

Algorithmique des images, textes et données

3. Détection de contours

David Parello

david.parello@univ-perp.fr

Université de Perpignan Via Domitia

S4 Licence 2023-2024



Approche naïve

- ▶ Détecteurs de contours horizontaux et verticaux
- ▶ Limitations de l'approche naïve

Approche par convolution

- ▶ Rappels sur la convolution
- ▶ Liens avec la dérivée et lissage des hautes fréquences
- ▶ Filtres de Prewitt et Sobel

Détecteur optimal : filtre de Canny

- ▶ Qualité attendues d'un détecteur de contours
- ▶ Réduction du bruit avec un filtrage gaussien
- ▶ Calcul du Gradient
- ▶ Suppression des non maxima
- ▶ Seuillage des contours
- ▶ Traquage par hysteresis

Approche naïve

- ▶ Détecteurs de contours horizontaux et verticaux
- ▶ Limitations de l'approche naïve

Approche par convolution

- ▶ Rappels sur la convolution
- ▶ Liens avec la dérivée et lissage des hautes fréquences
- ▶ Filtres de Prewitt et Sobel

Détecteur optimal : filtre de Canny

- ▶ Qualité attendues d'un détecteur de contours
- ▶ Réduction du bruit avec un filtrage gaussien
- ▶ Calcul du Gradient
- ▶ Suppression des non maxima
- ▶ Seuillage des contours
- ▶ Traquage par hysteresis

Détecteurs de contours horizontaux et verticaux

- Commençons par une image en noir et blanc présentant une seule discontinuité.



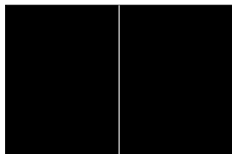
- Le contour est formé des pixels blancs dont le voisin de gauche est noir et des pixels noirs dont le voisin de gauche est blanc.
 - ▶ le contour est épais de deux pixels
 - ▶ par simplicité on fait le choix de garder seulement le voisin de gauche
- On construit une nouvelle image I_v telle que

$$I_v(i,j) = I(i,j) - I(i,j-1)$$

Détecteurs de contours horizontaux et verticaux

- On construit une nouvelle image I_v telle que

$$I_v(i,j) = I(i,j) - I(i,j-1)$$



- les pixels au bord ne sont pas traités

```
for(int i = 1 ; i < height -1 ; i++){  
    for(int j = 1 ; j < width -1 ; j++){  
        . . . .
```

- les pixels de I_v ont une valeur comprise entre -255 et 255 \rightarrow on prend la valeur absolue

- Notre détecteur ne détecte que les frontières verticales
 - Pour les frontières horizontales on utilisera une deuxième image

$$I_h(i,j) = I(i,j) - I(i-1,j)$$

Détecteurs de contours horizontaux et verticaux

- À partir de l'image d'origine on obtient donc deux images

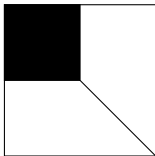
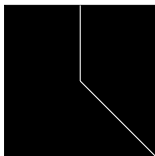
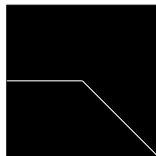


Image d'origine I



I_v

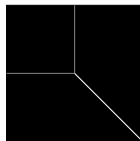


I_h

- On va utiliser la norme euclidienne du vecteur $[I_h, I_v]$ pour obtenir les contours de notre image

$$I_f(i,j) = \sqrt{I_h(i,j)^2 + I_v(i,j)^2}$$

- Les valeurs obtenues sont comprises entre 0 et $255\sqrt{2}$, il faut donc renormaliser la valeur entre 0 et 255.



