

Vision par ordinateur

Vision pré-attentive:
Détection de contours
Détection de points d'intérêt

Frédéric Devernay

Introduction

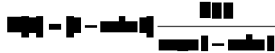
- Amélioration d'images : histogrammes, lissage, filtrage médian
- Contours d'une image de dimension quelconque (2D-3D) = extrema locaux du gradient ou zéros du laplacien
- Difficulté : tenir compte du bruit dans les images
- Filtres optimaux : meilleur compromis détection-localisation en tenant compte du bruit
- Implantation : convolution ou filtre récursif

Histogrammes

- Fonction :
 $i \rightarrow \text{hist}(i) = \text{nb de pixels d'intensité } i \text{ dans l'image}$
- Histogramme normalisé :



- Modification d'histogramme : $i \rightarrow i' = T(i)$
- T est monotone, et $0 \leq T(i) \leq 255$
- Recadrage ($T(\min i) = 0, T(\max i) = 255$) :



Histogrammes : égalisation

- Egalisation : trouver T tel que 
- Supposons p, p' et T continues, T croissante,

$$p'(i') = p(i) \frac{di}{di'}$$

avec $i' = T(i)$. D'où :

$$di' = \text{MaxInt } p(i) di,$$

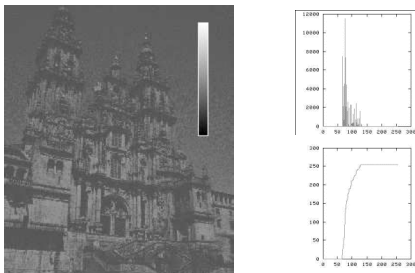
soit :

$$i' = T(i) = \text{MaxInt} \int_0^i p(w) dw.$$

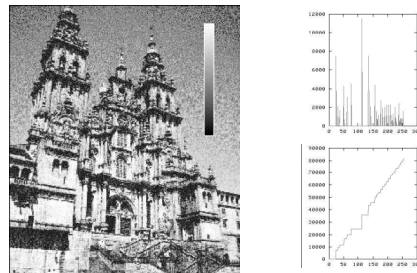
Dans le cas discret, la transformation s'écrit :

$$T(i) = \text{MaxInt} \sum_{j=0}^i \text{hist}(j) / \sum_{j=0}^{\text{MaxInt}} \text{hist}(j),$$

Avant égalisation



Après égalisation



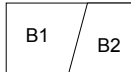
Contours : Généralités

- Filtrage linéaire = convolution de $I(x,y)$ par réponse impulsionnelle $f(x,y)$:

$$\begin{aligned} I_f(x, y) &= (f * I)(x, y) \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x', y') I(x - x', y - y') dx' dy' \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x - x', y - y') I(x', y') dx' dy' \end{aligned}$$

- Contour de type "marche" :

$$I(x, y) = B_1 + (B_2 - B_1)u(-x \sin \theta + y \cos \theta + \rho)$$



- Gradient nul partout sauf sur contour : orthogonal au contour

D'autres types de contours

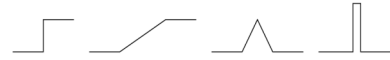


Figure 2.1. Quelques profils de contours : marche, rampe, toit, pic.

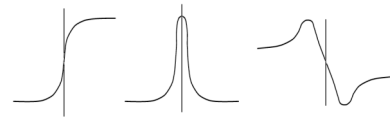


Figure 2.2. Les deux principes de la détection de contours : dérivée première et dérivée seconde (discontinuité du signal, dérivée première, dérivée seconde).

Filtrage non-linéaire

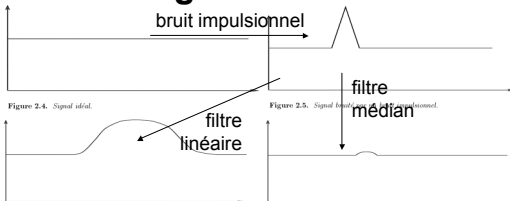


Figure 2.4. Signal lissé par un filtre linéaire : pas d'effet !

Figure 2.5. Signal lissé par un filtre médian de dimension 3.

