Détection de contours dans les images

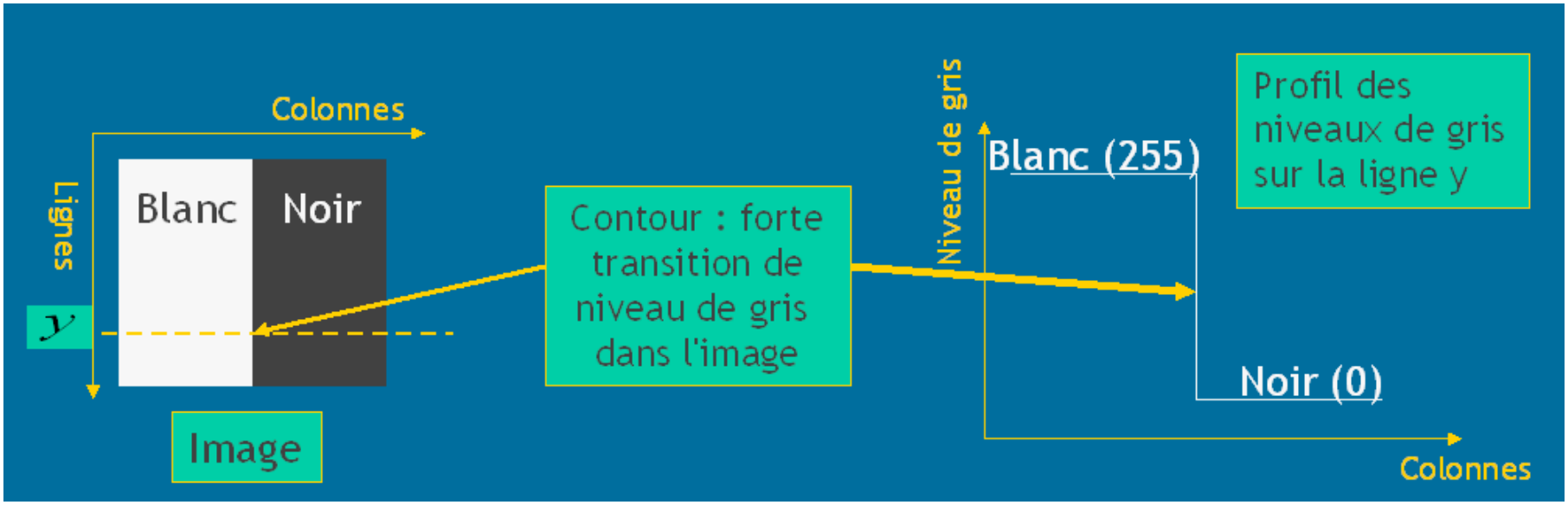
Séverine Dubuisson

Caractérisation d'un contour

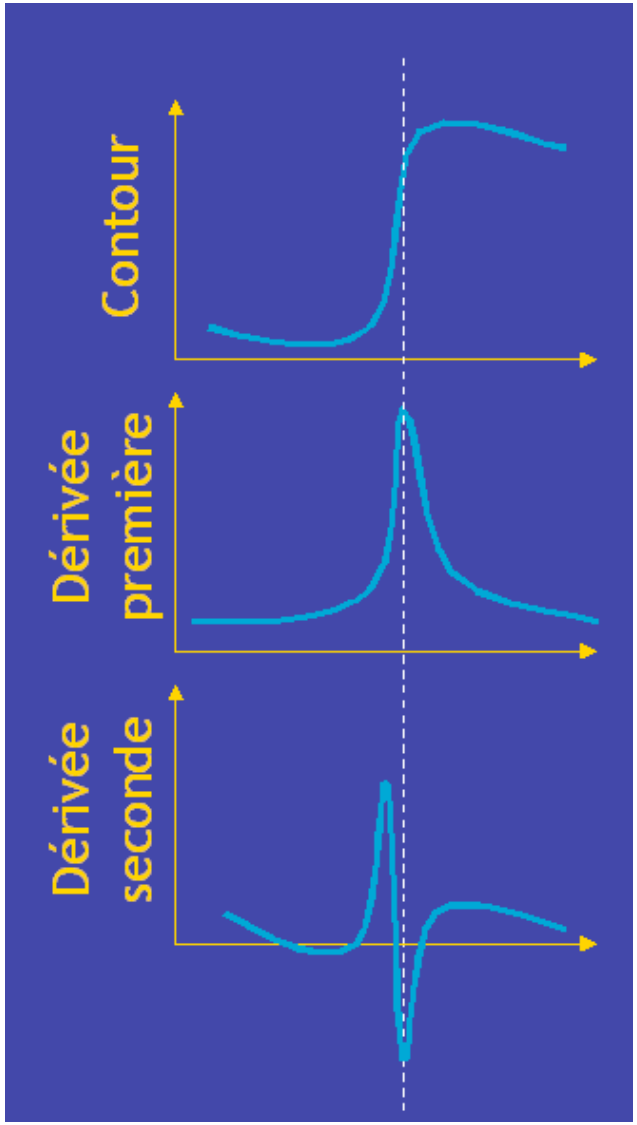
- Discontinuité de la fonction d'intensité des images
  - Discontinuité de la fonction de réflectance (texture, ombre, ...)
  - Discontinuité de profondeur (bords des objets)

Définitions d'un contour

- Définition spatiale :
  - Lieu de variations significatives de l'information de niveaux de gris (il existe peu de travaux sur les contours dans les images couleurs ou multi-spectrales) : rupture d'intensité
- Définition fréquentielle : haute fréquence du spectre d'amplitude de la transformée de Fourier de l'image



Détection d'un contour



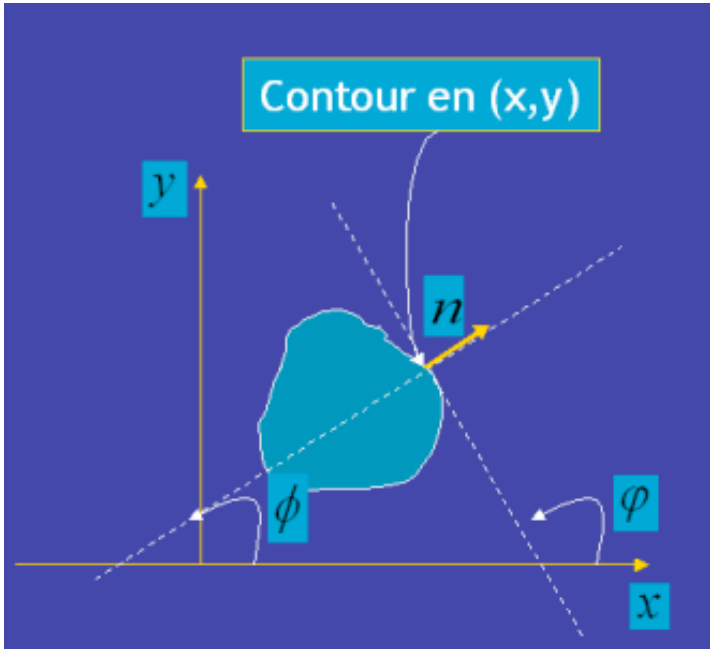
Les méthodes dérivatives

- Dérivée première d'un contour : gradient
  - Opérateurs dérivatifs du premier ordre
  - Recherche d'un passage par un maximum, ou extrama local
- Dérivée seconde d'un contour : laplacien
  - Opérateurs dérivatifs du second ordre
  - Recherche d'un passage par zéro

