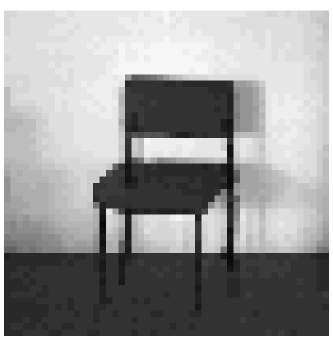
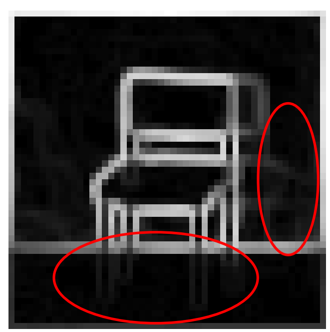
IX. Détection de contours

IX.1 Généralités

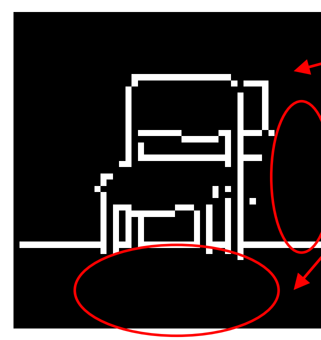
- Traitement pour accéder à des primitives de plus haut niveau
  - Pixels → Points de contours → ligne, bord → objet
- Nombreuses méthodes
  - <u>*Méthodes dérivatives*</u>, morphologiques, par EDP,...
- Plusieurs étapes pour la détection
  - Recherche des zones de fort contraste
  - Décision par seuillage : points « candidats au contours
  - Poursuite des contours (chaînage) : lignes de contours



Image



Zones de fort contraste



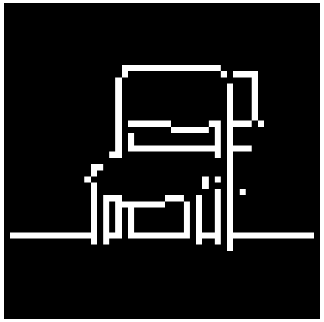
non détectés

Lignes de contour

IX. Détection de contours
IX.1 Généralités

- Des zones de fort contraste aux lignes de contour
  - Enchaînement de traitements

Image simple



Zones de fort contraste

Lignes de contour

IX. Détection de contours
IX.1 Généralités

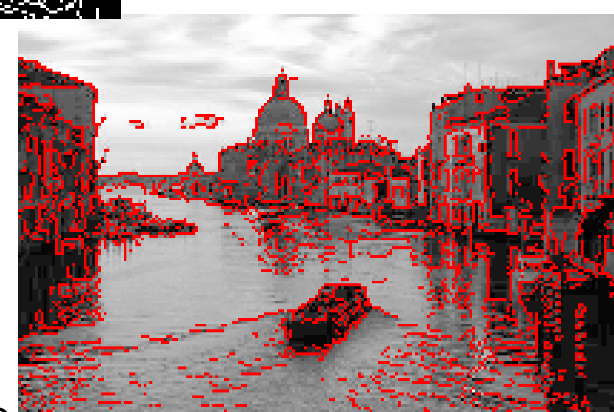
- Des zones de fort contraste aux lignes de contour
  - Enchaînement de traitements

Image complexe

points de contour

Zones de fort contraste

lignes de contour



IX. Détection de contours
IX.1 Généralités

- Méthodes dérivatives
  - Détection des Zones de Contraste 1D
    - Maximum du gradient (dérivée première)
    - Passage à zéro de la dérivée seconde
    - Équivalent au passage à zéro du Laplacien

1D → 2D

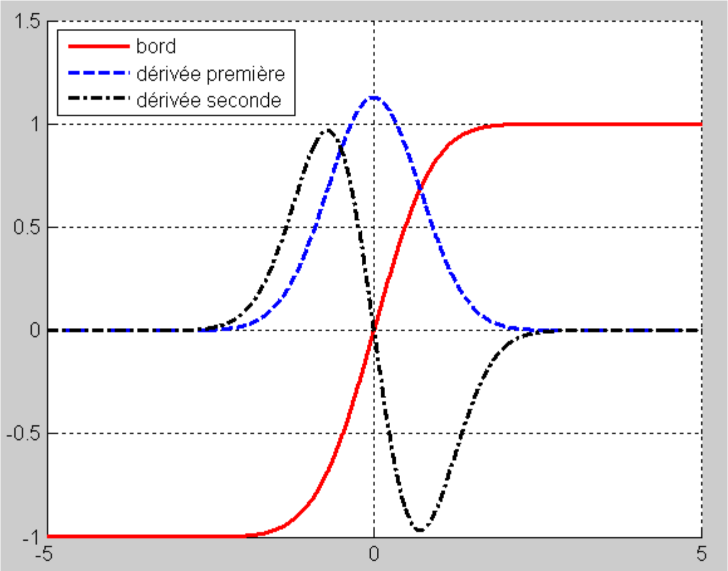


Illustration 1D , f(x)=bord; f'(x); f''(x)

- Détection des Zones de Contraste 2D
  - Norme du Gradient G(x,y)
    - Orientation du Gradient $\theta(x,y)$
    - Max du la norme G dans la direction θ
  - Passage à zéro de la dérivée seconde dans la direction θ
  - Passage à zéro du Laplacien

- Calcul des dérivées
  - Filtrage directe : Différences finies (Masque de convolution)
  - Filtrage récursif : Approche par filtrage optimal (cf. chap. X)

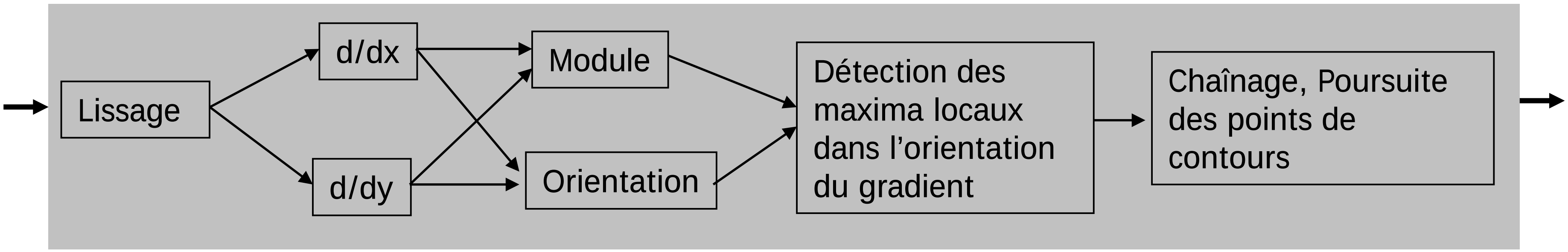
IX. Détection de contours

IX.1 Généralités

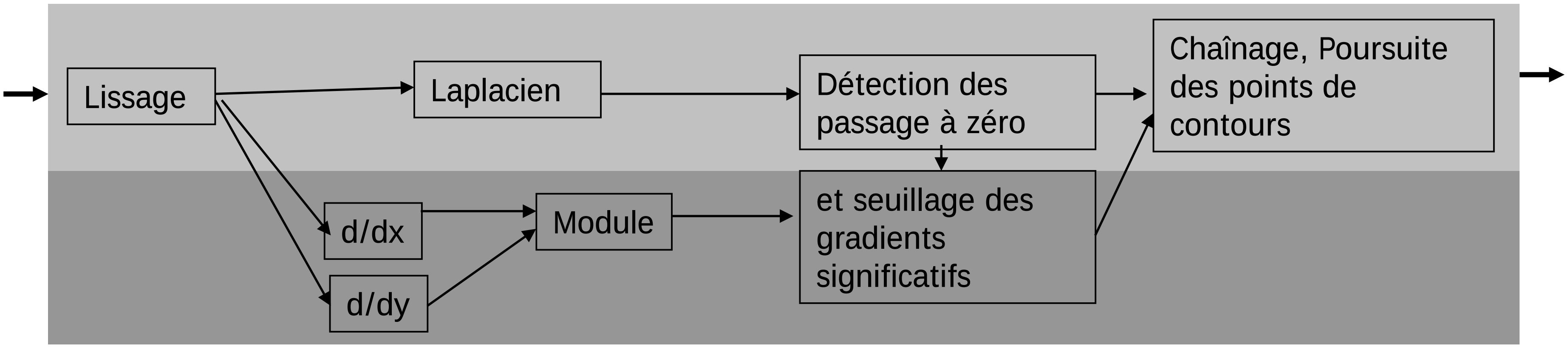
- Méthodes dérivatives
  - Font appel à un calcul de différence entre niveaux de gris sur des pixels voisins
  - Donc :
    - Sensibles aux bruit
  - Prétraitement des images
    - Réduction de bruit : lissage
    - Perte de précision dans la localisation des zones de contours
    - Compromis entre les 2 objectifs
      - Réduire le bruit
      - Garder une bonne localisation des contours
  - Filtre de détection de contours
    - Combinaison entre
      - lissage pour réduire le bruit et gradients (différences) pour accéder au contraste