I_f

Approche naïve

- ▶ Détecteurs de contours horizontaux et verticaux
- ▶ Limitations de l'approche naïve

Approche par convolution

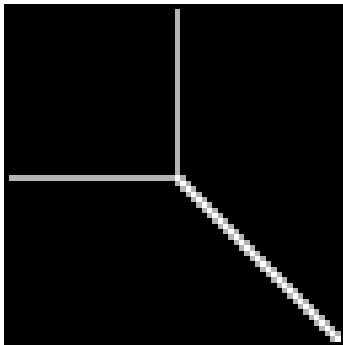
- ▶ Rappels sur la convolution
- ▶ Liens avec la dérivée et lissage des hautes fréquences
- ▶ Filtres de Prewitt et Sobel

Détecteur optimal : filtre de Canny

- ▶ Qualité attendues d'un détecteur de contours
- ▶ Réduction du bruit avec un filtrage gaussien
- ▶ Calcul du Gradient
- ▶ Suppression des non maxima
- ▶ Seuillage des contours
- ▶ Traquage par hysteresis

Limitations de l'approche naïve

- En zoomant sur l'image précente on remarque que



- ▶ Les contours du carré apparaissent gris ($180 \approx 255/\sqrt{2}$)
 - ★ nécessité d'établir un seuil pour savoir si les valeurs appartiennent au contour ou non.
- ▶ Le contour détecté sur la diagonale a 3 pixels d'épaisseurs
 - ★ difficile de détecter des contours fins
 - ★ détecte mieux les contours diagonaux que horizontaux ou verticaux

Limitations de l'approche naïve

- Regardons maintenant le résultat sur une “vraie” image



- Le détecteur naïf est sensible au bruit de l'image et détecte trop de contours

Approche naïve

- ▶ Détecteurs de contours horizontaux et verticaux
- ▶ Limitations de l'approche naïve

Approche par convolution

- ▶ Rappels sur la convolution
- ▶ Liens avec la dérivée et lissage des hautes fréquences
- ▶ Filtres de Prewitt et Sobel

Détecteur optimal : filtre de Canny

- ▶ Qualité attendues d'un détecteur de contours
- ▶ Réduction du bruit avec un filtrage gaussien
- ▶ Calcul du Gradient
- ▶ Suppression des non maxima
- ▶ Seuillage des contours
- ▶ Traquage par hysteresis

Rappels sur la convolution

- On rappelle que la convolution d'une fonction continue f par une fonction continue g est la fonction :

$$(f \star g)(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) g(u - x, v - y) du dv$$

- Pour des fonctions discrètes on obtient :

$$(f \star g)(x, y) = \sum_{u=-\infty}^{+\infty} \sum_{v=-\infty}^{+\infty} f(x, y) g(u - x, v - y)$$

- En appliquant ceci à une image I de dimension finie et un “noyau de convolution” 3×3 K les pixels de la convolution de I par K sont donnés par :

$$(I \star K)(i, j) = \sum_{k=0}^2 \sum_{l=0}^2 K(k, l) I(i + k - 1, j + l - 1)$$

Rappels sur la convolution

- Par exemple, les contours horizontaux et verticaux de notre détecteur naïf sont formés par des convolées avec les noyaux :

$$K_v = \begin{pmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{et} \quad K_h = \begin{pmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

- Considérons l'image dont les pixels sont donnés par :

$$I = \begin{bmatrix} 180 & 18 & 129 & 132 \\ 132 & 19 & 140 & 134 \\ 243 & 243 & 245 & 253 \\ 138 & 135 & 136 & 136 \end{bmatrix} \quad (I \star K_v) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 113 & 121 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{aligned} (I \star K_v)_{1,1} &= 180 \times 0 + 18 \times 0 + 129 \times 0 \\ &\quad + 132 \times (-1) + 19 \times (1) + 140 \times 0 \\ &\quad + 243 \times 0 + 243 \times 0 + 245 \times 0 \\ &= -113 \rightarrow 113 \text{ on garde la valeur absolue} \end{aligned}$$

Rappels sur la convolution

- Les formules précédentes peuvent être étendues à des noyaux de plus grande dimension $(2p + 1) \times (2p + 1)$ auquel cas :

$$(I \star K)(i, j) = \sum_{k=0}^{2p} \sum_{l=0}^{2p} K(k, l) I(i + k - p, j + l - p)$$