Travail sur la dérivée première : principe (1/

- Un contour d'orientation φ au point de coordonnées (x, y) est détecté par un maximum de la dérivée première, dans la direction ϕ du gradient

$$g(\phi) = \vec{\nabla} f(x, y)\vec{n} \qquad \vec{n} = \begin{pmatrix} \cos\phi \\ \sin\phi \end{pmatrix}$$



Travail sur la dérivée première : principe (2/

- On a donc $g(\phi) = \frac{\partial f}{\partial x}\cos\phi + \frac{\partial f}{\partial y}\sin\phi$
- En dérivant par rapport à ϕ et en annulant on obtient

$$\phi = \arctan\left(\begin{array}{c}\frac{\partial f}{\partial y}\\ \frac{\partial f}{\partial x}\end{array}\right) \qquad \varphi = \phi + \frac{\pi}{2}$$

- La norme du gradient est donnée par :

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

Gradient d'une image : interprétations

- Gradient : vecteur des dérivées horizontales et verticales de l'intensité
  - La dérivée directionnelle est maximisée dans la direction du gradient
  - La dérivée de $f(x, y)$ dans une direction $\vec{d}$ donnée s'écrit $\vec{\nabla} f(x, y) \cdot \vec{d}$
- L'amplitude (i.e module) du gradient est liée à la quantité de variation locale de l'intensité : plus le module est grand, plus le contour est fort
- Argument du gradient : direction de la plus grand pente

Cas discret (1/3)

- On estime le gradient en calculant une variation monodimensionnelle dans une direction ϕ donnée

$$\begin{aligned} G_\phi(x,y) &= (I \star W_\phi)(x,y) \\ &= \sum_{i=-m}^{m} \sum_{j=-n}^{n} I(x+i, y+j) \cdot W_\phi(i,j) \end{aligned}$$

  La taille de l'opérateur (autour de la position (x,y)) est donnée par le couple (m,n)

- Pour estimer le gradient, on choisit deux directions privilégiées orthogonales (les lignes et les colonnes) sur lesquelles on le projette
  - G_x est le gradient vertical
  - G_y est le gradient horizontal

Cas discret (2/3)

- La norme du gradient peut être calculée de trois manières :

$$\begin{aligned} G(x,y) &= \vec{\nabla} f(x,y) \\ &= \sqrt{(G_x(x,y))^2 + (G_y(x,y))^2} \\ G(x,y) &= \max\left(|G_x(x,y)|, |G_y(x,y)|\right) \\ G(x,y) &= |G_x(x,y)| + |G_y(x,y)| \end{aligned}$$

- La direction du gradient en un point est donnée par :

$$\phi(x,y) = \arctan\left(\frac{G_y(x,y)}{G_x(x,y)}\right)$$

Cas discret (3/3)

- On approche les dérivées directionnelles en utilisant les opérateurs anisotropes :

$$\begin{aligned} G_x(i,j) &\approx \triangle_x I(i,j) = I(i+1,j) - I(i,j) \\ G_y(i,j) &\approx \triangle_y I(i,j) = I(i,j+1) - I(i,j) \end{aligned}$$

- La norme du gradient peut être obtenue de trois manières :

$$\begin{aligned} G(i,j) &= \sqrt{(I(i+1,j) - I(i,j))^2 + (I(i,j+1) - I(i,j))^2} \\ G(i,j) &= \max\left(|I(i+1,j) - I(i,j)|, |I(i,j+1) - I(i,j)|\right) \\ G(i,j) &= |I(i+1,j) - I(i,j)| + |I(i,j+1) - I(i,j)| \end{aligned}$$

Filtres dérivatifs du premier ordre

- Un filtre est composé de deux masques :
  - H_x détecte les contours verticaux
  - H_y détecte les contours horizontaux
- Dans le domaine spatial : convolution linéaire de l'image avec les filtres
  - Image des contours verticaux, ou gradient vertical : $G_x = I \star H_x$
  - Image des contours horizontaux, ou gradient horizontal : $G_y = I \star H_y$
  - Module du gradient : $G = \sqrt{G_x^2 + G_y^2}$ ou $G = \max(|G_x|, |G_y|)$
- Filtres courants : Roberts, Prewitt, Sobel, Kirch

Opérateurs du premier ordre

- Opérateur de Roberts : approche les dérivées directionnelles suivant les axes orientés à 45° (contours obliques)
  - Filtres : $H_x = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \quad H_y = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$
  - Direction φ du contour : $\varphi = \frac{\pi}{4} + \phi$
- Opérateurs de Sobel ($c = 2$) et Prewitt ($c = 1$)
  - Filtres : $H_x = \begin{pmatrix} 1 & 0 & -1 \\ c & 0 & -c \\ 1 & 0 & -1 \end{pmatrix} \quad H_y = \begin{pmatrix} -1 & -c & -1 \\ 0 & 0 & 0 \\ 1 & c & 1 \end{pmatrix}$
  - Direction φ du contour : $\varphi = \frac{\pi}{2} + \phi$

Compléments sur Prewitt et Sobel

- Ces masques opèrent tout d'abord un lissage de l'image, suivi d'une opération de dérivation : ils sont dits séparables

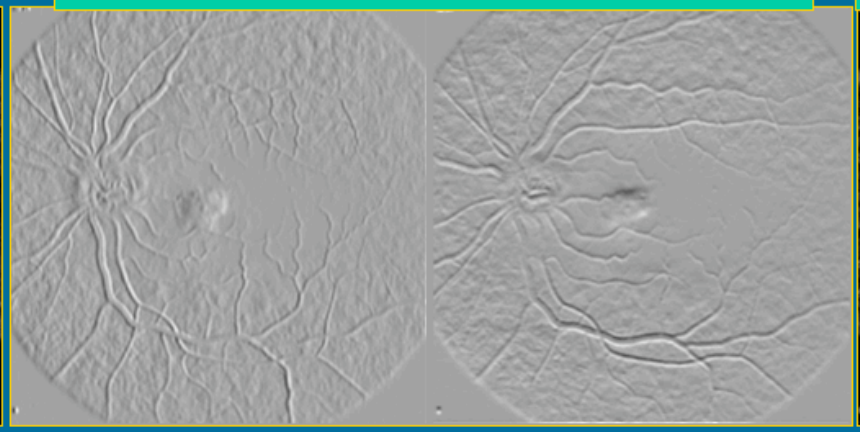
$$H_x = \begin{pmatrix} 1 \\ \mathsf{c} \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & -1 \end{pmatrix} \qquad H_y = \begin{pmatrix} 1 \\ \mathsf{0} \\ -1 \end{pmatrix} \cdot \begin{pmatrix} 1 & \mathsf{c} & 1 \end{pmatrix}$$

- Le lissage permet d'atténuer le bruit dans l'image, pouvant être interprété comme un contour :
  - Prewitt : filtre moyenneur
  - Sobel : filtre gaussien