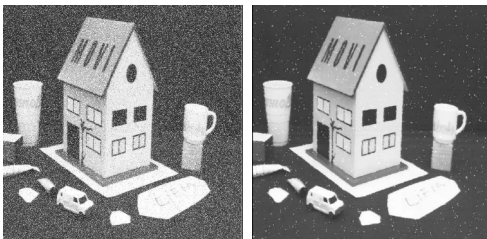
Filtre médian pour le bruit impulsionnel

Soit $(i_1, i_2, \dots, i_n)$ la suite des valeurs de I
 Soit $(j_1, j_2, \dots, j_n)$ la suite des valeurs de I après filtrage
 j_p : valeur médiane de (i_{p-1}, i_p, i_{p+1})
 $j_p = i_p$ si $i_p \leq i_{p+1}$ et $i_{p-1} \leq i_p$
 $j_p = i_{p-1}$ si $i_{p+1} \leq i_{p-1}$ et $i_{p-1} \leq i_p$...

Filtrage médian

- Permet d'éliminer un bruit type « speckle » (chatoiement) ou « poivre et sel »
- En chaque point de l'image, considérer la liste des valeurs d'intensité dans un voisinage (3x3, 5x5, ...)
- Trier cette liste de n valeurs
- Attribuer au pixel la valeur médiane des intensités, $I(n/2)$
- Problème : risque de « déplacer » les contours

Filtrage médian : 3x3



Filtrage médian : 5x5



Contours à partir des dérivées

- **Approche gradient** : extrema locaux de la norme du gradient dans la direction du gradient (= zéros de la dérivée seconde dans la direction du gradient)

$$G(x, y) = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right)^t$$

- **Approche laplacien** : passages par zéro du laplacien

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

Dérivation

- La dérivée du signal I convolué avec f :

$$(I * f)' = I * f'$$

$$(I * f)'' = I * f''$$

- On peut donc calculer directement les dérivées du signal convolué par convolution avec la dérivée de la réponse impulsionnelle

Filtrage séparable

- Filtres dont la réponse impulsionnelle est séparable, c.-à-d. :

$$f(x, y) = f_1(x)f_2(y)$$

- La convolution 2D revient à effectuer 2 convolutions 1D en cascade :

$$I * (f_1(x)f_2(y)) = (I * f_1(x)) * f_2(y)$$

- Temps de calcul en $2p$ au lieu de p^2

Masques de Sobel

Gradient par convolution par les filtres 3x3 :

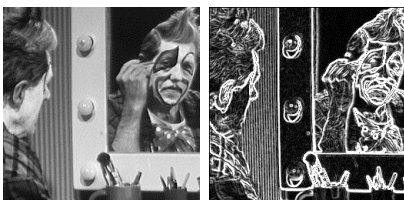
$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 1 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

On réécrit ces filtres :

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

Filtres classiques de calcul du gradient, implantation hardware simple.

Gradient par Sobel



- Démo HIPR2

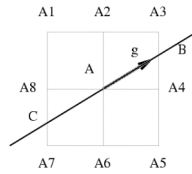
Filtres de Scharr

$$\begin{bmatrix} -3 & 0 & 3 \\ -8 & 16 & 8 \\ -3 & 0 & 3 \end{bmatrix} \quad \begin{bmatrix} 3 & 8 & 3 \\ 16 & 16 & -8 \\ 3 & 0 & -3 \end{bmatrix}$$

- Meilleurs que Sobel :
 - Plus isotropes,
 - Plus proches du filtre gaussien
 - Arithmétique entière, donc reste rapide

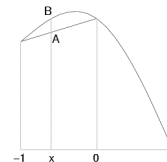
Approche gradient

- Calcul de la direction du gradient en A
- Norme du gradient en B et C par interpolation linéaire entre resp. (A3,A4) et (A7,A8)
- A est un point de contour si $I_A \geq I_B$ et $I_A > I_C$