- Plus la taille du noyau augmente, plus le coût du calcul de l'image convolée augmente
 - ▶ On se limite généralement à des noyaux de taille 3×3 ou 5×5

Approche naïve

- ▶ Détecteurs de contours horizontaux et verticaux
- ▶ Limitations de l'approche naïve

Approche par convolution

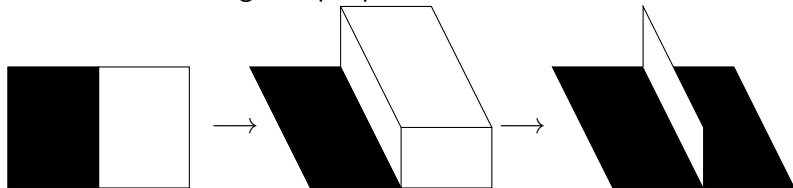
- ▶ Rappels sur la convolution
- ▶ Liens avec la dérivée et lissage des hautes fréquences
- ▶ Filtres de Prewitt et Sobel

Détecteur optimal : filtre de Canny

- ▶ Qualité attendues d'un détecteur de contours
- ▶ Réduction du bruit avec un filtrage gaussien
- ▶ Calcul du Gradient
- ▶ Suppression des non maxima
- ▶ Seuillage des contours
- ▶ Traquage par hysteresis

Liens avec la dérivée et lissage des hautes fréquences

- Si l'on redessine l'image simple précédente en 3D :



- ▶ Le lieu de contour correspond au saut de hauteur
 - ▶ Cela correspond au pic de la dérivée
- Pour une fonction f dérivable sur \mathbb{R} sa dérivée est donnée par :

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x-h)}{2h}$$

- ▶ Dans le cas d'une image discrète, h ne peut valoir moins que 1, on approxime donc sa dérivée selon x par :

$$\frac{\partial I}{\partial x}(x, y) = \frac{I(x+1, y) - I(x-1, y)}{2}$$

Liens avec la dérivée et lissage des hautes fréquences

- Notre détecteur de contour peut donc correspondre au gradient de notre image :

$$\nabla I = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right]$$

- ▶ Le gradient indique la direction de la plus forte pente d'une fonction de $\mathbb{R}^2 \rightarrow \mathbb{R}$

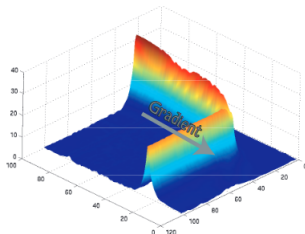
- Son module, aussi appelé la force du gradient, correspond à :

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

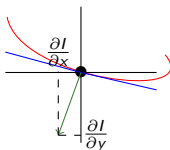
- ▶ En pratique il faut normaliser les valeurs des dérivées et de la norme.
- ▶ Les valeurs des dérivées sont entre $-255/2$ et $255/2$
- ▶ Les valeurs du module du gradient sont entre 0 et $255\sqrt{2}$

Liens avec la dérivée et lissage des hautes fréquences

- En représentant nos pixels en 3D, on peut relier les pixels de valeurs proches entre eux
 - ▶ on parle de courbe d'équipotence



- Le gradient d'un pixel indique les coordonnées du vecteur normale à la tangente de la courbe d'equipotence.



Liens avec la dérivée et lissage des hautes fréquences

- Malheureusement les images dérivées sont très bruitées...
 - ▶ En modélisant notre image comme la somme d'une image non bruitée et d'un bruit :

$$I(x) = I_{\text{pure}}(x) + \sum_i \varepsilon_i \sin(\omega_i x)$$

- ▶ Alors la dérivée aura pour expression :

$$I'(x) = I'_{\text{pure}}(x) + \sum_i \varepsilon_i \omega_i \cos(\omega_i x)$$

- ▶ Le bruit d'amplitude ε_i est amplifié par un facteur $\omega_i = 2\pi\nu_i$ (où ν_i est la fréquence).
- ▶ Les bruits de haute fréquence vont donc fortement perturber la dérivée (et le gradient)
- ▶ Il faut donc appliquer un filtre passe bas ou de lissage afin d'éliminer ces hautes fréquences.

