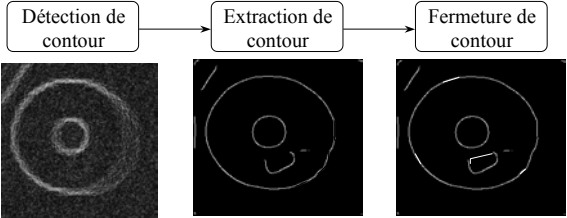
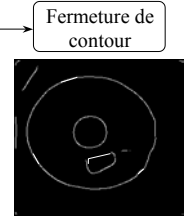
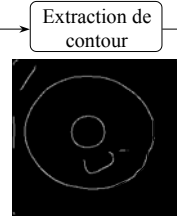
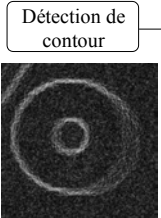
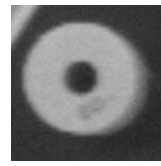


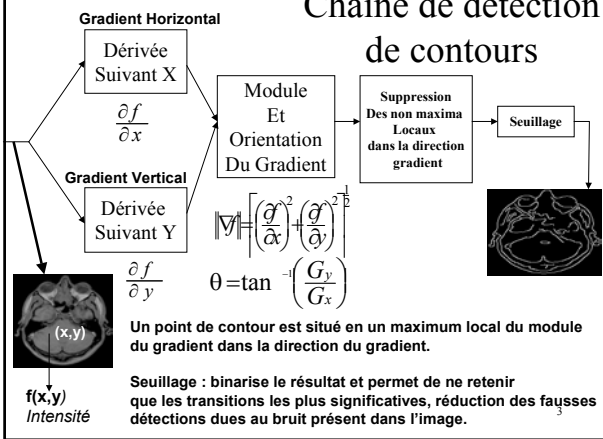
TD Implantation du filtre de Deriche sur DLX

Thierry Grandpierre

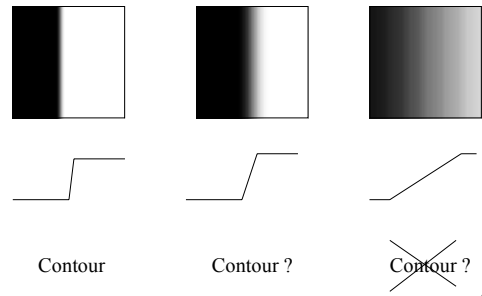
Détection des Contours : méthodes dérivatives



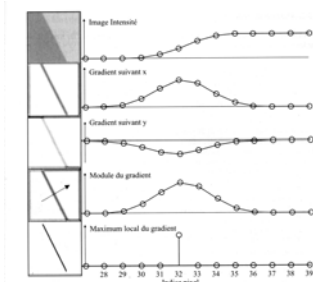
Chaîne de détection de contours



Problèmes



Gradient



Maximum local

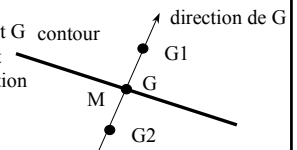
Un contour = ligne de crête dans l'image de la norme du gradient (I_G):

- des niveaux de gris toujours élevés
- de faibles dénivelés le long de ces lignes
- de forts dénivelés dans les autres directions

==> les points de **contour** = **maxima locaux** de I_G

Le principe est de comparer le gradient G contour en un point M avec les gradients $G1$ et $G2$ des deux voisins pris dans la direction du gradient

si $G > G1$ et $G > G2$, alors M est un maximum local



Différentes méthodes

- Dérivation au premier ordre
Prewitt, Sobel, Roberts, Kirsh
- Dérivation au second ordre
Laplacien, Marr et Hildreth
- Filtrage optimal
Canny-Deriche, Shen

Caractéristiques:

Complexité, précision de localisation, sensibilité au bruit, création de faux contours