Approche gradient : interpolation subpixel

- Position du contour = maximum de la parabole passant par les trois valeurs de la norme du gradient



$$m = \frac{a - c}{2(a - 2b + c)}$$

Interpolation subpixel : démonstration en 1D

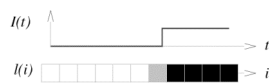
$$I(t) = \begin{cases} 0 & \text{if } t > \alpha - \frac{1}{2} \\ 1 & \text{otherwise} \end{cases}$$

$$l(i) = \int_{-\frac{1}{2}}^{\frac{1}{2}} I(t) dt$$

$$l(-\infty) = \dots = l(-2) = l(-1) = 1$$

$$l(0) = \alpha, 0 \leq \alpha \leq 1$$

$$l(1) = l(2) = \dots = l(+\infty) = 0$$



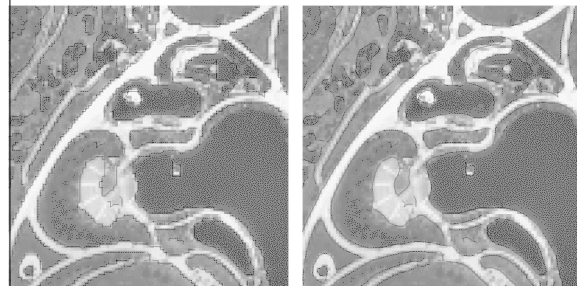
$$\nabla l(i) = \sum_{k=-\infty}^{+\infty} g_k (l(i+k) - l(i-k)) = \dots - g_2 l(i-2) - g_1 l(i-1) + g_1 l(i+1) + g_2 l(i+2) + \dots$$

$$\begin{aligned} \nabla l(-1) &= \dots - g_2 - g_1 + g_1 \alpha = C - (1 - \alpha) g_1 \\ \nabla l(0) &= \dots - g_2 - g_1 = C - g_1 \\ \nabla l(1) &= \dots - g_2 - g_1 \alpha = C - \alpha g_1 \end{aligned}$$

$$C = - \sum_{k=2}^{\infty} g_k$$

$$m = \alpha - \frac{1}{2}$$

Interpolation subpixel



sans

avec

NMS et seuillage par hystérésis

- On commence par supprimer les points qui ne sont pas des maxima locaux de la norme du gradient (Non-Maxima suppression)
- On ne conserve que les contours (=composantes connexes) composés de points tous supérieurs au seuil bas sb et dont au moins un est supérieur à sh

Seuillage par hystérésis

- Calculer images I_h ($N > sh$) et I_b ($N > sb$)
- Réaliser une expansion en composantes connexes de I_h sur tous les points de I_b
- Peut être amélioré en ne réalisant une expansion que dans la direction du contour

Filtrage optimal (Canny)

- Étant donnés :
 - un modèle de contour (marche)
 - un modèle de bruit (blanc gaussien)
- Caractériser les performances en termes de :
 - détection (réponse au voisinage du contour)
 - localisation (contour détecté proche contour réel)
 - réponse unique (un contour = une réponse)
- En déduire une famille de filtres optimaux vis-à-vis de ces critères

Modèle de contour

Contour type "marche" avec bruit blanc gaussien:

$$I(x) = Au_{-1}(x) + \eta(x)$$

u_{-1} : fonction de Heaviside

$$u_{-1}(x) = \begin{cases} 0 & \text{pour } x \text{ dans }]-\infty, 0] \\ 1 & \text{pour } x \text{ dans }]0, +\infty[\end{cases}$$

$\eta(x)$: bruit gaussien

$$\eta_0^2 = E[\eta^2(x)]$$

Détection par convolution :

$$\Theta(x_0) = \int_{-\infty}^{+\infty} I(x)f(x_0 - x)dx$$

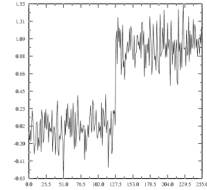


Figure 3: Noisy test signal with SNR=1

Détection

Bonne probabilité de détection = Faible probabilité de ne pas détecter un vrai point de contour + faible probabilité de marquer de faux points de contours = **Rapport Signal/Bruit au point de discontinuité (RSB)**

$$\Sigma = \frac{A \int_{-\infty}^0 f(x)dx}{n_0 \sqrt{\int_{-\infty}^{+\infty} f'^2(x)dx}}$$