Exemple de filtrage : Sobel

Image originale

Frontières horizontales et verticales

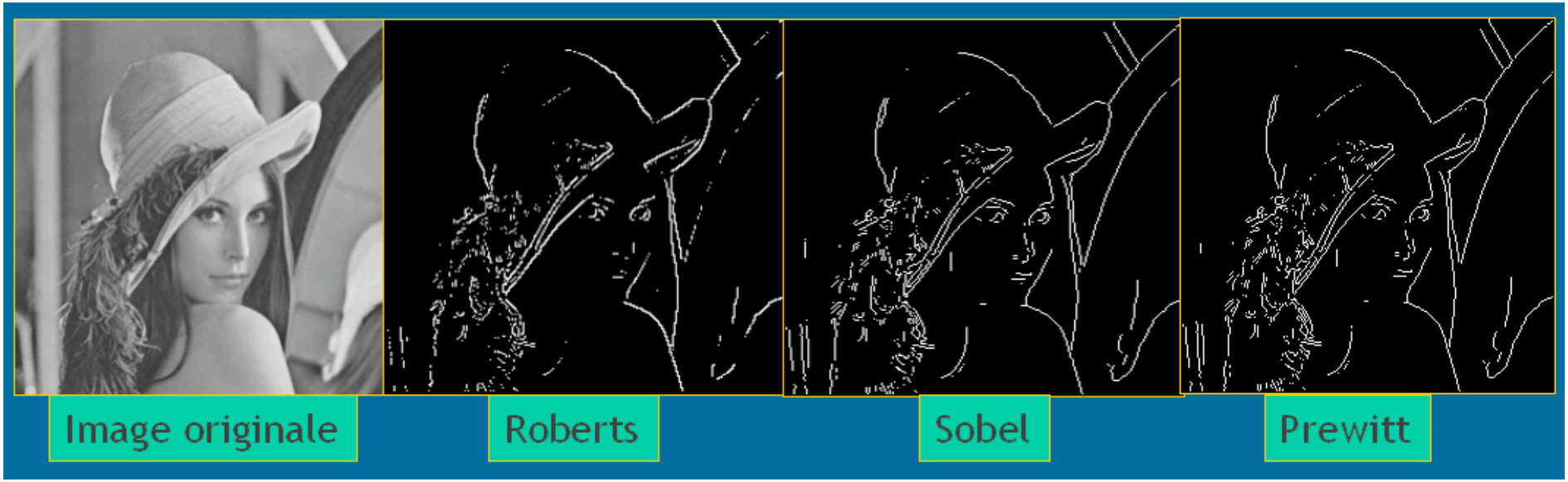


Module du gradient

L'image des dérivées directionnelles est grise car l'information d'intensité a été éliminée lors du filtrage (la fréquence continue est éliminée)

L'image du module de gradient met en évidence les frontières (blanc)

Résultats comparatifs



Opérateur de Kirch (1/2)

- Opérateur à 8 masques, chacun correspondant à une direction préférentielle obtenue par rotations de $\frac{\pi}{8}$ successives

$$H_0 = \begin{pmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & 3 \end{pmatrix} \quad H_1 = \begin{pmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{pmatrix}$$

$$H_2 = \begin{pmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{pmatrix} \quad H_3 = \begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{pmatrix}$$

$$H_4 = \begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{pmatrix} \quad H_5 = \begin{pmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{pmatrix}$$

Opérateur de Kirch (2/2)

$$H_6 = \begin{pmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{pmatrix} \quad H_7 = \begin{pmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix}$$

- Le gradient retenu est celui correspondant à la valeur maximum, donnée par $\max_{i=0,\cdots,7}\{|I \star H_i|\}$
- L'orientation retenue est celle du masque ayant maximisé le gradient
- Cet opérateur est aussi appelé gradient boussole ou gradient compas

Opérateur de MDIF

- Combinaison d'un filtre moyenneur et de l'opérateur de Prewitt

$$H_x = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix} \star \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & -1 & 0 \\ 1 & 2 & 0 & -2 & -1 \\ 1 & 3 & 0 & -3 & -1 \\ 1 & 2 & 0 & -2 & -1 \\ 0 & 1 & 0 & -1 & 0 \end{pmatrix}$$

$$H_y = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix} \star \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 2 & 3 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & -2 & -3 & -2 & -1 \\ 0 & -1 & -1 & -1 & 0 \end{pmatrix}$$

Taille des filtres

- Les masques les plus couramment utilisés sont de taille 3×3 $(m = n = 1)$
- On peut généraliser : cas du Sobel (Asfar) pour une taille de voisinage quelconque

$$\begin{aligned} W_x(i,j) &= i/(|i| \cdot (|i| + |j|)) \\ W_y(i,j) &= j/(|j| \cdot (|i| + |j|)) \end{aligned}$$

- Quelle taille choisir pour un opérateur ?
  - Plus le masque est grand moins le gradient est sensible au bruit (bruit atténué par le lissage)
  - Plus le masque est grand, moins bonne est la localisation des contours (le lissage rend l'image floue)

Du gradient au contour

- Pour détecter les contours d'une image, il faut procéder de la sorte :
  1. Mesure en chaque point de l'image de l'amplitude et de la direction du gradient
  2. Détermination des seuils d'amplitude
  3. Suppression directionnelle des non-maxima locaux
  4. Extraction des éléments essentiels
  5. Reconstruction des contours par hysteresis

Détermination des seuils d'amplitude

- Technique de Voorhees et Poggio
- Seuillage basé sur la validation de deux seuils, un seuil bas S_b et un seuil haut S_h, avec $S_b < S_h$, de la manière suivante :
  - si $G(x, y) < S_b$ alors on est sur que (x, y) n'est pas un point de contour
  - si $G(x, y) > S_h$ alors on est sur que (x, y) est un point de contour
  - si $Sb < G(x, y) < S_h$ alors l'appartenance à un contour dépend du contexte

Suppression directionnelle des non maxima locaux (1/2)

- Extraction des maxima locaux dans la direction du gradient
  - Opération sur un voisinage de l'image de la norme du gradient
  - On compare le gradient d'un point avec celui de ses deux voisins M_1 et M_2 dans la direction perpendiculaire au contour
- Un point M est un maximum local si :

$$\|\vec{\nabla}_M\| > \|\vec{\nabla}_{M_1}\| \quad \text{et} \quad \|\vec{\nabla}_M\| \geq \|\vec{\nabla}_{M_2}\|$$

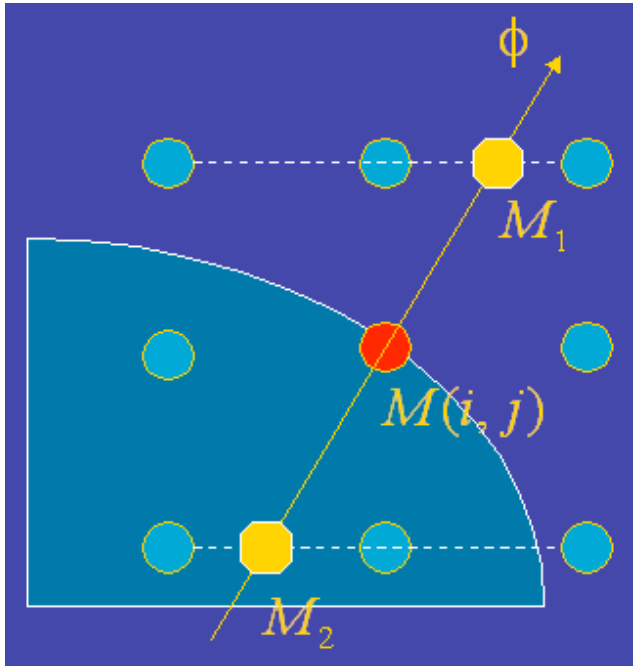
Suppression directionnelle des non maxima locaux (2/2)