IX. Détection de contours
IX.2 Principe général

- Schéma fonctionnel d’une détection de contours



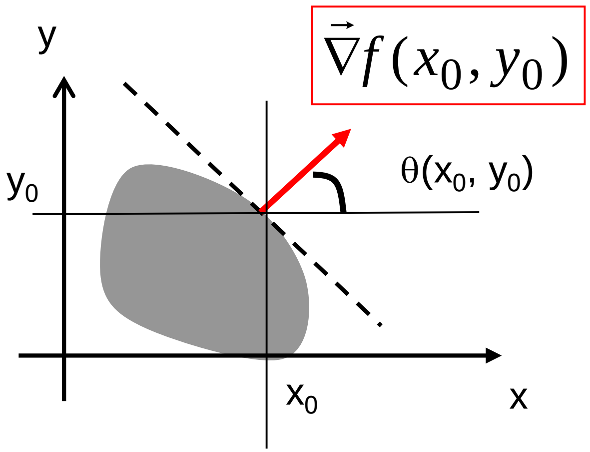
Gradient



Laplacien

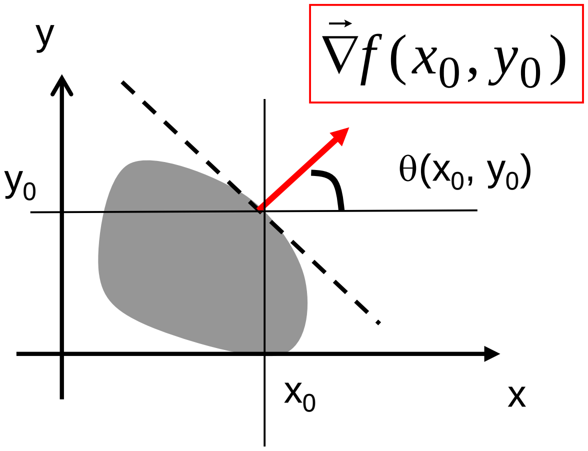
IX. Détection de contours
IX.3 Dérivées premières et secondes

- Modèle continu
  - Image f(x,y)
    - Dérivée première en x : $$g_x(x,y) = \frac{\partial f(x,y)}{\partial x}$$
    - Dérivée première en y : $$g_y(x,y) = \frac{\partial f(x,y)}{\partial y}$$
    - Vecteur Gradient : $$\vec{\nabla} f(x,y) = \vec{G}(x,y) = \left(\frac{\partial f(x,y)}{\partial x} \quad \frac{\partial f(x,y)}{\partial y} \right)'$$
      - Norme : $$\left|\vec{\nabla} f(x,y)\right| = \left|\vec{G}(x,y)\right| = \sqrt{g_x^2 + g_y^2}$$
      - Orientation : $$\theta(x,y) = arctg\left(\frac{g_y}{g_x}\right)$$



IX. Détection de contours
IX.3 Dérivées premières et secondes

- Modèle continu
  - Image f(x,y)
    - Vecteur Gradient :
      - Norme : Amplitude du contraste
      - Orientation : orthogonale au contour
    - Composante G_x
      - Détection des lignes <u>verticales</u>
    - Composante G_y
      - Détection des lignes <u>horizontales</u>



IX. Détection de contours
IX.3 Dérivées premières et secondes

- Modèle continu
  - Image f(x,y)
    - Dérivée seconde en x :
    - Dérivée seconde en y :
    - Laplacien :

$$g'_x(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2}$$

$$g'_y(x,y) = \frac{\partial^2 f(x,y)}{\partial y^2}$$

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

IX. Détection de contours

IX.3 Dérivées premières et secondes

- Modèle continu
  - Image f(x,y)
  - Opérateur Laplacien

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

    - Linéaire : une convolution : un exemple de noyau
    - Opérateur non directionnel

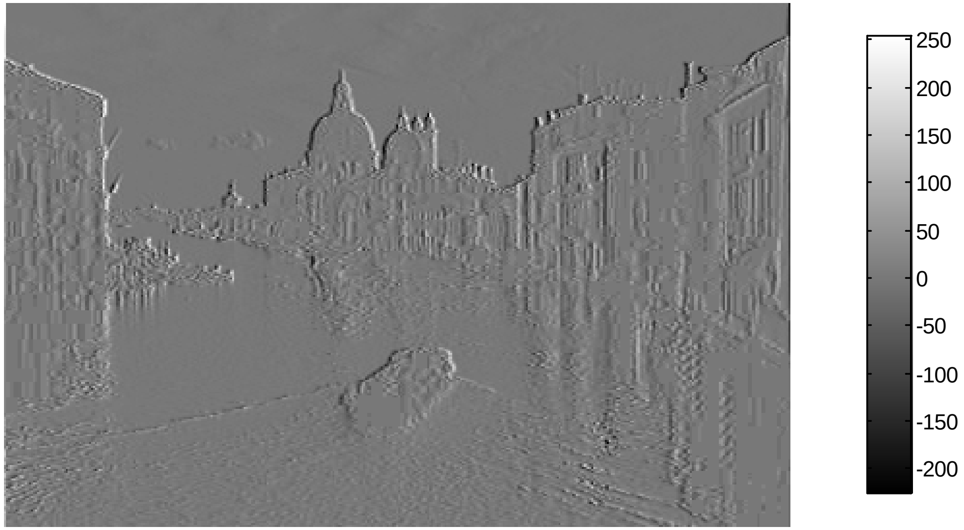
$$K_L = \frac{1}{4}\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

  - Opérateur : Norme du Gradient
    - Non linéaire : 2 convolutions, puissance de 2, racine de 2

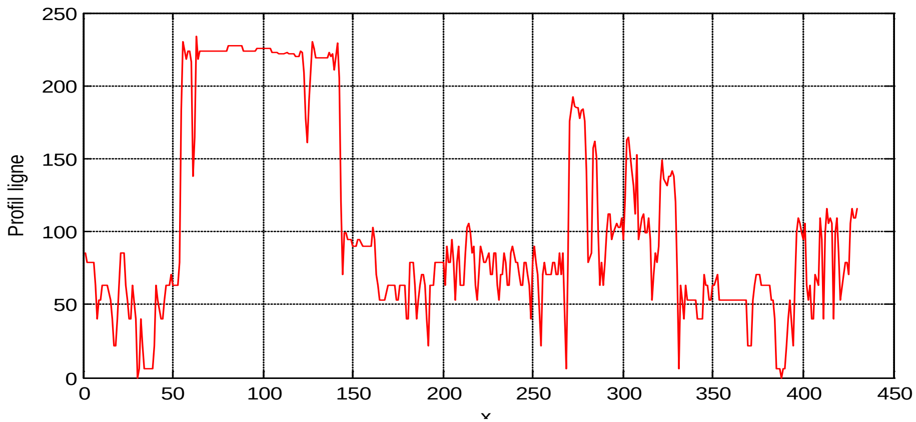
$$\left|\vec{\nabla} f(x,y)\right| = \sqrt{g_x^2(x,y) + g_y^2(x,y)}$$

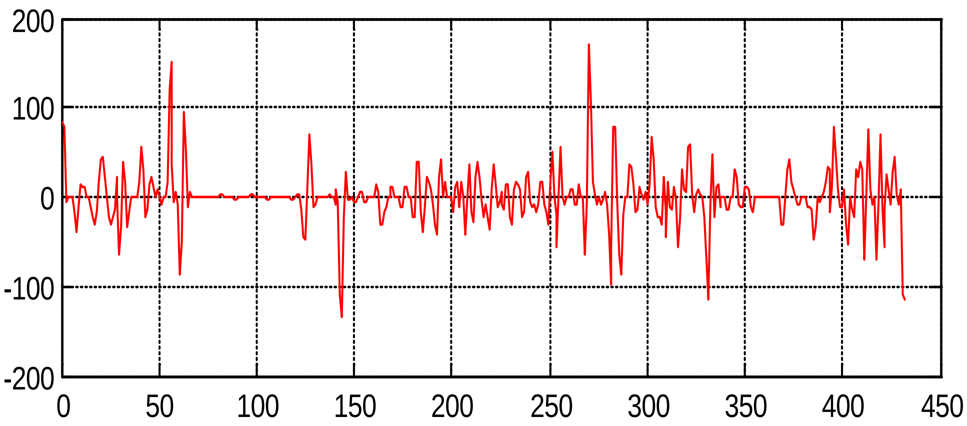
IX. Détection de contours
IX.3 Dérivées premières et secondes