7

Exemples de filtres à réponse impulsionnelle finie

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

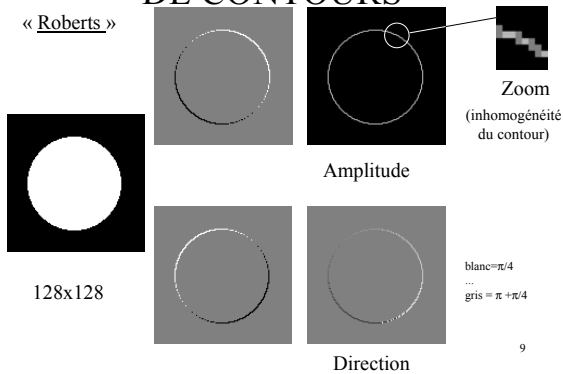
$$\text{Prewitt} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\text{Sobel} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

8

EXEMPLE DE DETECTION DE CONTOURS

« Roberts »



9

Masques Sobel

- Image d'intensité $N \times N$
- $P(i,j)$ = pixel i,j avec $i,j \in \{0, N-1\}$ (origine en haut à gauche)
- Image monodim: $p(k)$ avec $k \in \{0, N^2-1\}$
- Transformée en $z \rightarrow P(z)$

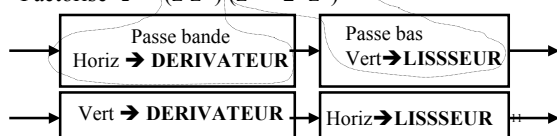
10

Sobel

- 2 Masques :

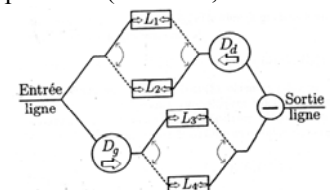
$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

- Horizontal : $gh(k) = p(k-N+1) - p(k-N-1) + 2p(k) - 2p(k-1) + p(k+N+1) - p(k+N-1)$
- $SH(z) = Gh(z)P(z^{-N+1} - z^{-N-1} + 2z - 2z^{-1} + z^{N+1} - z^{N-1})$
Factorisé $\Rightarrow (z-z^{-1})(z^{-N+2} + z^N)$



Cany Deriche

- Sobel : faible rapport signal/bruit
 - Transformée en z de Deriche :
- $$D(z) = k_D \left(\frac{z}{(1 - e^{-\alpha}z)^2} - \frac{z^{-1}}{(1 - e^{-\alpha}z^{-1})^2} \right)$$
- 2 pôles \rightarrow 2 sens de parcours (croissant, décroissant)



Stocker 1 ligne entière
4 mémoire lignes (mode ping pong)

Cany Deriche

Le facteur α définit la **largeur du filtre**, donc le compromis entre la détection et la localisation. Plus α est grand, plus on localise précisément le contour. Plus il est petit, plus on détecte facilement la présence des contours.

$$D(z) = k_D \left(\frac{z}{(1 - e^{-\alpha}z)^2} - \frac{z^{-1}}{(1 - e^{-\alpha}z^{-1})^2} \right)$$

Algorithme (sans gérer les bords)