- Les modules de gradient en M_1 et M_2 doivent être interpolés

$$\|\vec{\nabla}_{M_1}\| = \frac{u_j}{u_i}\|\vec{\nabla}(i, j-1)\| + \frac{u_i - u_j}{u_i}\|\vec{\nabla}(i+1, j)\|$$

$$\|\vec{\nabla}_{M_2}\| = \frac{u_j}{u_i}\|\vec{\nabla}(i, j+1)\| + \frac{u_i - u_j}{u_i}\|\vec{\nabla}(i-1, j)\|$$

avec $u_i = G_x(i, j)$ et $u_j = G_y(i, j)$



Extraction des éléments essentiels

- Points pour lesquels on est sûr de l'appartenance à un contour
  - $(x, y) : G(x, y) > S_h$
  - Points de départ pour l'étape de reconstitution des contours

Reconstitutions des points par hystérésis

- On part des points $M(x, y)$ déterminés éléments essentiels
- On utilise deux seuils S_b et S_h, avec $S_b < S_h$, et pour chaque point M, on peut procéder des deux manières suivantes :
- Procédure 1 : on ne valide que les ensembles connexes de points présentant au moins un point dont la norme de gradient est supérieure à S_h
- Procédure 2 : on valide tous les pixels connexes aux points détectés et possédant une norme de gradient supérieure à S_b
- On répète la procédure pour tous les nouveaux points validés

Travail sur la dérivée seconde : principe (1/2

- Un changement linéaire d'amplitude d'une fonction f correspond au passage par zéro de sa dérivée seconde
- Dans le cas d'une image, il n'existe pas une dérivée seconde unique, mais 4 dérivées partielles (selon x^2, y^2, xy et yx)
- En pratique, on a recours au laplacien faisant la somme des deux dérivées partielles principales

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

- Un passage par zéro est caractérisé par une pente positive dans le cas d'une frontière ayant une amplitude augmentant de gauche à droite ou de bas en haut dans l'image

Travail sur la dérivée seconde : principe (2/2

- Pour limiter les réponses dues au bruit, on filtre l'image par g

$$\nabla^2(I \star g) = (\nabla^2 I) \star g = I \star (\nabla^2 g)$$

- En général on utilise un filtre gaussien g

$$g(r) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{r^2}{2\sigma^2}} \quad \text{où} \quad r = \sqrt{x^2 + y^2}$$

dont le laplacien est :

$$\nabla^2 g(r) = -\frac{1}{\pi\sigma^4}\left(1 - \frac{r^2}{2\sigma^2}\right) e^{-\frac{r^2}{2\sigma^2}}$$

Laplacien classique : cas discret

- On approche le laplacien par les différences finies

$$\begin{aligned} L(i,j) &= -\overbrace{[I(i,j-1) - 2I(i,j) + I(i,j+1)]}^{\frac{\partial^2 f}{\partial x^2}} \\ &- \underbrace{[I(i-1,j) - 2I(i,j) + I(i+1,j)]}_{\frac{\partial^2 f}{\partial y^2}} \end{aligned}$$

- Dans le domaine spatial, on calcule le laplacien par convolution entre l'image et un masque, dont les deux plus courants sont :

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

Avantages/Inconvénients du laplacien

- Avantages :
  - Un seul paramètre est nécessaire (variance de la gaussienne)
  - Pas de seuil significatif : les points remarquables sont détectés par un critère de changement de signe au niveau d'un passage par zéro
  - Les contours sont fermés
- Inconvénients :
  - Plus grande sensibilité au bruit
  - Aucune information sur l'orientation du contour

Approche DOG

- DOG : *Difference of Gaussians*
- Lissage à l'aide d'un filtre gaussien $g(x, y)$ puis calcul de la dérivée seconde dans la direction du gradient

$$I_z = \triangle (I \star g(x, y)) = I \star \triangle g(x, y)$$

- La dérivée seconde d'une gaussienne s'approche par la différence de deux gaussiennes
- Recherche des passages par zéro

Filtre optimum

- Principe du filtrage : trouver la réponse impulsionnelle d'un filtre susceptible d'extraire les points de contour tout en réduisant le bruit
- Introduire un filtre à réponse impulsionnelle séparable $h(x, y) = h_1(x)h_2(y)$
  - Réduction du temps de calcul
  - Prise en compte de caractéristiques différentes suivant les directions
  - Généralisation à des dimensions quelconques

Principe du filtrage

- Soit un filtre séparable

$$I(x,y)\star h(x,y) = I(x,y)\star(h_1(x)h_2(y)) = (I(x,y)\star h_1(x))\star h_2(y)$$

- Calcul de la dérivée première :

$$\frac{\partial(I(x,y)\star h(x,y))}{\partial x} = I(x,y)\star\left(\frac{\partial h_1(x)}{\partial x}h_2(y)\right)$$

- Calcul de la dérivée seconde (laplacien) :

$$\triangle(I(x,y)\star h(x,y)) = I(x,y)\star\left(\frac{\partial^2 h_1(x)}{\partial x^2}h_2(y) + h_1(x)\frac{\partial^2 h_2(y)}{\partial y^2}\right)$$

Modèles du signal et du bruit (1/3)

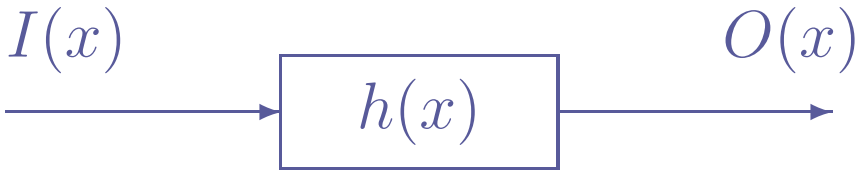
- Un contour est un seuil d'amplitude A auquel on ajoute un bruit $n(x)$ gaussien blanc de variance σ_0^2 (modèle additif) :

$$I(x) = AY(x) + n(x) \quad \text{avec} \quad Y(x) = U(x) = \begin{cases} 0 & \text{si} \quad x < 0 \\ 1 & \text{si} \quad x \geq 0 \end{cases}$$

- Le bruit est un bruit blanc additif

$$\begin{aligned} E(n(x)) &= 0 \\ E(n^2(x)) &= \sigma_0^2 \quad \forall x \end{aligned}$$

Modèles du signal et du bruit (2/3)



$$O(x) = \int_{-\infty}^{+\infty} I(x-u)h(u)du$$

$$O(x) = \int_{-\infty}^{+\infty} AY(x-u)h(u)du + \int_{-\infty}^{+\infty} n(x-u)h(u)du$$