- Calcul de la dérivée d'une image
  - Par définition : calcul local de différences de niveaux de gris entre le pixel courant et ses voisins
  - Fait ressortir le bruit (petites fluctuations non contrôlées) dans l'image f(x,y)
  - Illustration avec un opérateur gradient sur les lignes :
    - Gx =[1 0 -1] (A appliquer – convoluer- sur chaque ligne de l'image)

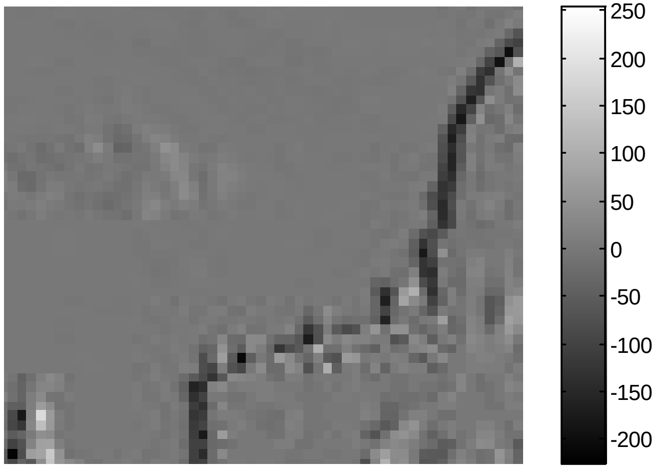


IX. Détection de contours
IX.3 Dérivées premières et secondes





Résultat de Gx sur un profil ligne

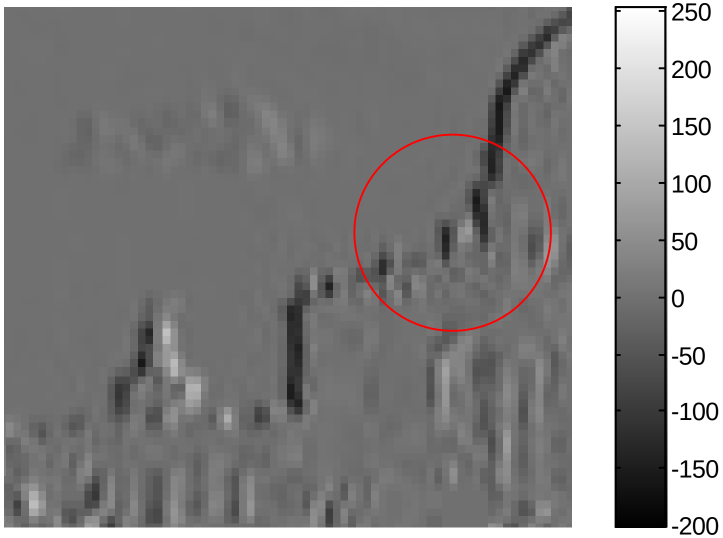


Un zoom sur l'image filtrée par Gx

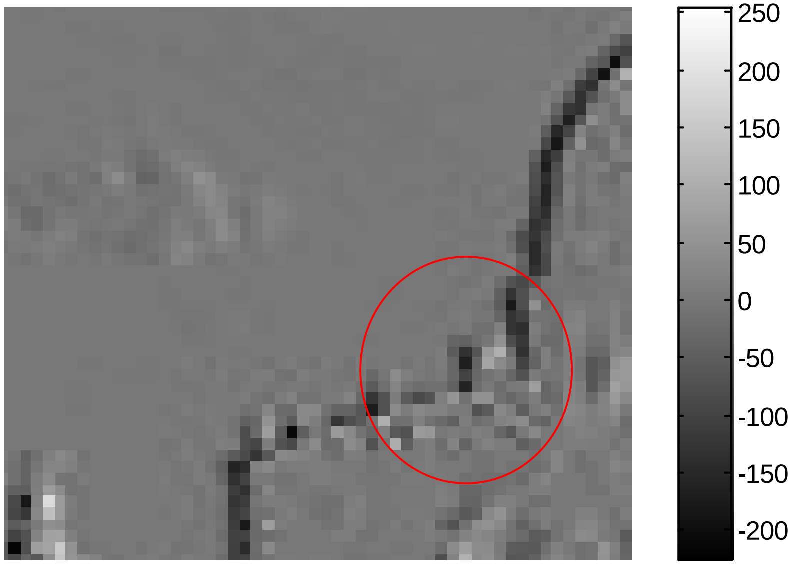
IX. Détection de contours
IX.3 Dérivées premières et secondes

- Calcul de la dérivée d'une image
  - Idée : Lissage de l'image pour retirer le bruit puis faire la dérivée
  - Comment combiner « lissage » et « dérivée » qui ont des objectifs opposés ?
    - Le lissage atténue les différences de niveaux de gris entre pixels voisins
    - Le gradient (dérivée première) calcule la différence de niveaux de gris entre les pixels voisins selon une direction donnée
  - Réponse
    - Faire le lissage dans la direction orthogonale du gradient
    - Rappel : L'opérateur « Gradient » est un opérateur directionnel

IX. Détection de contours
IX.3 Dérivées premières et secondes



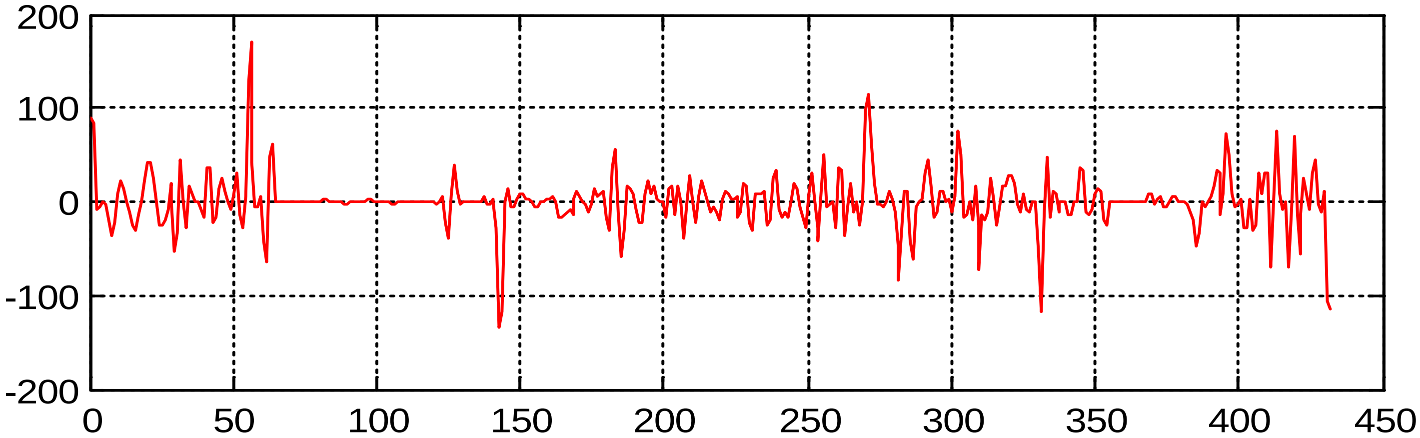
Un zoom sur l'image filtrée par un lissage en y, puis par Gx