Liens avec la dérivée et lissage des hautes fréquences

- Le bruit peut être lissé avec un filtre moyennneur

$$\frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} = \frac{1}{9} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$$

- Ou un filtre gaussien discrétisé $G_{\sigma}(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$

- Pour $\sigma = 0.8$

$$\begin{pmatrix} G(-1, -1) & G(-1, 0) & G(-1, 1) \\ G(0, -1) & G(0, 0) & G(0, 1) \\ G(1, -1) & G(1, 0) & G(1, 1) \end{pmatrix} \approx \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

- Pour $\sigma = 0.6$

$$\frac{1}{8} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

- Les coefficients des filtres sont généralement des puissances de 2 pour réduire le coût des multiplications/divisions à l'aide de shifts binaires.

Approche naïve

- ▶ Détecteurs de contours horizontaux et verticaux
- ▶ Limitations de l'approche naïve

Approche par convolution

- ▶ Rappels sur la convolution
- ▶ Liens avec la dérivée et lissage des hautes fréquences
- ▶ Filtres de Prewitt et Sobel

Détecteur optimal : filtre de Canny

- ▶ Qualité attendues d'un détecteur de contours
- ▶ Réduction du bruit avec un filtrage gaussien
- ▶ Calcul du Gradient
- ▶ Suppression des non maxima
- ▶ Seuillage des contours
- ▶ Traquage par hysteresis

Filtres de Prewitt et Sobel

- Les filtres de Prewitt/Sobel (1970) combinent un lissage unidimensionnel $[1, c, 1]$ et la dérivée selon une direction perpendiculaire au lissage (obtenue par le filtre $[-1, 0, 1]$).

- ▶ On commence par lisser selon l'axe des abscisses :

$$\begin{aligned}l_2(x, y-1) &= \frac{1}{2+c} (I(x-1, y-1) + cI(x, y-1) + I(x+1, y-1)) \\l_2(x, y+1) &= \frac{1}{2+c} (I(x-1, y+1) + cI(x, y+1) + I(x+1, y+1))\end{aligned}$$

- ▶ Puis on calcule la dérivée selon l'axe des ordonnées :

$$\begin{aligned}\frac{\partial l_2}{\partial y} &= l_2(x, y+1) - l_2(x, y-1) \\&= (I(x-1, y+1) + cI(x, y+1) + I(x+1, y+1))/(2+c) \\&\quad - ((I(x-1, y-1) + cI(x, y-1) + I(x+1, y-1))/(2+c))\end{aligned}$$

- ▶ Ce qui correspond au masque de convolution :

$$K_y = \frac{1}{2+c} \begin{pmatrix} -1 & 0 & 1 \\ -c & 0 & c \\ -1 & 0 & 1 \end{pmatrix} = \frac{1}{2+c} \begin{pmatrix} 1 \\ c \\ 1 \end{pmatrix} \otimes \begin{pmatrix} -1 & 0 & 1 \end{pmatrix}$$

Filtres de Prewitt et Sobel

- De même, en lissant selon l'axe des ordonnées et en dérivant selon l'axe des abscisses on obtient le masque

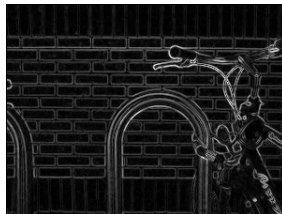
$$K_x = \frac{1}{2+c} \begin{pmatrix} -1 & -c & -1 \\ 0 & 0 & 0 \\ 1 & c & 1 \end{pmatrix} = \frac{1}{2+c} \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & c & 1 \end{pmatrix}$$