- Supposons le contour placé en $x = 0$, la sortie du filtre s'écrit :

$$O(x) = A\int_{-\infty}^{x} h(u)du + \int_{-\infty}^{+\infty} n(x-u)h(u)du$$

$$O(0) = A\int_{-\infty}^{0} h(u)du + \int_{-\infty}^{+\infty} n(-u)h(u)du$$

Modèles du signal et du bruit (3/3)

- Rechercher la réponse impulsionnelle d'un filtre séparable susceptible de diminuer le bruit et appliquer les opérateurs de dérivée
  1. Modéliser le signal à extraire et le bruit de l'image
  2. Transformer le problème d'extraction des points de contour en problème d'optimisation
  3. Résoudre et trouver une solution approchée
     $\rightsquigarrow$ on souhaite que la sortie présente un maximum lorqu'il y a une arête : un contour optimal est situé en $x = 0$

Modèle analytique de Canny

- Canny propose en 1986 un étude sur la détection de contours
  - Etude dans le cas 1D de signal bruité
  - A formalisé trois critères que doivent valider un détecteur de contours :
    - Détection : robustesse au bruit
    - Localisation : précision de la localisation du point de contour
    - Unicité : une seule réponse par contour
- A chaque critère est associée une formule mathématique
- La maximisation de ces critères conduit à la résolution d'une équation différentielle dont la solution est le filtre f permettant de détecter le contour

Critère de détection

- But : maximiser le rapport signal sur bruit, ou *SNR* (*Signal Noise Ratio*)

$$E(N^2) = E\left[\int_{-\infty}^{+\infty} n(-u)h(u)du \int_{-\infty}^{+\infty} n(-v)h(v)dv\right]$$

$$E(N^2) = \sigma_0^2 \int_{-\infty}^{+\infty} h^2(u)du$$

- On cherche le maximum du critère :

$$C_1 = \text{SNR} = \frac{A}{\sigma_0} \frac{|\int_{-\infty}^{0} h(u)du|}{\sqrt{\int_{-\infty}^{\infty} h^2(u)du}} = \frac{A}{\sigma_0}\Sigma(h)$$

Critère de localisation (1/3)

- But : rechercher l'écart entre le maximum de la sortie en x_0 et 0 : $O'(x_0) = 0$

$$O'(x) = \underbrace{\int_{-\infty}^{+\infty} AY(x-u)h'(u)du}_{\text{Contribution } S(x) \text{ du signal}} + \underbrace{\int_{-\infty}^{+\infty} n(x-u)h'(u)du}_{\text{Contribution } N(x) \text{ du bruit}}$$

$$S(x) = Ah(x)$$

$$N(x) = \int_{-\infty}^{+\infty} n(x-u)h'(u)du$$

Critère de localisation (2/3)

$$E(N^2(x)) = \sigma_0^2 \int_{-\infty}^{+\infty} h'^2(u)du$$

- Supposons x_0 proche de 0, on a :

$$S(x_0) \approx Ah(0) + x_0 Ah'(0) + o(x_0^2)$$

- h est impaire :

$$S(x_0) \approx x_0 Ah'(0)$$

Critère de localisation (3/3)

$$O'(x_0) = S(x_0) + N(x_0) \approx Ax_0h'(0) + N(x_0)$$

- D'où :

$$x_0 = -\frac{N(x_0)}{Ah'(0)} \quad \text{et} \quad E(x_0^2) = \frac{\sigma_0^2}{A^2}\frac{\int_{-\infty}^{+\infty} h'^2(u)du}{h'^2(0)}$$

- On obtient le critère de localisation :

$$\boxed{C_2 = \frac{1}{\sqrt{E(x_0^2)}} = \frac{A}{\sigma_0}\frac{|h'(0)|}{\sqrt{\int_{-\infty}^{+\infty} h'^2(u)du}} = \frac{A}{\sigma_0}\Lambda(h)}$$

Critère d'unicité de la réponse (1/3)

- But : réduire les réponses multiples
- Le détecteur de contour ne doit pas identifier de mutiples pixels de contours quand un seul existe
- On veut que la distance entre les pics de la réponse au bruit soit approximativement la largeur de bande de la réponse du filtre à un seul contour.

Critère d'unicité de la réponse (2/3)

- Rice a montré que la distance moyenne $\overline{d}$ entre deux passages par zéro d'un processus gaussien, obtenu par filtrage d'un bruit blanc $g(x)$, est donnée par :

$$\overline{d} = \pi \left(\sqrt{\frac{-R(0)}{R''(0)}} \right)$$

où $R(\tau) = \int_{-\infty}^{+\infty} g(x+\tau)g(x)dx$ est la fonction d'auto-corrélation de g

- On cherche l'intervalle moyen entre deux passages par zéro de h'

Critère d'unicité de la réponse (3/3)

- On a $R''(\tau) = \int_{-\infty}^{+\infty} g''(x+\tau)g(x)dx$
- Soit $R''(0) = -\int_{-\infty}^{+\infty} g'^2(x)dx$
- Et ainsi $\overline{d} = \pi \left[\frac{\int_{-\infty}^{+\infty} g^2(x)dx}{\int_{-\infty}^{+\infty} g'^2(x)dx} \right]^{\frac{1}{2}}$
- Dans notre problème, $g = h'$, et on obtient ainsi le critère d'unicité :

$$\boxed{C_3 = 2\overline{d} = 2\pi \left[\frac{\int_{-\infty}^{+\infty} h'^2(x)dx}{\int_{-\infty}^{+\infty} h''^2(x)dx} \right]^{\frac{1}{2}}}$$

Résolution : approche de Canny (1/3)

- Canny a proposé de majorer $\Sigma\Lambda$ avec $\overline{d} = kW$
- h est impaire, de support fini $[-W, W]$
- On cherche à minimiser $\int_{-W}^{0} h^2(y)dy$
- Avec les trois contraintes :

$$\begin{aligned}
\int_{-W}^{0} h(y)dy &= D_1 \\
\int_{-W}^{0} h'^2(y)dy &= D_2 \\
\int_{-W}^{0} h''^2(y)dy &= D_3
\end{aligned}$$

Résolution : approche de Canny (2/3)

- D'où la fonction critère :

$$\phi = \int_{-W}^{0} \left(h^2(y) + \lambda_1 h(y) + \lambda_2 h'^2(y) + \lambda_3 h''^2(y) \right) dy$$

- Ce qui conduit à l'équation :

$$\boxed{C = 2h + \lambda_1 - 2\lambda_2 h'' + 2\lambda_3 h^{(4)} = 0}$$

- Dont la solution générale est :

$$h(x) = e^{-\alpha x}(a_1 \sin \omega x + a_2) \cos \omega x) + e^{\alpha x}(a_3 \sin \omega x + a_4 \cos \omega x) - \frac{\lambda_1}{2}$$

- Avec $\lambda_1 - \frac{\lambda_2^2}{4} > 0$, $\alpha^2 - \omega = \frac{\lambda_2}{2\lambda_4}$ et $4\alpha^2\omega^2 = \frac{4\lambda_4 - \lambda_2}{4\lambda_4^2}$

Résolution : approche de Canny (3/3)