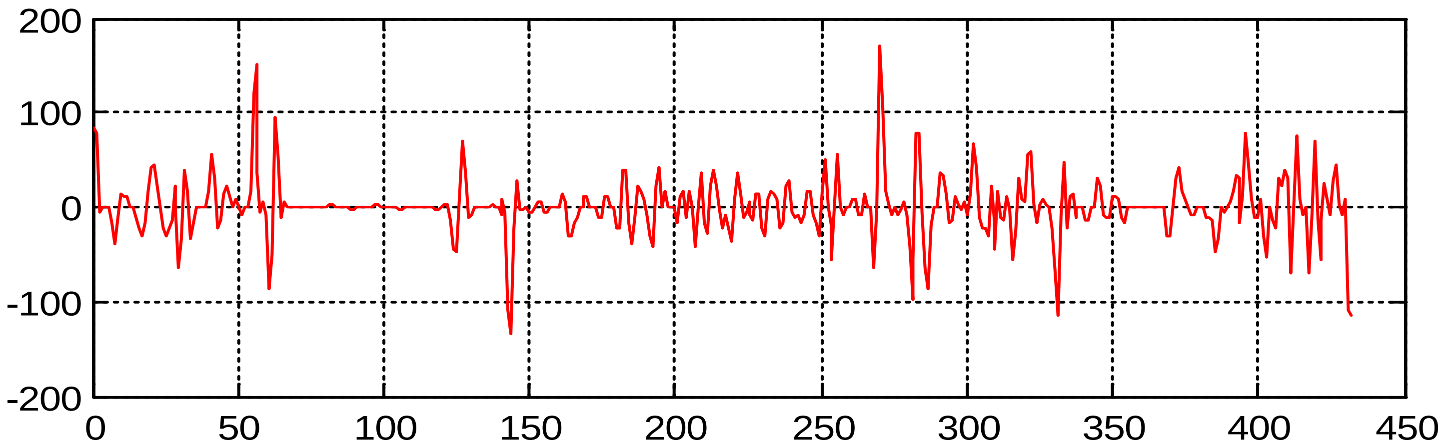
Un zoom sur l'image filtrée par Gx

IX. Détection de contours
IX.3 Dérivées premières et secondes

Profil ligne après Gx



Profil ligne après Lissage en y puis Gx



IX. Détection de contours
IX.3 Dérivées premières et secondes

- Dérivée d'une image filtrée
  - Image *f(x,y)* ,
  - Prétraitement par filtrage (réduction de bruit : lissage)
    - Noyau de convolution de lissage : *h(x,y)*
    - Image $f_l(x,y)$: Image après filtrage Passe-Bas de noyau *h(x,y)*
  - Gradient en x de l'image filtrée

$$f_l(x,y) = f(x,y) * h(x,y)$$

$$\frac{\partial f_l(x,y)}{\partial x} = \frac{\partial f(x,y) * h(x,y)}{\partial x} = \frac{\partial f(x,y)}{\partial x} * h(x,y) = \frac{\partial h(x,y)}{\partial x} * f(x,y)$$

$$= h_{dx}(x,y) * f(x,y)$$

$$\frac{\partial f_l(x,y)}{\partial y} = \frac{\partial h(x,y)}{\partial y} * f(x,y)$$

  - Idem pour le gradient en y

$$= h_{dy}(x,y) * f(x,y)$$

IX. Détection de contours

IX.3 Dérivées premières et secondes

- Dérivée d'une image filtrée
  - Image *f(x,y)* et image $f_l(x,y)$ après filtrage Passe-Bas
  - Comment combiner lissage et dérivée ?
    - Gradient en x : lissage en Y et dérivée en X
    - Noyau : h_{dx}

$$\boxed{\frac{\partial f_l(x,y)}{\partial x} = \frac{\partial h(x,y)}{\partial x} * f(x,y) = h_{dx}(x,y) * f(x,y)}$$

    - Gradient en y : lissage en X et dérivée en Y
    - Noyau : h_{dy}

$$\boxed{\frac{\partial f_l(x,y)}{\partial y} = \frac{\partial h(x,y)}{\partial y} * f(x,y) = h_{dy}(x,y) * f(x,y)}$$

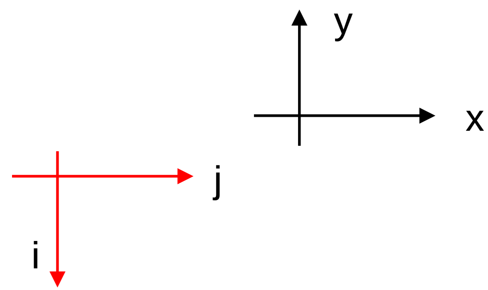
IX. Détection de contours
IX.3 Dérivées premières et secondes

- Dérivée d'une image filtrée
  - Image f(x,y)
  - Prétraitement par lissage (réduction de bruit)
    - Noyau de convolution symétrique <u>séparable</u> h(x,y)
    - 2 Noyaux 1D : lissage 1D et gradient direction ⊥
  - Gradient du noyau de convolution
    - Gradient en x : $\underbrace{\frac{dh_x(x)}{dx}}_{\text{Dérivée}} \; \underbrace{h_y(y)}_{\text{Lissage}}$
    - Gradient en y : $\underbrace{\frac{dh_y(y)}{dy}}_{\text{Dérivée}} \; \underbrace{h_x(x)}_{\text{Lissage}}$
  - Exemple

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

IX. Détection de contours
IX.4 Opérateurs Numériques

- Opérateurs Gradient
- Rappel de notations
  - En continu
  - En discret



- Approximation du gradient en numérique

$$\frac{\partial I(x,y)}{\partial x} \approx \frac{\Delta I[i,j]}{\Delta j} = I[i,j+1] - I[i,j]$$

Opérateur non symétrique

(ROBERTS)

$$\frac{\partial I(x,y)}{\partial x} \approx \frac{\Delta I[i,j]}{\Delta j} = \frac{1}{2}\left(I[i,j+1] - I[i,j-1]\right)$$

Mieux :

Opérateur symétrique

  - Noyau de convolution 1D
    - Gradient x
    - Gradient y

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

$$\frac{1}{2}\begin{pmatrix}1 \\ 0 \\ -1\end{pmatrix}$$

IX. Détection de contours

IX.4 Opérateurs Numériques

- Opérateurs Dérivée et Lissage

En x (ligne)

$$\begin{pmatrix} 1 & 0 & -1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 2 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$$

En y (colonne)

$$\begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} \qquad \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} \qquad \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

Gradient en x lissé

$$\begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix}$$

Gradient en y lissé

$$\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$$

IX. Détection de contours
IX.4 Opérateurs Numériques

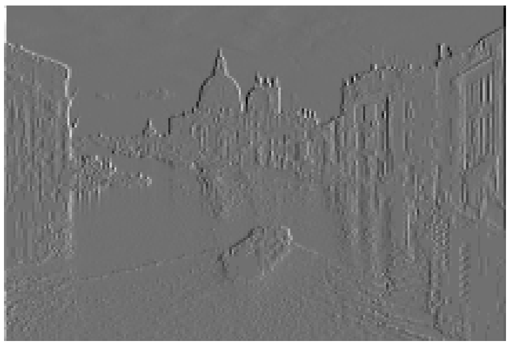
- Opérateurs Gradient
- Association lissage + dérivée dans les directions orthogonales
  - Différents noyaux en fonction de cette combinaison
  - Les plus utilisés sur 2 orientations

	Gx	Gy
PREWITT	$\frac{1}{3}\begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix}$	$\frac{1}{3}\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$
SOBEL	$\frac{1}{4}\begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}$	$\frac{1}{4}\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$
SCHARR	$\frac{1}{16}\begin{pmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{pmatrix}$	$\frac{1}{16}\begin{pmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{pmatrix}$

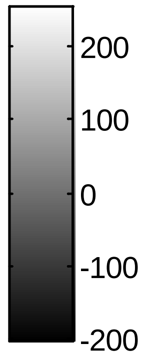
IX. Détection de contours
IX.4 Opérateurs Numériques