Pour k de 0 à $N-1$

$$y_m(k) = p(k-1) + 2e^{-\alpha}y_m(k-1) - e^{-2\alpha}y_m(k-2)$$

Fin pour k

Pour k de $N-1$ à 0

$$y_p(k) = p(k+1) + 2e^{-\alpha}y_p(k+1) - e^{-2\alpha}y_p(k+2)$$

$$d(k) = (1 - e^{-\alpha})^2 (y_p(k) - y_m(k))$$

Fin pour k

2 +

2 x

3 +

3 x

5 ADD, 5 MUL

Cany Deriche

$$y_m(-2) = y_m(-1) = p(-1) = 0$$

Pour k de 0 à $N-1$

$$y_m(k) = p(k-1) + 2e^{-\alpha}y_m(k-1) - e^{-2\alpha}y_m(k-2)$$

Fin pour k

$$y_p(N+1) = y_p(N) = y_m(N-1) \text{ et } p(N) = k_D y_m(N-1)$$

Pour k de $N-1$ à 0

$$y_p(k) = p(k+1) + 2e^{-\alpha}y_p(k+1) - e^{-2\alpha}y_p(k+2)$$

Fin pour k

$$y_m(-2) = y_m(-1) = y_p(0) \text{ et } p(-1) = k_D y_p(0)$$

Pour k de 0 à $N-1$

$$y_m(k) = p(k-1) + 2e^{-\alpha}y_m(k-1) - e^{-2\alpha}y_m(k-2)$$

$$d(k) = (1 - e^{-\alpha})^2 (y_p(k) - y_m(k))$$

Fin pour k

14

Exemples

Image originale



Image traitée avec $\alpha = 0.5$



Image traitée avec $\alpha = 1$



Image traitée avec $\alpha = 10$

15

Deriche : Lisseur 1D

Lisseur de même résolution que dérivateur

$$L(z) = k_L \left[\frac{(\alpha + 1)e^{-\alpha}z - e^{-2\alpha}z^2}{(1 - e^{-\alpha}z)^2} + \frac{1 + (\alpha - 1)e^{-\alpha}z^{-1}}{(1 - e^{-\alpha}z^{-1})^2} \right]$$

$$y_m(-2) = y_m(-1) = p(-1) = 0$$

Pour k de 0 à $N-1$

$$y_m(k) = p(k) + (\alpha - 1)e^{-\alpha}p(k-1) + 2e^{-\alpha}y_m(k-1) - e^{-2\alpha}y_m(k-2)$$

Fin pour k

$$y_p(N+1) = y_p(N) = a y_m(N-1) \text{ et } p(N) = p(N+1) = b y_m(N-1)$$

Pour k de $N-1$ à 0

$$y_p(k) = (\alpha + 1)e^{-\alpha}p(k+1) - e^{-2\alpha}p(k+2) + 2e^{-\alpha}y_p(k+1) - e^{-2\alpha}y_p(k+2)$$

Fin pour k

$$y_m(-2) = y_m(-1) = c y_p(0) \text{ et } p(-1) = d y_p(0)$$

Pour k de 0 à $N-1$

$$y_m(k) = p(k) + (\alpha - 1)e^{-\alpha}p(k-1) + 2e^{-\alpha}y_m(k-1) - e^{-2\alpha}y_m(k-2)$$

$$l(k) = \frac{(1 - e^{-\alpha})^2}{1 + 2e^{-\alpha} - e^{-2\alpha}} (y_p(k) + y_m(k))$$

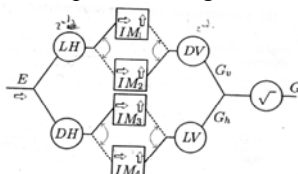
Fin pour k

7 ADD, 8 MUL

16

Deriche 2D

- $GH(z) = D(z)L(z^N)P(z)$
- $GV(z) = D(z^N)L(z)P(z)$
- \rightarrow ordre sans importance mais stockage intermédiaire important : ex dans G finir D avant L
- NB opérations : 26 mul et 24 ADD
- Exemple : 25 image $512^2 \Rightarrow 328$ MIPS



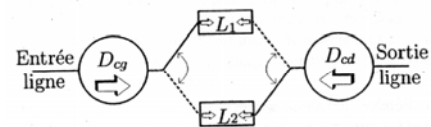
4 mémoires images (mode ping pong) 17

Optimisation Garcia-Lorca

$$\gamma = e^{-\alpha} \quad D(z) = k_D \left(\frac{z}{(1 - \gamma z)^2} - \frac{z^{-1}}{(1 - \gamma z^{-1})^2} \right)$$

$$D(z) = k'_D (z - z^{-1}) \frac{1}{(1 - \gamma z)^2} \frac{1}{(1 - \gamma z^{-1})^2}$$

\Leftrightarrow mise en cascade : puissance idem mais gain en mémoire ligne



18

Optimisation Garcia-Lorca

$$D(z) = k'_D (z - z^{-1}) \frac{1}{(1 - \gamma z)^2} \frac{1}{(1 - \gamma z^{-1})^2}$$

⇔

Pour k de 0 à $N - 1$

$$y_m(k) = p(k) + 2\gamma y_m(k - 1) - \gamma^2 y_m(k - 2)$$

Fin pour k

Pour k de $N - 1$ à 0

$$y_p(k) = y_m(k) + 2\gamma y_p(k + 1) - \gamma^2 y_p(k + 2)$$

$$d(k + 1) = (1 - \gamma)^2 (1 - \gamma^2) (y_p(k + 2) - y_p(k))$$

Fin pour k

19

Garcia Lorca : Lissage 1D

- La même transf. n'est pas intéressante, partir d'une simplification calcul (trapèze)

$$L'(z) = (z + 2 + z^{-1}) \frac{1}{(1 - \gamma z)^2} \frac{1}{(1 - \gamma z^{-1})^2}$$