- Le filtre de Prewitt utilise un lissage uniforme $c = 1$
- Le filtre de Sobel utilise un lissage Gaussien $c = 2$

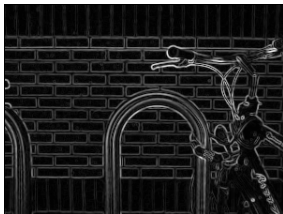
Filtres de Prewitt et Sobel



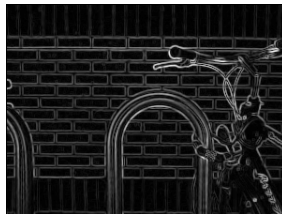
Image original



Filtre naïf



Filtre Sobel

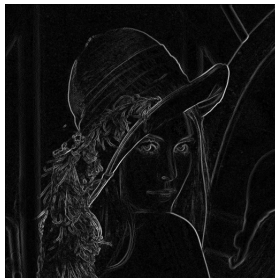


Filtre Prewitt

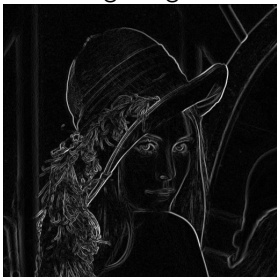
Filtres de Prewitt et Sobel



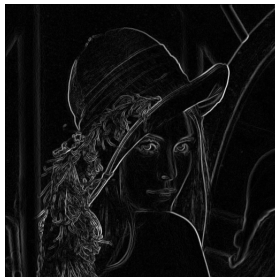
Image original



Filtre naïf



Filtre Sobel



Filtre Prewitt

Approche naïve

- ▶ Détecteurs de contours horizontaux et verticaux
- ▶ Limitations de l'approche naïve

Approche par convolution

- ▶ Rappels sur la convolution
- ▶ Liens avec la dérivée et lissage des hautes fréquences
- ▶ Filtres de Prewitt et Sobel

Détecteur optimal : filtre de Canny

- ▶ Qualité attendues d'un détecteur de contours
- ▶ Réduction du bruit avec un filtrage gaussien
- ▶ Calcul du Gradient
- ▶ Suppression des non maxima
- ▶ Seuillage des contours
- ▶ Traquage par hysteresis

Qualité attendues d'un détecteur de contours

- En 1986, John Canny a explicité trois critères que l'on peut attendre d'un bon détecteur de contours.
 1. Bonne détection → plus le filtre "lisse" le bruit plus la détection est bonne. Il faut donc maximiser le rapport signal sur bruit
 2. Bonne localisation : minimisation des distances entre les contours détectés et les contours réels
 3. clarté de la réponse : une seule réponse par contour et pas de faux positifs
- Trouver un filtrage optimal c'est trouver un compromis entre ces 3 critères.

Qualité attendues d'un détecteur de contours

- Le filtre de Canny est un des plus efficaces, facile à implémenter et donc un des plus populaires
- L'implémentation du détecteur de Canny peut être décomposée en 4 étapes :
 1. Appliquer un filtre Gaussien pour "lisser" l'image et en réduire le bruit
 2. Calculer le gradient de l'image lissée
 3. Suppression des non maxima pour affiner les contours
 4. Appliquer un double seuillage par hysteresis pour se débarrasser des "faux contours"

Approche naïve

- ▶ Détecteurs de contours horizontaux et verticaux
- ▶ Limitations de l'approche naïve

Approche par convolution

- ▶ Rappels sur la convolution
- ▶ Liens avec la dérivée et lissage des hautes fréquences
- ▶ Filtres de Prewitt et Sobel

Détecteur optimal : filtre de Canny

- ▶ Qualité attendues d'un détecteur de contours
- ▶ Réduction du bruit avec un filtrage gaussien
- ▶ Calcul du Gradient
- ▶ Suppression des non maxima
- ▶ Seuillage des contours
- ▶ Traquage par hysteresis