- Illustration

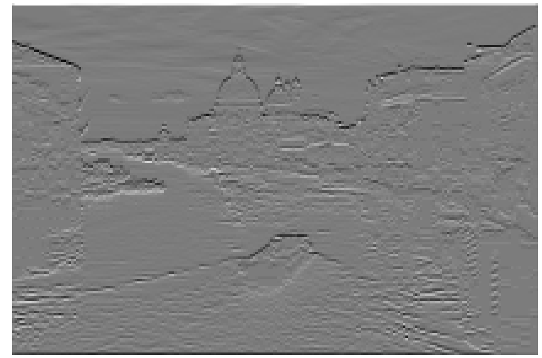
Gradient x lissé



$$g_x(x,y)$$



Gradient y lissé



$$g_y(x,y)$$

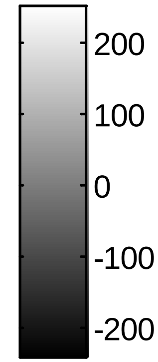
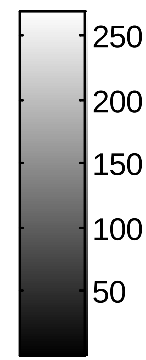


image originale

Norme Gradient



$$\left|\vec{\nabla} f(x,y)\right| = \sqrt{g_x^2(x,y) + g_y^2(x,y)}$$

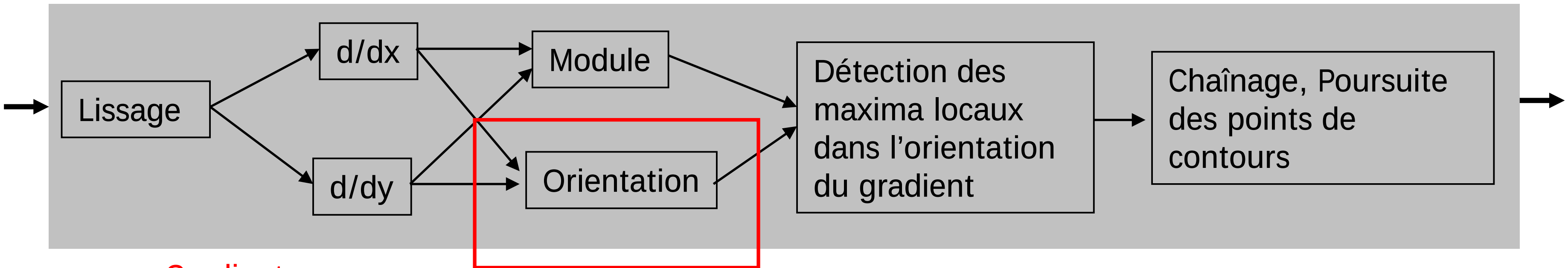
IX. Détection de contours
IX.4 Opérateurs Numériques

- Opérateurs Gradient
- Association lissage + différence
  - Les noyaux à différences finies les plus utilisés sont des noyaux 3×3
  - Détection à une échelle spatiale faible
    - Petites régions spatiales
  - Lissage à une échelle spatiale faible
    - Moyenne sur des petites régions → Lissage du bruit plutôt HF
- Si plus de réduction de bruit
  - Lissage sur une fenêtre spatiale plus grande
    - Exemple en 1D : [1 4 6 4 1]/16
  - Si combinaison avec le même opérateur gradient [1 0 -1]
    - Différence d'échelle entre gradient et lissage
    - Pour remédier à cela : approche par filtrage optimal (chap. X)
      - Un seul paramètre de réglage : α
      - α paramètre d'échelle à la fois pour le lissage et pour la dérivation

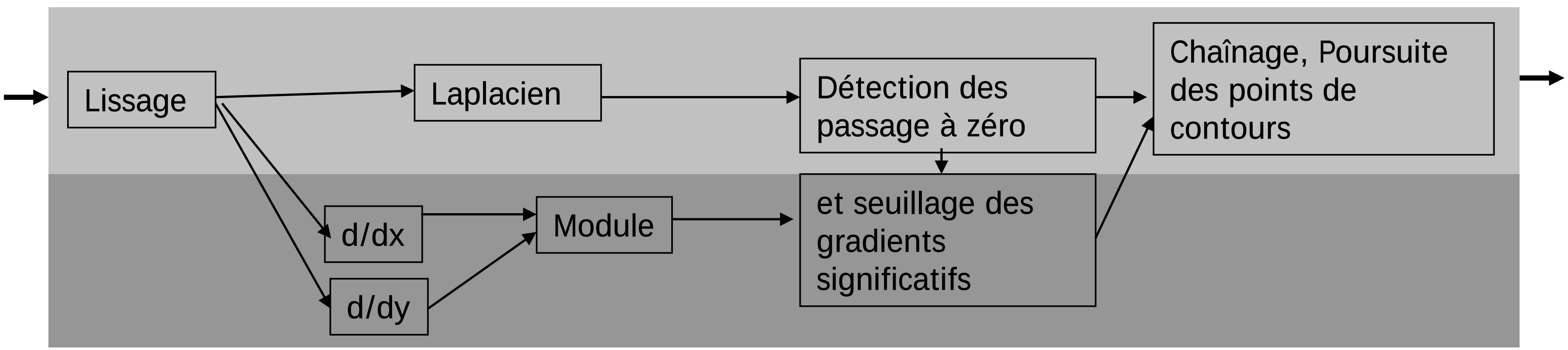
$$\frac{1}{16}\begin{pmatrix} 1 & 0 & -1 \\ 4 & 0 & -4 \\ 6 & 0 & -6 \\ 4 & 0 & -4 \\ 1 & 0 & -1 \end{pmatrix}$$

IX. Détection de contours
IX.4 Opérateurs numériques

- Rappel du schéma fonctionnel



Gradient



Laplacien

IX. Détection de contours

IX.4 Opérateurs Numériques

- Opérateur Gradient multidirectionnel
- 2 stratégies pour le calcul de l'orientation
  - Avec 2 mesures orthogonales de gradient
    - Gradient en x, Gradient en y, calcul de θ

$$I_x[i,j] = g_x * I[i,j] \qquad I_y[i,j] = g_y * I[i,j] \qquad \theta[i,j] = arctg\left(\frac{I_y[i,j]}{I_x[i,j]}\right)$$

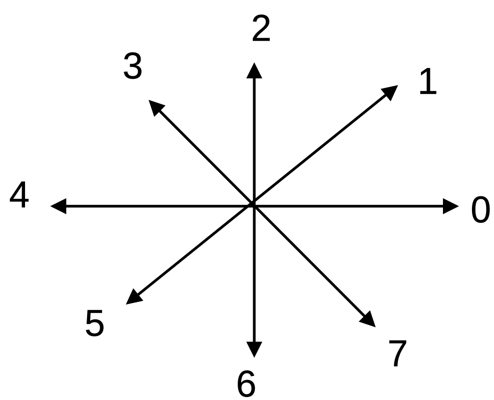
  - Avec Gradients suivant des directions multiples
    - Gradient à l'orientation θ_k : $G_{\theta k}$, k=0 à N_θ
      - Orientation : orientation du maximum des gradients $G_{\theta k}$
      - Gradient : Valeur maximale entre les gradients $G_{\theta k}$

$$|G[i,j]| = \left[\underset{k=0...7}{Max}\left(G_{\theta k}[i,j]\right)\right] \qquad \theta[i,j] = Arg\left[\underset{k=0...7}{Max}\left(G_{\theta k}[i,j]\right)\right]$$

IX. Détection de contours

IX.4 Opérateurs Numériques

- Opérateur Gradient multidirectionnel
- Exemple : Opérateurs de KIRSH
  - 8 orientations notées suivant le codage de Freeman



Gx : K_0

$$\frac{1}{15}\begin{pmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{pmatrix}$$

G_{45} : K_1

$$\frac{1}{15}\begin{pmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{pmatrix}$$

Gy : K_2

$$\frac{1}{15}\begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{pmatrix}$$

Etc ...

$$|G[i,j]| = \left[\underset{k=0...7}{Max} \left(G_{\theta_k}[i,j] \right) \right]$$

$$\theta[i,j] = Arg\left[\underset{k=0...7}{Max} \left(G_{\theta_k}[i,j] \right) \right] \times 45°$$

IX. Détection de contours
IX.4 Opérateurs Numériques

- Opérateurs Dérivée Seconde
- Approximation en numérique
  - Même approche que pour le gradient
  - Mais beaucoup plus sensible aux bruits que le Gradient

$$\frac{\partial^2 I(x,y)}{\partial x^2} = \frac{\partial\ I_x(x,y)}{\partial x} \approx \left(-I[i,j+1] + 2I[i,j] - I[i,j-1]\right)$$

$$\frac{\partial^2 I(x,y)}{\partial y^2} = \frac{\partial\ I_y(x,y)}{\partial y} \approx \left(-I[i+1,j] + 2I[i,j] - I[i-1,j]\right)$$