Partie récursive idem au dérivateur

$D(z)$: partie non récursive → dérivateur de Sobel

$L'(z)$: partie non récursive → lisseur de Sobel

20

Garcia Lorca : optimisation

- Factorisation des parties non récursive

$$\frac{(z - z^{-1})}{(z + 2 + z^{-1})} = \frac{(1 - z^{-1})}{(1 + z^{-1})} \frac{(1 - z)}{(1 + z)}$$

Lisseur et
Dérivateur
Garcia Lorca

$$Dgl(z) = kgl_D (1 - z^{-1}) \frac{1}{(1 - \gamma z)^2} \frac{1}{(1 - \gamma z^{-1})^2}$$

$$Lgl(z) = kgl_L (1 + z^{-1}) \frac{1}{(1 - \gamma z)^2} \frac{1}{(1 - \gamma z^{-1})^2}$$

LGL(z)

21

Garcia Lorca : gradient 2D

$$GH(z) = Dgl(z) Lgl(z^N) P(z)$$

$$DH(z) = Dgl(z^N) Lgl(z) P(z)$$

$$\text{En posant : } LGL(z) = \frac{1}{(1 - \gamma z)^2} \frac{1}{(1 - \gamma z^{-1})^2}$$

On obtient :

$$GH(z) = kgl LGL(z) LGL(z^N) \frac{(1 - z^{-1})(1 + z^{-N})}{(1 - z^{-N})(1 + z^{-1})} P(z)$$

On pose :

$$R_h(z) = \frac{(1 - z^{-1})(1 + z^{-N})}{(1 - z^{-N})(1 + z^{-1})}$$

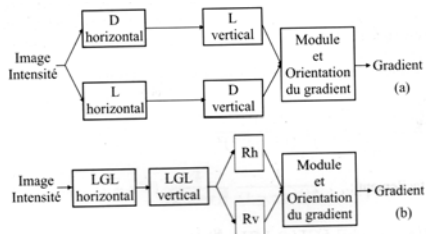
- Rh et Rv = filtres locaux : masques

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

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

22

Garcia Lorca : gradient 2D



Complexité de D et L équivalente à LGL et donc gain de 50%

23

Garcia Lorca : gradient 2D

Filtrage Horizontal : 4 x, 4 +

Pour i de 0 à $N - 1$ (chaque ligne)

$$th(i, -2) = th(i, -1) = 0$$

Pour k de 0 à $N - 1$

$$th(i, k) = p(i, k) + 2\gamma th(i, k - 1) - \gamma^2 th(i, k - 2)$$

Fin pour k

$$th(i, N + 1) = th(i, N) = th(i, N - 1)$$

Pour k de $N - 1$ à 0

$$th(i, k) = p(i, k) + 2\gamma th(i, k + 1) - \gamma^2 th(i, k + 2)$$

Fin pour k

$$th(i, -2) = th(i, -1) = th(i, 0)$$

Pour k de 0 à $N - 1$

$$th(i, k) = th(i, k) + 2\gamma th(i, k - 1) - \gamma^2 th(i, k - 2)$$

Fin pour k

Fin pour i

Fin pour j

Fin pour j

Fin pour j

Fin pour j

Fin pour j

Fin pour j

Fin pour j

Fin pour j

Fin pour j

Fin pour j

Fin pour j

Fin pour j

Fin pour j

Fin pour j

Fin pour j

Fin pour j

Fin pour j

Fin pour j

Fin pour j

Filtrage Vertical : 5 x, 4 +

Pour j de 0 à $N - 1$ (chaque colonne)

$$tv(-2) = tv(-1) = 0$$

Pour k de 0 à $N - 1$

$$tv(k) = th(k) + 2\gamma tv(k - 1) - \gamma^2 tv(k - 2)$$

Fin pour k

$$tv(N + 1) = tv(N) = tv(N - 1)$$

Pour k de $N - 1$ à 0

$$tv(k) = th(k) + 2\gamma tv(k + 1) - \gamma^2 tv(k + 2)$$

Fin pour k

$$tv(-2) = tv(-1) = tv(0)$$

Pour k de 0 à $N - 1$

$$tv(k) = kgl tv(k) + 2\gamma tv(k - 1) - \gamma^2 tv(k - 2)$$

Fin pour k

Fin pour j

Dérivation locale : 4 +

Pour j de 1 à $N - 1$ (chaque colonne)