- Avec les conditions aux limites :

$$\begin{aligned} h(0) &= h(-W) = h'(W) = 0 \\ h'(0) &= c_3 \end{aligned}$$

- En utilisant une technique d'optimisation numérique sous contrainte, Canny a trouvé les valeurs optimales suivantes :

$$\begin{aligned} \Sigma\Lambda &= 1,12 \\ k &= 0,58 \end{aligned}$$

Résolution : approche de Dériche (1/2)

- Dériche a proposé d'utiliser un filtre à réponse impulsionnelle infinie (*RII*)

$$h_2(x) = a_3 e^{-\alpha|x|} \sin \omega x$$

- On calcule ainsi les expressions :

$$\Lambda = \sqrt{2\alpha} \qquad \Sigma = \sqrt{\frac{2\alpha}{\alpha^2 + \omega^2}}$$

$$\Sigma\Lambda = \frac{2\alpha}{\alpha^2 + \omega^2} \qquad k = \sqrt{\frac{\alpha^2 + \omega^2}{5\alpha^2 + 6\alpha^2\omega^2 + \omega^4}}$$

- On pose $\alpha = \omega m$

Résolution : approche de Deriche (2/2)

- Pour $m \ll 1$, on a $\Sigma \simeq \sqrt{\frac{2\alpha}{\alpha^2/m^2}} \Rightarrow \Lambda\Sigma \approx 2m$ donc $k \simeq 1$
  - k excellent
  - $\Lambda\Sigma$ est très petit car m faible : aucun intérêt
- Pour $m = 1$, on a $\Sigma \simeq \sqrt{\frac{1}{\alpha}} \Rightarrow \Lambda\Sigma \approx \sqrt{2}$ donc $k \simeq 0,58$
  - k identique à celui de Canny
  - $\Lambda\Sigma$ meilleur de 25% par rapport à celui de Canny
- Pour $m \gg 1$, on a $\Sigma \simeq \sqrt{\frac{2}{\alpha}} \Rightarrow \Lambda\Sigma \approx 2$ donc $k \simeq 0,44$
  - Ce cas correspond à la limite de l'opérateur pour $\omega \rightarrow 0$ $(\sin \omega x \approx \omega x)$

Pour résumer

- La valeur SNR doit être la plus grande possible : on veut beaucoup de signal et peu de bruit
- La valeur de la localisation représente l'inverse de la distance du contour détecté au vrai contour : on la veut la plus grande possible
- Le critère d'unicité est une contrainte supplémentaire pour éviter les réponses multiples et représente la distance moyenne entre les passages par zéro de f'
- Canny cherche le filtre qui maximise le produit du SNR et de la localisation sujet à la contrainte d'unicité de la réponse
  $\rightsquigarrow$ parfois compliqué à résoudre : on l'approche par la dérivée première d'une fonction gaussienne.

Algorithme de Canny-Deriche simplifié

1. Lire l'image I
2. Créer un masque Gaussien $1D$ G pour convoluer I (σ à paramétrer)
3. Créer un masque $1D$ pour la dérivée première de G dans les directions en x et y (G_x et G_y)
4. Convoluer I et G selon les lignes puis les colonnes (I_x et I_y)
5. Convoluer I_x avec G_x (I'_x) et I_y avec G_y (I'_y)
6. Calculer la magnitude $M(x,y) = \sqrt{I'_x(x,y)^2 + I'_y(x,y)^2}$

Suivi de contour (1/2)

- L'extraction de points de contour nous permet d'obtenir une image binaire
  - Pixels blancs : points de contour
  - Pixels noirs : autres pixels de l'image
- Problèmes à résoudre :
  - Certains points de contour n'ont pas été détectés
  - Certains pixels détectés comme points de contour n'en sont pas
- Nécessité d'obtenir les contours fermés des objets/régions de l'image

Suivi de contour (2/2)

- Grouper les points de contours en chaînes et relier ces chaînes afin d'obtenir des "formes"
- On peut ensuite obtenir des informations sur ces formes : courbure, pente, coins, etc.
- Procédure :
  1. Affiner les composantes contour connectées à une épaisseur de 1 pixel
  2. Trouver des chemins simplement connectés
  3. Les lier entre eux en utilisant une modélisation par graphe de l'image des contours
  4. Calculer les propriétés du contour obtenu

Affinage (*thinning*) (1/4)

- On travaille sur l'image binaire contenant les points de contour
- La suppression d'un point de contour ne doit pas changer le nombre de composantes connectées dans l'image binaire
- Un point final est un pixel blanc ayant exactement un voisin
- Si on considère un voisinage 3×3 avec comme point central p un pixel blanc (point de contour), alors :

  $\hookrightarrow$ p est un point simple si on peut le changer de blanc en noir sans changer le nombre de composantes connectées dans le voisinage

Affinage (*thinning*) (2/4)

- Exemple 1 de voisinage : $\begin{matrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{matrix}$ $\Rightarrow p$ est simple-8 mais pas simple-4

- Exemple 2 de voisinage : $\begin{matrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{matrix}$ $\Rightarrow p$ n'est ni simple-8 ni simple-4

- Un 1-pixel blanc $I(x, y)$ (ou point final) est un bord-N si son voisin de coordonnée $(x, y-1)$ est noir (définitions similaires avec les directions S, E et O)

Affinage (*thinning*) (3/4)

- Un algorithme simple :

```
Pour D=N faire
    Éliminer tous les bords-D qui sont simples et ne sont pas des
    points finaux
Répéter pour les directions E, O et S
```

- Chaque direction est traitée séparément afin d'éviter l'élimination de composantes connexes entières
- Le résultat dépend de l'ordre de traitement des directions

Affinage (*thinning*) (4/4)

- Exemple d'application $N \rightarrow E \rightarrow O \rightarrow S$:

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \qquad \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \qquad \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

Représentation par un graphe (1/3)

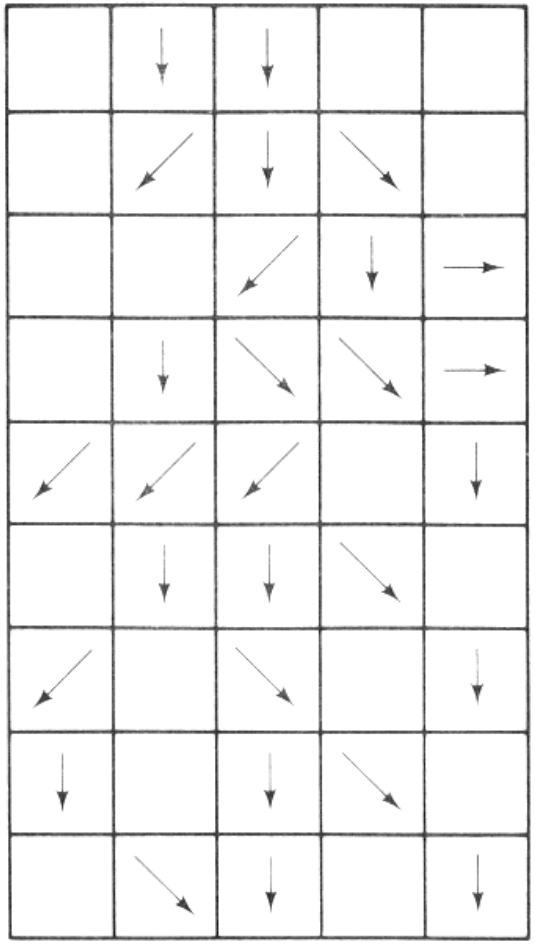
- But : créer une représentation structurée en graphe de l'image de contours
- On part d'une image binaire E des contours
- L'image est vue comme un maillage (i.e. graphe)
- Un pixel "point de contour" est un sommet de ce graphe
- Suivi de contour = recherche d'un chemin entre deux sommets du graphe