IX. Détection de contours
IX.4 Opérateurs Numériques

- Opérateurs Dérivée Seconde
- Approximation en numérique
  - Noyau de convolution 1D
    - Dérivée seconde x $\rightarrow \begin{pmatrix} -1 & 2 & -1 \end{pmatrix}$
    - Dérivée seconde y $\rightarrow \begin{pmatrix} -1 \\ 2 \\ -1 \end{pmatrix}$
  - Combinaison avec le lissage
    - Indispensable , car très sensible aux bruits

$$\frac{\partial^2 h(x,y)}{\partial x^2} \quad \frac{1}{4}\begin{pmatrix} -1 & 2 & -1 \\ -2 & 4 & -2 \\ -1 & 2 & -1 \end{pmatrix} \qquad \frac{\partial^2 h(x,y)}{\partial y^2} \quad \frac{1}{4}\begin{pmatrix} -1 & -2 & -1 \\ 2 & 4 & 2 \\ -1 & -2 & -1 \end{pmatrix}$$

IX. Détection de contours

IX.4 Opérateurs Numériques

- Opérateurs Laplacien
- Approximation en numérique
  - Même approche que pour le gradient

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

  - Noyau de convolution 2D
  - Toujours Combinaison avec le lissage

$$K_L = \frac{1}{4}\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad K_L = \frac{1}{8}\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad K_L = \frac{1}{12}\begin{bmatrix} 1 & 2 & 1 \\ 2 & -12 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

IX. Détection de contours
IX.5 Points de contour

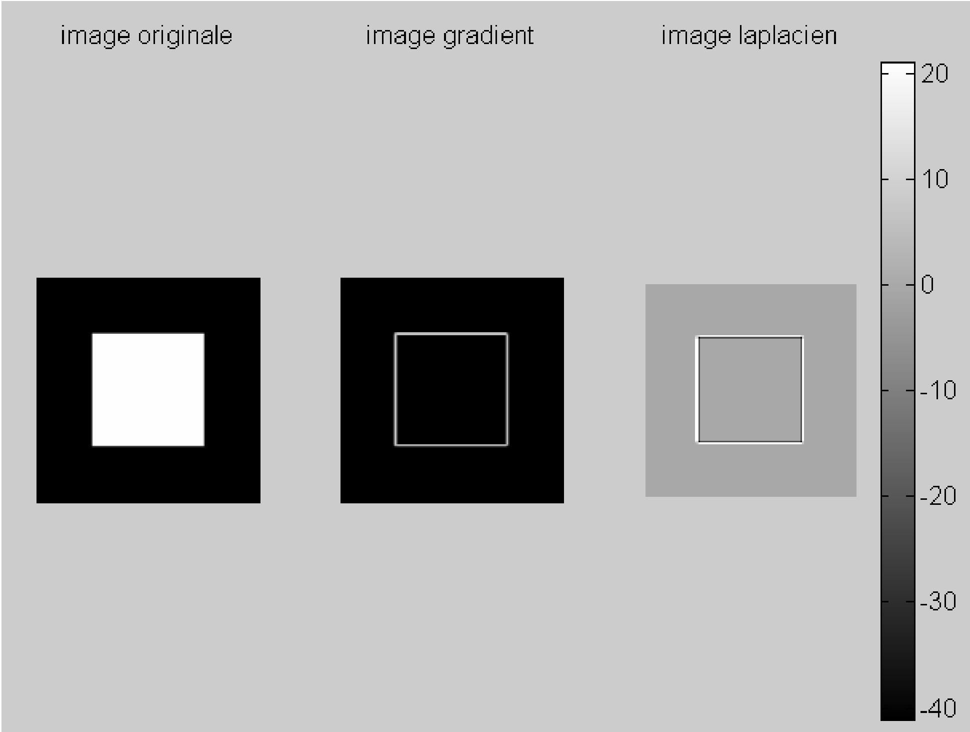
- A partir des opérateurs aux différences finies pour calculer Gradient ou Laplacien
- Mais ce ne sont pas nécessairement des points de contour
  - On veut obtenir une carte binaire
    - C(i,j)=1 : pixel(i,j) est un contour ; C(i,j)=0 : pixel(i,j) n'est pas un contour

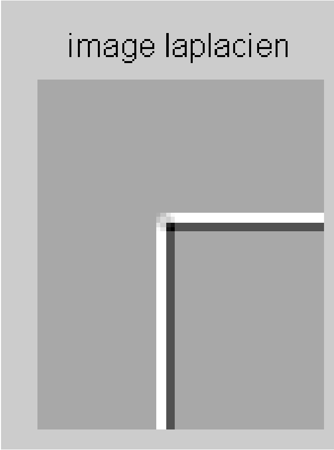
IX. Détection de contours
IX.5 Points de contour

- A partir des images de Gradient et/ou de Laplacien
  - Mise en place de techniques de seuillage
    - Double seuillage ou seuillage par hystéresis
  - Obtention de points non structurés
    - Qualité médiocre pour les utiliser tels quels comme contour
      - bruités
      - épais
      - Interrompus
      - Pas nécessairement fermés
  - Passage des points détectés aux points de contours
    - Post-traitements plus ou moins complexes
      - Seuillage
      - Autres filtrages
      - Chaînage
      - ....

IX. Détection de contours
IX.5 Points de contour

- A partir des opérateurs aux différences finies pour calculer Gradient ou Laplacien
- Dans le cas idéal : sans bruit
  - Post-traitement inutile