Pour i de 1 à $N - 1$ (chaque ligne)

$$th(i, j) = tv(i, j) - tv(i, j - 1)$$

$$tv(i, j) = tv(i, j) + tv(i, j - 1)$$

$$gh(i, j) = th(i, j) + th(i - 1, j)$$

$$gv(i, j) = tv(i, j) - tv(i - 1, j)$$

Fin pour i

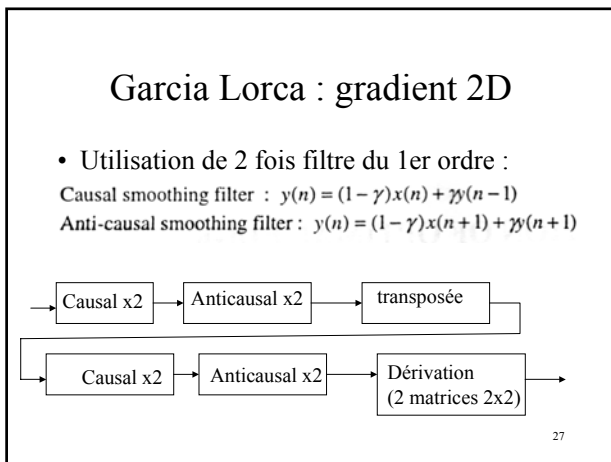
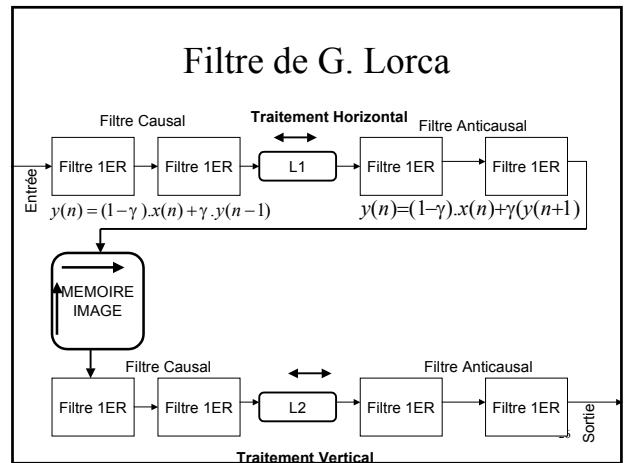
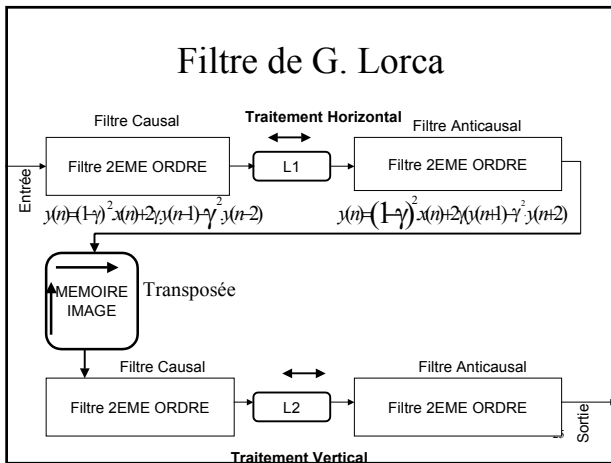
Fin pour j

9 MUL

12 ADD

138MPS (25images/sec, 512x512)

24



Bibliographie

- « Real time execution of optimal edge detectors on risc and DSP » L. Lacassagne, F. Lohier, P. Garda, IEEE ICASSP'98, Seattle, USA, 12-15 Mai 1998.
- « Méthodes et architectures pour le TSI en temps réel », édition Hermes/Lavoisier

The number 28 is located at the bottom right of the slide.