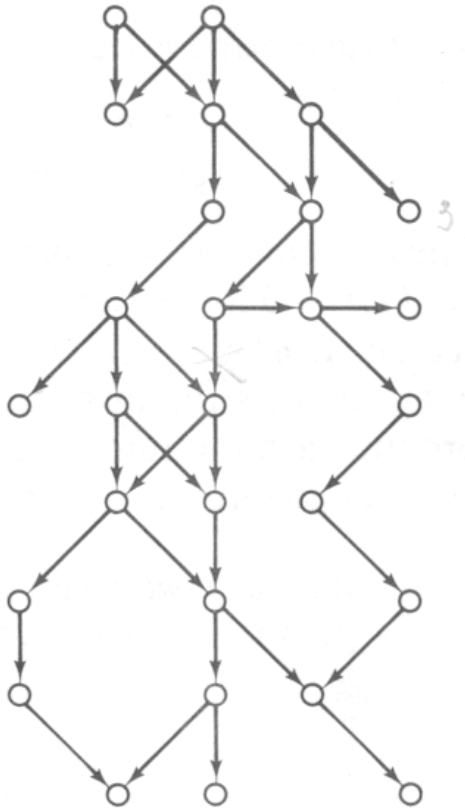
Représentation par un graphe (2/3)

- Exemple : on connaît l'orientation du gradient en chaque point de contour
- Deux pixels voisins détectés "points de contour" peuvent être reliés si les directions de leur gradient sont proches
- Application : pour relier le pixel a au pixel b, b doit être un des huit voisins de a dans la direction du contour en a ($\phi(a)$), soit :

$$|(\phi(a) - \phi(b)) \text{mod} 2\pi| \quad < \quad \frac{\pi}{2}$$

Représentation par un graphe (3/3)





Recherche heuristique (1/3)

- Soient a et b deux extrémités d'un contour
- Pour tout pixel p situé sur un chemin solution, on définit une fonction d'évaluation :

$$f(p) = g(p) + h(p)$$

où :

  - $g(p)$ est le coût du chemin de a à p
  - $h(p)$ est le coût du chemin de p à b

- Une solution : L'algorithme A^*

Recherche heuristique (2/3)

- Principe de l'algorithme A^* :

  $n = a$

  `tant que` $n \neq b$ `faire`

  Placer les successeurs de n dans L

  Déterminer p tel que $f(p) = \min_{q \in L}(f(q))$

  $n = q$

  Ajouter p à C

  `fin tant que`

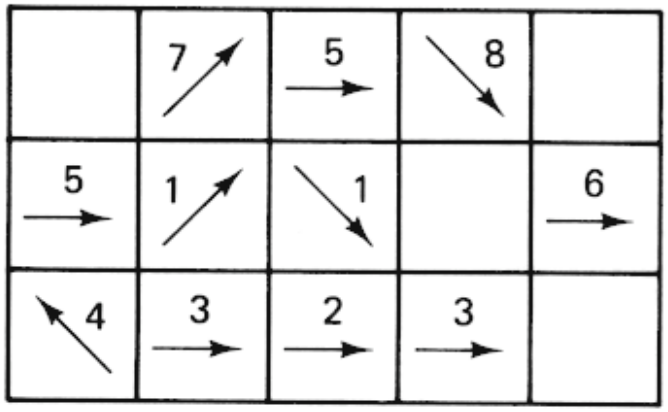
  `si` C est vide `alors` échec

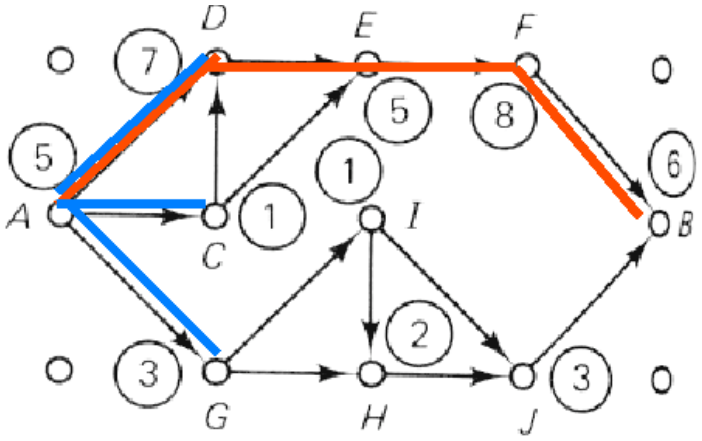
  `sinon` C contient le chemin optimal de a vers b

- Quelques valeurs intervenant dans le calcul de h :
  - Importance de l'arête : $||\max_p(\nabla I(p))|| - ||\nabla I(p)||$
  - Courbure : $\phi(p) - \phi(p')$
  - Distance à l'objectif : $d = \text{dist}(p, b)$

Recherche heuristique (3/3)

- Exemple d'illustration de fonction coût : somme (inverse) des modules du gradient





Programmation dynamique (1/4)

- Technique de résolution de problèmes d'optimisation
- On veut résoudre :

$$\max_{x_i}\{h(x_1, x_2, \ldots, x_n)\}$$

- On ne possède pas de connaissance a priori sur h
- Seule technique : énumération exhaustive des solutions possibles

Programmation dynamique (2/4)

- Soit $n = 4$ et le problème $\max_{x_i}\{h(x_1, x_2, x_3, x_4)\}$ à résoudre
- Si $h(\cdot) = h_1(x_1, x_2) + h_2(x_2, x_3) + h_3(x_3, x_4)$
- On a alors :

$$\begin{aligned} f_1(x_2) &= \max_{x_1}\{h_1(x_1, x_2)\} \\ f_2(x_3) &= \max_{x_2}\{f_1(x_2) + h_2(x_2, x_3)\} \\ f_3(x_4) &= \max_{x_3}\{f_2(x_3) + h_3(x_3, x_4)\} \end{aligned}$$

- Et finalement :

$$\max_{x_i}\{h(x_1, x_2, x_3, x_4)\} = \max_{x_4}\{f_3(x_4)\}$$

Programmation dynamique (3/4)

- En généralisant :

$$\begin{aligned} f_0(x_1) &= 0 \\ f_{n-1}(x_n) &= \max_{x_{n-1}}\{f_{n-2}(x_{n-1}) + h_{n-1}(x_{n-1}, x_n)\} \\ \max_{x_i}\{h(x_1, x_2, \ldots, x_n)\} &= \max_{x_n}\{f_{n-1}(x_n)\} \end{aligned}$$