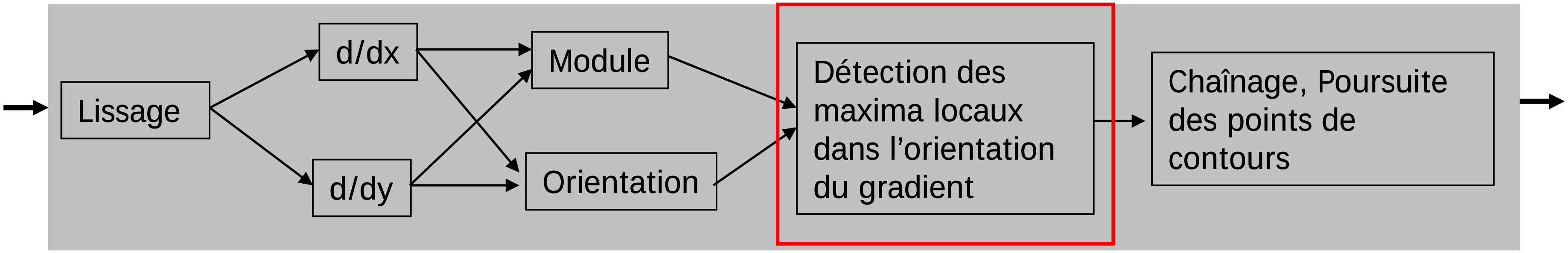


zoom

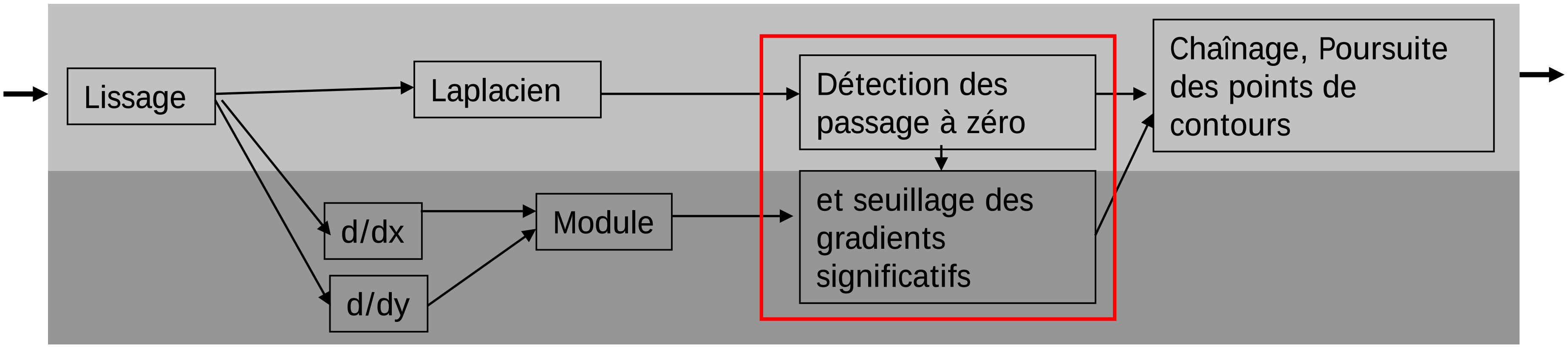
IX. Détection de contours

IX.5 Points de contour

- Rappel du schéma fonctionnel

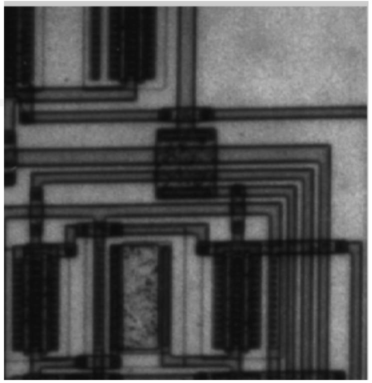


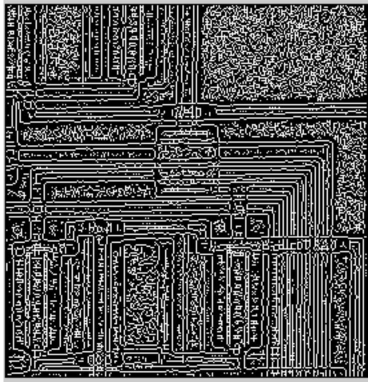
Gradient



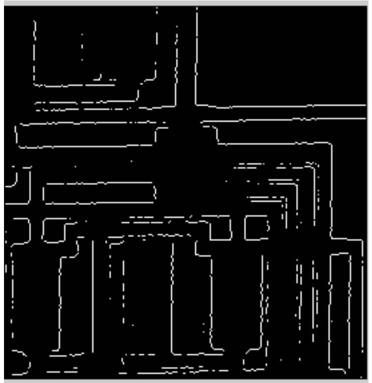
Laplacien

IX. Détection de contours
IX.5 Points de contour





- Points de contour candidats
  - Gradient
    - Maxima locaux dans la direction du gradient
  - Laplacien
    - Passage à zéro du Laplacien
    - Pas de calcul d'orientation
  - Rappel : sensibilité aux bruits
- Amélioration de la robustesse par seuillage
  - Contours significatifs
- On obtient un ensemble non structuré de points de contour
  - Puis chaînage des points de contour



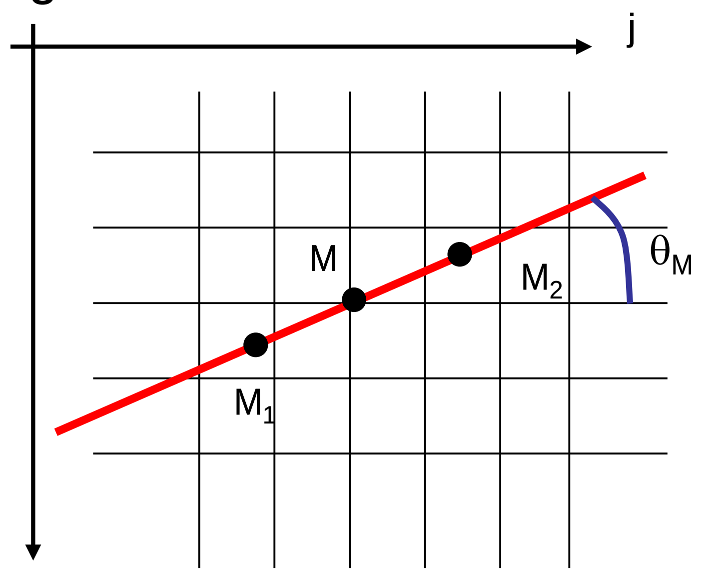
IX. Détection de contours
IX.5 Points de contour

- Points de contour candidats
  - Gradient
    - Maxima locaux dans la direction du gradient

M[i_0, j_0] : pixel courant de coordonnées [i_0, j_0]

M1 : voisin « arrière » dans la direction du gradient de M

M2 : voisin « avant » dans la direction du gradient de M



Coordonnées des voisins “avant” et “arrière”

θ_M : Orientation du gradient en M

d : Paramètre à fixer (norme de la distance aux premiers voisins) , typiquement 1 ou 2

M1[i_1, j_1] : $i_1 = i_0 - d.\sin(\theta_M)$; $j_1 = j_0 + d.\cos(\theta_M)$

M2[i_2, j_2] : $i_2 = i_0 + d.\sin(\theta_M)$; $j_2 = j_0 - d.\cos(\theta_M)$

$\longrightarrow$ M1 et M2 ne sont pas nécessairement sur la grille

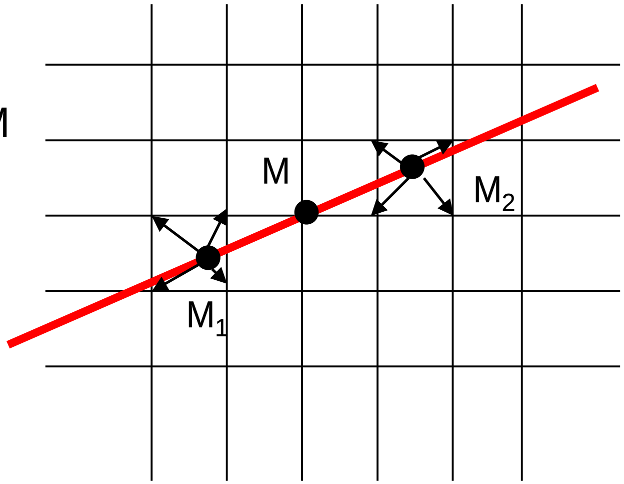
IX. Détection de contours
IX.5 Points de contour

- Points de contour candidats
  - Gradient
    - Maxima locaux dans la direction du gradient

M : pixel courant

M1 : voisin « arrière » dans la direction du gradient de M

M2 : voisin « avant » dans la direction du gradient de M



M est un maximum local:

$|G(M)| > |G(M_1)|$ et $|G(M)| > |G(M_2)|$

Difficulté :

M1 et M2 ne sont pas nécessairement sur la grille

→ Calculer leur gradient par interpolation linéaire ou affectation au point le plus proche

IX. Détection de contours
IX.5 Points de contour

- Points de contour candidats
  - Gradient
    - Maxima locaux dans la direction du gradient
    - Solution approchée
      - Quantification de la valeur de l'orientation avec par exemple, les masques de KIRSH

$$\theta[i,j] = Arg\left[\underset{k=0\ldots7}{Max}\left(I_{\theta k}[i,j]\right)\right]$$

      - Les points M1 et M2 sont sur la grille, car les orientations possibles sont : 0°, 45°, 90° et 135°

IX. Détection de contours
IX.5 Points de contour

- Points de contour candidats
  - Gradient
    - Maxima locaux dans la direction du gradient
    - Seuillage des maxima significatifs
      - Méthode par seuil à hystérésis = double seuillage
        - Seuil haut, Seuil bas

|Grad(pc)| ≥ seuil_haut → pc pixel courant est point de contour (PC)

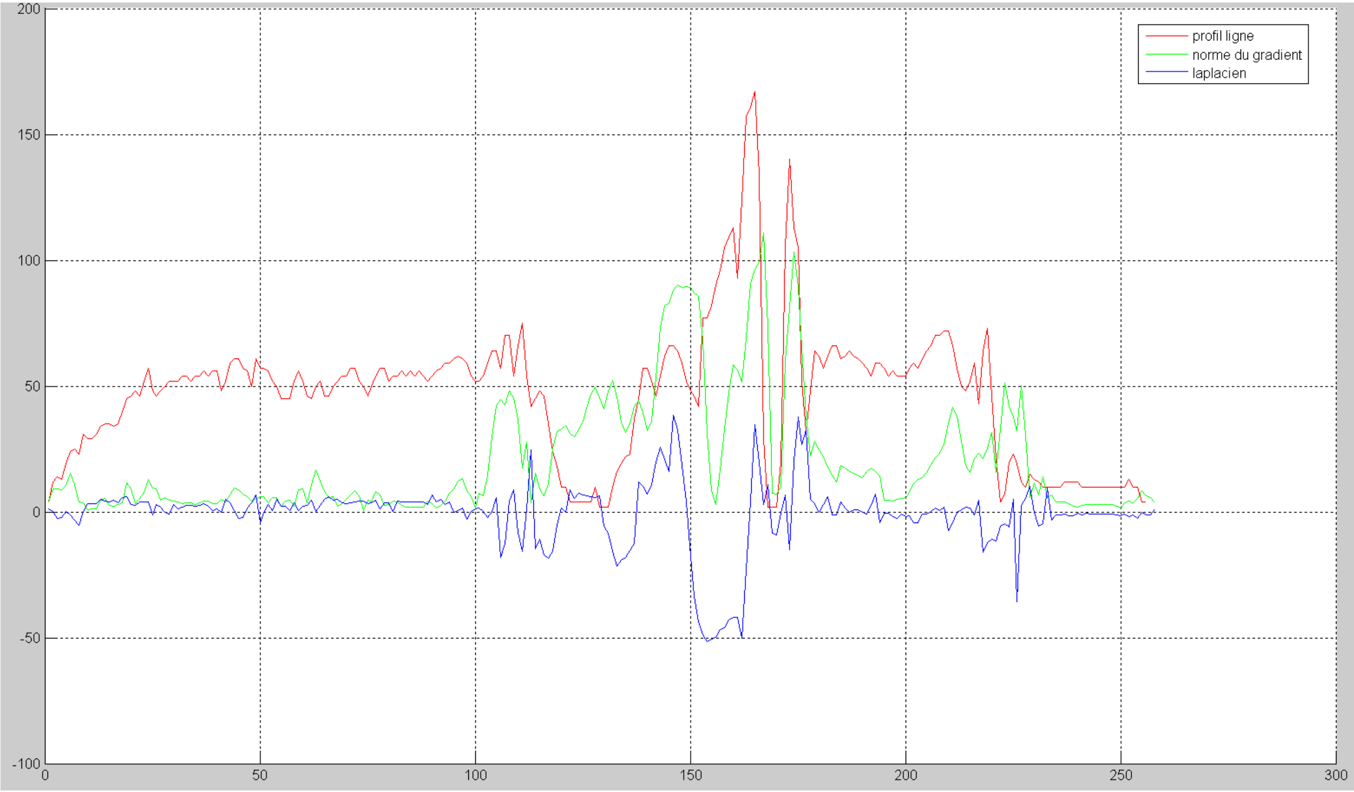
|Grad(pc)| < seuil_bas → pc pixel courant n'est pas un point de contour (NPC)

