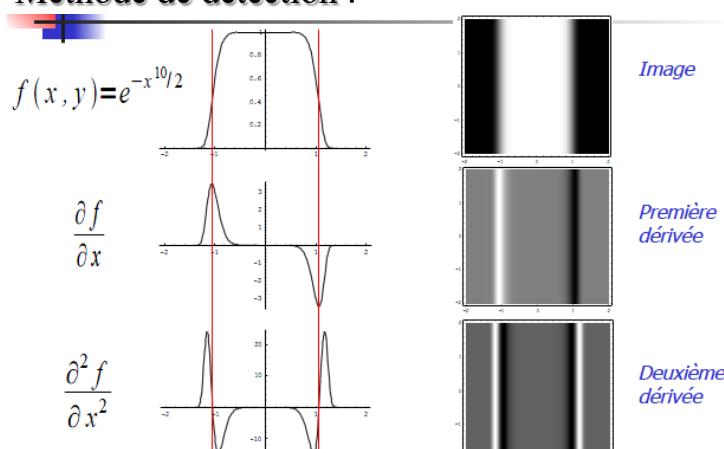
Le contour correspond à un maximum de la dérivée première et un zéro pour la dérivée seconde :



Traitement d'image

- **Détection de contours:**

➤ **Méthode de détection :**



Source : Caroline Rougier. Traitement d'images (IFT2730). Univ. de Montréal.

Traitement d'image

- **Détection de contours:**

➤ **Méthode de détection :**

Pour appliquer le Laplacien sur une image plusieurs approximation discrètes existent sous forme de masque de type :

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

A noter que le Laplacien est approximé par une seule matrice de convolution symétrique.

Traitement d'image

- **Détection de contours:**

➤ **Synthèse:**

Plusieurs filtres existent pour détecter les contours. Mais lequel choisir ?

- ✓ Il faut savoir qu'aucun opérateur n'est parfait.
- ✓ En réalité, le résultat obtenu est incomplet :
 - ❖ Il résulte des pixels superflus : qui n'appartiennent pas à un contour mais sont détectés
 - ❖ Il y'a aussi des pixels appartenant à un contour non détectés → manque
 - ❖ Il y a des erreurs de position et d'orientation des pixels de contours
- ✓ Le choix donc dépend de l'utilisateur
- ✓ Enfin la détection des contours est un traitement de segmentation de bas niveau dans une chaîne de traitement d'image.
- ✓ La détection des contours par gradient ou laplacien est une méthode dite locale (les filtres sont appliqués dans un voisinage de chaque pixel).
- ✓ Il existe d'autres méthodes de type globale: [Transformée de Hough](#) par exemple.

Traitement d'image

- Segmentation : principe de base

Conseils

- La segmentation d'une image cause encore aujourd'hui beaucoup de problèmes
 - *Aucune méthode ne fonctionne pour toutes les images*
 - *Pas de garantie, pas de recette miracle !*
- Le **pré-traitement** des images, la **sélection de capteurs et sources d'énergie** appropriées, et la **prise contrôlée des images** rendent cette étape plus facile et plus efficace
- La segmentation aide beaucoup pour la reconnaissance, mais elle n'est pas obligatoire dans tous les cas