- Soit la fonction coût :

$$\begin{aligned} S(x_1, \ldots, x_n) &= \sum_{i=1}^{n} |\nabla I(x_i)| - \alpha \sum_{i=2}^{n} |\phi(x_i) - \phi(x_{i-1})| - \beta \sum_{i=2}^{n} d(x_i, x_{i-1}) \\ S(x_1, \ldots, x_n) &= S(x_1, \ldots, x_{n-1}) + |\nabla I(x_n)| - \alpha|\phi(x_n) - \phi(x_{n-1})| \\ &\quad -\beta d(x_n, x_{n-1}) \\ S(x_1, \ldots, x_n) &= S(x_1, \ldots, x_{n-1}) + f(x_{n-1}, x_n) \end{aligned}$$

Programmation dynamique (4/4)

- On a donc (en posant $n = k$) :

$$S(x_1, \ldots, x_k) = \max_{x_{k-1}} \{S(x_1, \ldots, x_{k-1}) + f(x_{k-1}, x_k)\}$$

- Avec : $S(x_1) = |\nabla I(x_1)|$
- Application de l'algorithme de Dijkstra sur le graphe $G(S, A)$, où :
  - S est l'ensemble des sommets du graphe, valués par la valeur du gradient au pixel correspondant
  - A est l'ensemble des arêtes du graphe, valuées par la distance entre les deux pixels de contour qu'elles relient

La transformée de Hough (1/2)

- Permet de rechercher des formes particulières et simples dans l'image des points de contour :
  - Détection de lignes
  - Détection de cercles, ellipses
- Les formes géométriques ont une forme paramétrique :
  - La droite : $y = ax + b$ ou $\rho = x\cos\theta + y\sin\theta$
  - Le cercle : $(x-a)^2 + (y-b)^2 = r^2$
  - La facette plane : $ax + by + c = z$
  - Les surfaces quadratiques, cubiques :
    $z = a_{00} + a_{10}x + a_{20}y + a_{11}xy + a_{20}x^2 + a_{02}y^2 + a_{21}x^2y + \ldots$

La transformée de Hough (2/2)

- Principe : changer d'espace de représentation

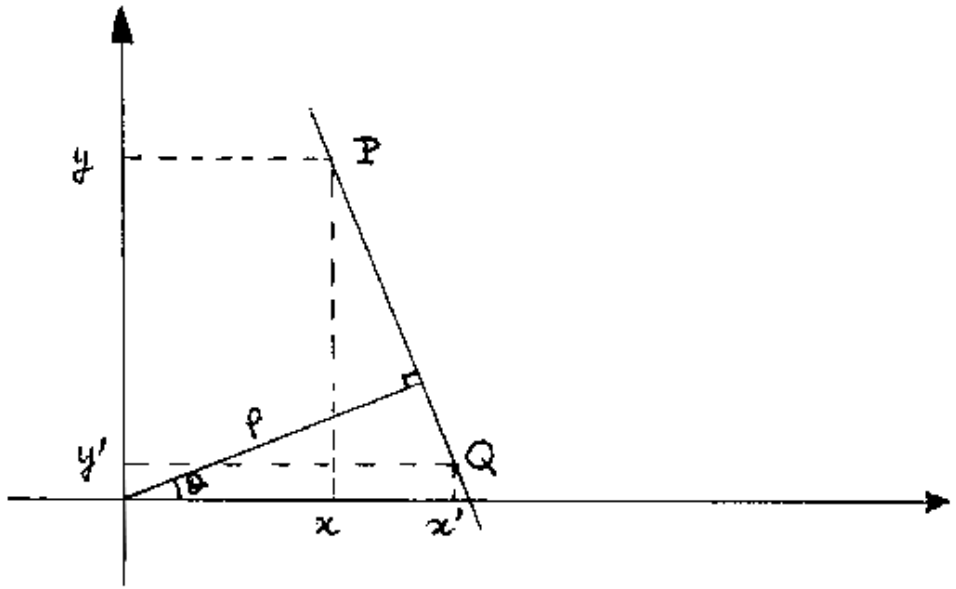
  $\hookrightarrow$ Passer de l'espace de l'image à celui des paramètres $I \rightarrow \Re^p$, soit $(x, y) \rightarrow (a_1 \cdots a_p)^t$

- Algorithme général :
  1. En chaque point, calculer la(les) vecteur(s) de paramètres le(s) plus probable(s)
  2. Dans l'espace $\Re^p$, pour chaque point (vecteur de paramètres), calculer le nombre de points ayant choisi ce vecteur de paramètres
  3. Garder le vecteur de paramètres apparu le plus souvent dans le choix des pixels de l'image

- L'espace discret des paramètres est également appelé accumulateur

La transformée de Hough : détection de droites (1/3)

- Soit $P(x, y)$ un pixel détecté point de contour. Ce point appartient à toutes les droites d'équation $\rho = x \cos\theta + y \sin\theta$
- On a deux données (x et y) et deux inconnues (ρ et θ), avec une infinité de solutions possibles
- On doit utiliser une deuxième contrainte, un point de contour $Q(x', y')$



La transformée de Hough : détection de droites (2/3)

- Soit une image I de taille $N \times N$, alors les droites ayant une intersection avec I sont caractérisées par :

$$\begin{aligned} \rho &\in [0, \sqrt{2}N] \\ \theta &\in]-\frac{\pi}{2}, \pi[\end{aligned}$$

- Si on utilise un accumulateur de taille $N \times N$, alors la précision sur le résultat sera au mieux $(\Delta\rho, \Delta\theta) = (\sqrt{2}, \frac{3\pi}{2N})$
- Exemple d'une image de taille 512×512 : $(\Delta\rho, \Delta\theta) = (1.4, 0.5)$

  $\hookrightarrow$ En fait toutes les valeurs de l'accumulateur ne correspondent pas à une droite traversant l'espace image
- Chaque élément (ρ, θ) de l'accumulateur A est un compteur incrémenté par les sources d'information (pixel) votant pour la droite

La transformée de Hough : détection de droites (3/3)

- Calcul de l'accumulateur A en utilisant une carte de contours comme espace image I :
  1. $A \leftarrow 0$
  2. `Pour tout` couple de pixels p et q de I
     - Calculer les coordonnées (ρ, θ) de la droite passant par p et q
     - Convertir les paramètres en indices : $(\tilde{\rho}, \tilde{\theta}) = (\frac{\rho}{\Delta\rho}, \frac{\theta}{\Delta\theta})$
     - Mise à jour de l'accumulateur : $A(\tilde{\rho}, \tilde{\theta}) \leftarrow A(\tilde{\rho}, \tilde{\theta}) + 1$
  3. `fin pour`
  4. Garder le couple (ρ^*, θ^*) pour lequel $A(\rho^*, \theta^*)$ est maximal
- Si on a n pixels de contour, la complexité est en $O(n^2)$

La transformée de Hough : détection de cercles

- Un cercle de centre (a, b) et de rayon r a pour équation :

$$(x - a)^2 + (y - b)^2 = r^2$$

- L'accumulateur A est un tableau à 3 dimensions (a, b, r)
- La cellule $A(a, b, r)$ est augmentée de 1 si le point (a, b) est à une distance r du point de fort gradient considéré (x, y)
- Complexité en $O(n^3)$
- Pour une ellipse, il faut rajouter deux paramètres : la largeur et la hauteur de l'ellipse $\Rightarrow$ complexité en $O(n^5)$