seuil_bas ≤|Grad(pc)| < seuil_haut → pc pixel courant est un point de contour possible (PCP)

      - Réaffectation des points de contour possible par propagation
        - Technique de chaînage

IX. Détection de contours
IX.5 Points de contour

- Points de contour candidats
  - Laplacien
    - Passage à zéro du Laplacien



En théorie : bonne précision spatiale de détection

En pratique : effet du bruit et détection de petites régions non significatives

IX. Détection de contours
IX.5 Points de contour

- Points de contour candidats
  - Laplacien
    - Détection du Passage à zéro du Laplacien
    - Masque de voisinage à 4 premiers voisins

	v1	
v2	pixc	v3
	v4	

lvi : valeur du Laplacien aux points vi

pc pixel courant est point de contour si

Signe(lv2)≠Signe(lv3) et |lv2.lv3|>0 et |lv2-lv3|>seuil

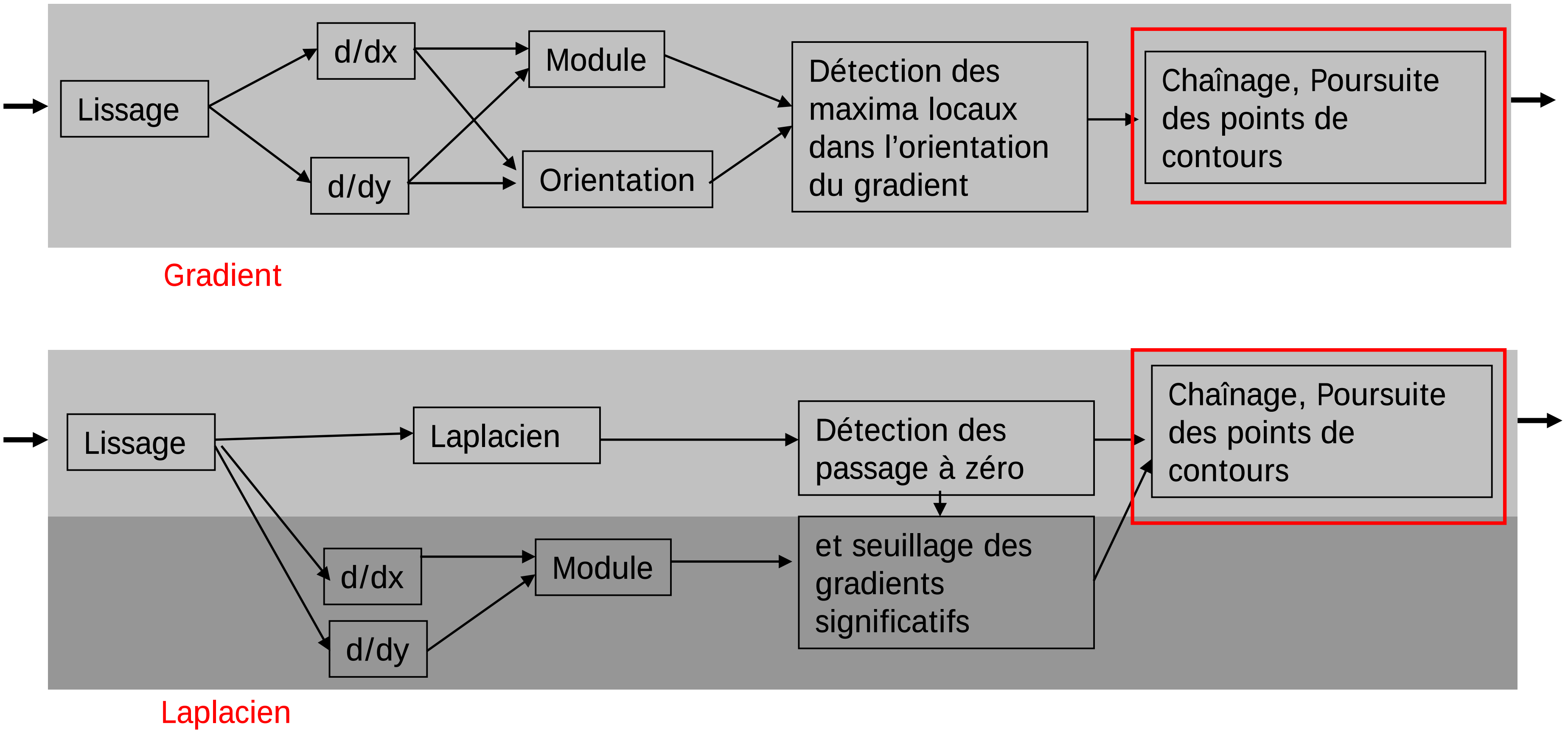
ou

Signe(lv1)≠Signe(lv4) et |lv1.lv4|>0 et |lv1-lv4|>seuil

IX. Détection de contours

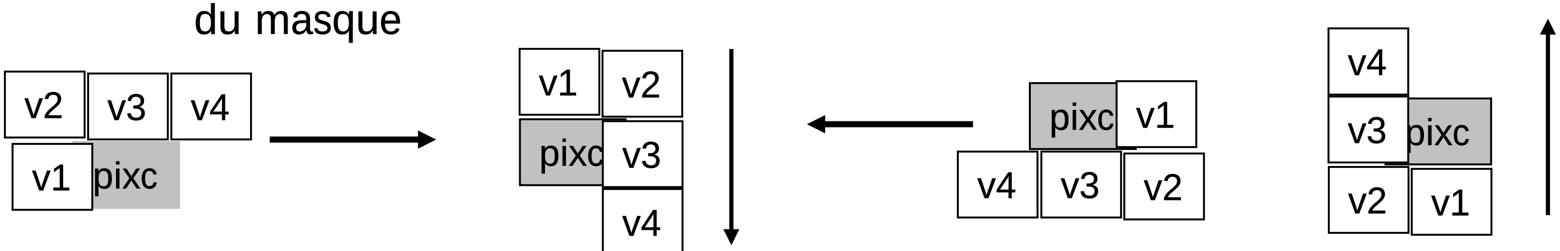
IX.6 Chaînage des points de contour

- Rappel du schéma fonctionnel



IX. Détection de contours
IX.6 Chaînage des points de contour

- Un exemple d'algorithme de chaînage
  - Double seuillage :
    - Pixel : soit PC, NPC, PCP
    - 4 balayages successifs
      - Chaque masque définit un voisinage
      - A chaque masque est associé un sens de balayage
      - Pour passer d'un sens de balayage à un autre : flip horizontal et flip vertical du masque



  - Pour chaque balayage

Si pixc est PCP, alors

si ∃ pix ∊ Voisin(pixc) / pix est PC alors pixc est PC

Sinon pixc reste dans la même catégorie