Réduction du bruit avec un filtrage gaussien

- On rappelle que le noyau de convolution d'un lissage gaussien est donné par la matrice $G = (g_{i,j})_{0 \leq i,j \leq 2p}$ avec $g_{i,j} = G_{\sigma}^{(2)}(i - p, j - p)$.

$$G_{\sigma}^{(2)}(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} = \left(\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \right) \cdot \left(\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{y^2}{2\sigma^2}} \right)$$

- On remarque que $G_{\sigma}^{(2)}(x, y) = G_{\sigma}^{(1)}(x) \cdot G_{\sigma}^{(1)}(y)$ où

$$G_{\sigma}^{(1)}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$$

- ▶ On peut “séparer” un filtre gaussien 2D en un produit de deux filtres gaussien 1D.
- ▶ On dit que le filtre gaussien est **séparable**
- ▶ Cela permet entre autre de réduire les coûts de calculs.

Réduction du bruit avec un filtrage gaussien

- La réduction du bruit se fait généralement par un filtrage Gaussien, souvent de taille 5×5 , avec, par exemple $\sigma = 1.4$

$$\frac{1}{128} \begin{pmatrix} 1 & 3 & 4 & 3 & 1 \\ 3 & 6 & 8 & 6 & 3 \\ 4 & 8 & 10 & 8 & 4 \\ 3 & 6 & 8 & 6 & 3 \\ 1 & 3 & 4 & 3 & 1 \end{pmatrix}$$

- Augmenter la taille du noyau réduit la sensibilité du filtre au bruit
 - ▶ le noyau étant plus grand, on lisse avec plus de valeurs
- La localisation des erreurs diminue lorsque la taille augmente
 - ▶ on prend en compte la valeur de pixels de plus en plus éloignées du pixel de départ
- Un noyau 5×5 est un bon compromis en pratique.

Approche naïve

- ▶ Détecteurs de contours horizontaux et verticaux
- ▶ Limitations de l'approche naïve

Approche par convolution

- ▶ Rappels sur la convolution
- ▶ Liens avec la dérivée et lissage des hautes fréquences
- ▶ Filtres de Prewitt et Sobel

Détecteur optimal : filtre de Canny

- ▶ Qualité attendues d'un détecteur de contours
- ▶ Réduction du bruit avec un filtrage gaussien
- ▶ Calcul du Gradient
- ▶ Suppression des non maxima
- ▶ Seuillage des contours
- ▶ Traquage par hysteresis

Calcul du Gradient

- Le gradient est calculé de la même façon que précédemment

$$\nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right)$$

- De là on peut calculer l'intensité du gradient

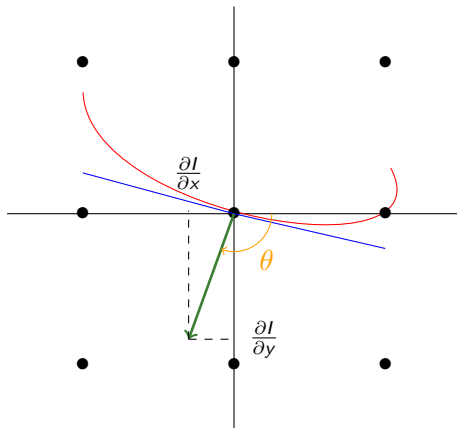
$$\|\nabla I\|_2 = \sqrt{\frac{\partial I}{\partial x}^2 + \frac{\partial I}{\partial y}^2}$$

- Mais également l'angle de la normale avec l'axe horizontal

$$\Theta = \text{atan2} \left(\frac{\partial I}{\partial y}, \frac{\partial I}{\partial x} \right)$$

- La fonction arctan détermine l'angle à π près, pour obtenir l'angle à 2π près il faut regarder le signe des dérivées partielles.
- En pratique on peut utiliser la fonction `atan2` de la bibliothèque `math.h`

Calcul du Gradient



Approche naïve

- ▶ Détecteurs de contours horizontaux et verticaux
- ▶ Limitations de l'approche naïve

Approche par convolution

- ▶ Rappels sur la convolution
- ▶ Liens avec la dérivée et lissage des hautes fréquences
- ▶ Filtres de Prewitt et Sobel

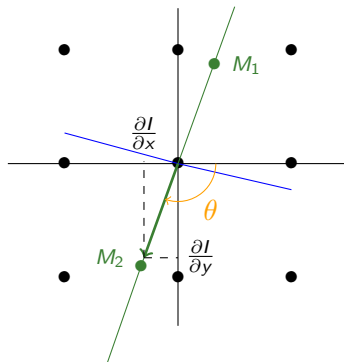
Détecteur optimal : filtre de Canny

- ▶ Qualité attendues d'un détecteur de contours
- ▶ Réduction du bruit avec un filtrage gaussien
- ▶ Calcul du Gradient
- ▶ **Suppression des non maxima**
- ▶ Seuillage des contours
- ▶ Traquage par hysteresis

Suppression des non maxima

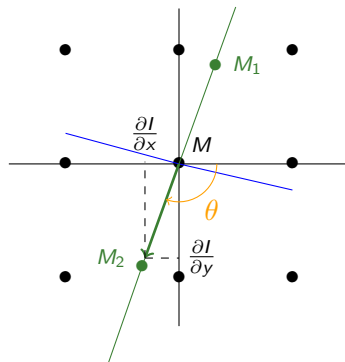
- Nous avons déjà remarqué que les bords obtenus par un détecteur ont une largeur de plusieurs pixels.
- L'algorithme de suppression des non maxima consiste à éliminer les pixels sur lesquels le gradient n'est pas maximal afin d'affiner les contours trouvés.
- On commence par placer deux points M_1 , M_2 sur la normale, de part et d'autre du point (i, j) , à une distance unité (distance de 1).
- La valeur de la norme du gradient aux points M_1 et M_2 est calculée par interpolation à partir des pixels voisins.
 - ▶ généralement on effectue une interpolation à l'ordre 0 ou 1.

Suppression des non maxima



- Le pixel (i, j) est retenu comme appartenant au contour si la norme de son gradient est supérieure à celle des points M_1 et M_2 .
 - ▶ Le point M_1 a pour coordonnées $(i - \cos(\theta), j - \sin(\theta))$
 - ▶ Le point M_2 a pour coordonnées $(i + \cos(\theta), j + \sin(\theta))$
- Dans le cas contraire, le pixel est éliminé et on met sa valeur à 0.
 - ▶ on élimine les pixels qui ne sont pas sur un maximum du gradient.

Suppression des non maxima



- Le gradient donne la direction du pixel ayant la plus forte intensité.
 - cela ne veut pas dire que M_2 a une plus grande valeur que M ni que M_1 a une plus faible valeur que M
 - il faut donc vérifier les valeurs de M_1 **ET** de M_2

Approche naïve

- ▶ Détecteurs de contours horizontaux et verticaux
- ▶ Limitations de l'approche naïve

Approche par convolution

- ▶ Rappels sur la convolution
- ▶ Liens avec la dérivée et lissage des hautes fréquences
- ▶ Filtres de Prewitt et Sobel

Détecteur optimal : filtre de Canny

- ▶ Qualité attendues d'un détecteur de contours
- ▶ Réduction du bruit avec un filtrage gaussien
- ▶ Calcul du Gradient
- ▶ Suppression des non maxima
- ▶ **Seuillage des contours**
- ▶ Traquage par hysteresis

Seuillage des contours

- La dernière étape consiste à binariser l'image avec un seuil
 - ▶ Si la valeur de la norme du gradient est inférieure à un certain seuil alors le point n'appartient pas au contour → on met sa valeur à 0.
 - ▶ Si la valeur de la norme du gradient est supérieure à un certain seuil alors le point appartient contour → on met sa valeur à 255.
- Il est difficile de déterminer la valeur d'un seuil unique de façon efficace.
- Une approche plus raisonnable consiste à utiliser un double seuillage avec un seuil haut et un seuil bas
 - ▶ Les valeurs au dessus du seuil haut sont les contours "certains" de l'image
 - ▶ Les valeurs en dessous du seuil bas ne sont pas des contours
 - ▶ Les valeurs entre les deux sont des contours "potentiels"

Approche naïve

- ▶ Détecteurs de contours horizontaux et verticaux
- ▶ Limitations de l'approche naïve

Approche par convolution

- ▶ Rappels sur la convolution
- ▶ Liens avec la dérivée et lissage des hautes fréquences
- ▶ Filtres de Prewitt et Sobel

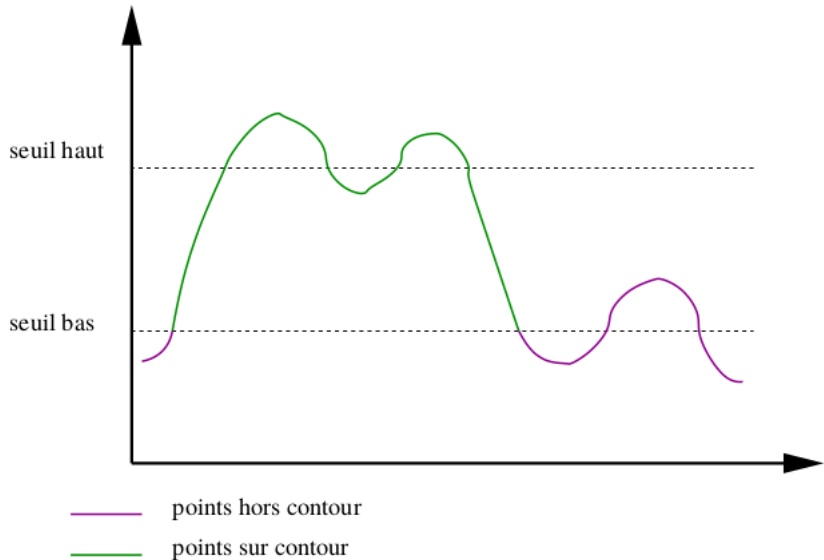
Détecteur optimal : filtre de Canny

- ▶ Qualité attendues d'un détecteur de contours
- ▶ Réduction du bruit avec un filtrage gaussien
- ▶ Calcul du Gradient
- ▶ Suppression des non maxima
- ▶ Seuillage des contours
- ▶ Traquage par hysteresis

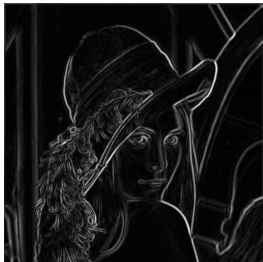
Traquage par hysteresis

- Pour déterminer si un contour potentiel est un contour certain ou non on regarde les pixels voisins
 - ▶ Si un voisin est un contour certain, alors le pixel devient un “contour certain” et sa valeur est passée à 255
 - ▶ Dans le cas contraire il est éliminé et sa valeur est passée à 0
- En pratique on construit une liste de tous les contours certains
 - ▶ si un pixel voisin d'un contour certain est un contour potentiel alors le pixel devient également un contour certain et on le rajoute à la liste.
- Il existe plusieurs approches pour regarder parmi les pixels voisins
 - ▶ On regarde parmi les 8 pixels voisins
 - ▶ On regarde seulement parmi les pixels les plus proches de la ligne de gradient

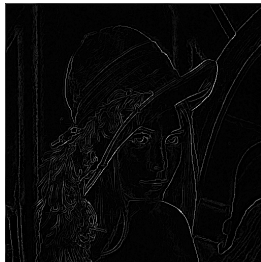
Traquage par hysteresis



Les étapes du filtre de Canny



Passage du filtre de Sobel



Suppression des non-maximaux



Double seuillage (35%, 20%)



Traquage par hysteresis