Localisation

- Les points détectés comme contours doivent être proches du vrai contour = maximiser écart type de la position des passages par zéro = inverse de l'espérance de la distance point de contour vrai - détecté

$$\Lambda = \frac{A |f'(0)|}{n_0 \sqrt{\int_{-\infty}^{+\infty} f'^2(x)dx}}$$

Réponse unique

- Pas de réponses multiples
- Distance moyenne entre les passages par zéro de la réponse au bruit gaussien :

$$x_{max} = 2\pi \sqrt{\frac{\int_{-\infty}^{+\infty} f'^2(x)dx}{\int_{-\infty}^{+\infty} f''^2(x)dx}}$$

- différent de détection, car ce critère prend en compte la réponse dans le voisinage
- ce critère utilise l'hypothèse "bruit gaussien"

Critère d'optimalité

- On maximise $\Sigma \cdot \Lambda$ sous la contrainte du 3ième critère
- On obtient ainsi une équ. diff. :

$$f(x) = a_1 e^{\alpha \cdot x} \sin(\omega \cdot x) + a_2 e^{\alpha \cdot x} \cos(\omega \cdot x) + a_3 e^{-\alpha \cdot x} \sin(\omega \cdot x) + a_4 e^{-\alpha \cdot x} \cos(\omega \cdot x)$$

- Sol. approchée = filtre gaussien :

$$g(x) = -\frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}} \text{ perf : } k=0.51 \quad \Sigma \Lambda = .92$$

Un détecteur optimal

- Solution de l'équation de Canny pour les filtres à réponse impulsionnelle infinie :

$$f(x) = \frac{S}{\omega} e^{-\alpha|x|} \sin(\omega x)$$

- Performances :

$$\Lambda = \sqrt{(2\alpha)} \quad \Sigma = \sqrt{\frac{2\alpha}{\alpha^2 + \omega^2}}$$

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

Différentes solutions...

- On pose $\alpha = m\omega$

$$\begin{array}{lll} 1 & m \gg 1 & \Lambda = \sqrt{2\alpha} \quad \Sigma = \sqrt{\frac{2}{\alpha}} \quad \Sigma\Lambda = 2 \quad k = .44 \\ 2 & m = 1 & \Lambda = \sqrt{2\alpha} \quad \Sigma = \sqrt{\frac{1}{\alpha}} \quad \Sigma\Lambda = \sqrt{2} \quad k = .58 \\ 3 & m = \sqrt{3} & \Lambda = \sqrt{2\alpha} \quad \Sigma = \sqrt{\frac{3}{2\alpha}} \quad \Sigma\Lambda = \sqrt{3} \quad k = .5 \end{array}$$

- Cas 3 : meilleur qu'une gaussienne
- Cas 1 : meilleur compromis = optimal

Un détecteur optimal (Deriche)

$$g(x) = Sx e^{-\alpha|x|}$$

- Solution exacte de l'équation de Canny
- Implantation récursive

$$y_1(n) = x(n-1) - b_1 y_1(n-1) - b_2 y_1(n-2)$$

$$y_2(n) = x(n+1) - b_1 y_2(n+1) - b_2 y_2(n+2)$$

$$y(n) = c[y_1(n) - y_2(n)]$$

$$c = (1 - e^{-\alpha})^2 \quad b_1 = -2e^{-\alpha} \quad b_2 = e^{-2\alpha}$$

Filtres dérivés de Deriche

- Lissage

$$l(n) = k_0(\alpha |n| + 1)e^{-\alpha|n|}$$

- Dérivée seconde

$$d2(n) = k_1(1 - \alpha |n|)e^{-\alpha|n|}$$

- Implantation récursive :

$$y_1(n) = a_0 x(n) + a_1 x(n-1) - b_1 y_1(n-1) - b_2 y_1(n-2)$$

$$y_2(n) = a_2 x(n+1) + a_3 x(n+2) - b_1 y_2(n+1) - b_2 y_2(n+2)$$

$$y(n) = k[y_1(n) + y_2(n)]$$

Passage en 2D

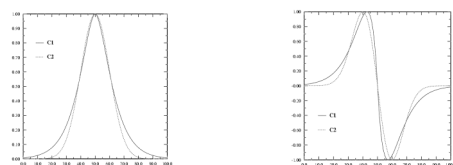
- Gradient en x = lissage sur colonnes, dérivées premières sur lignes (séparable)
- Orientation locale d'un contour :

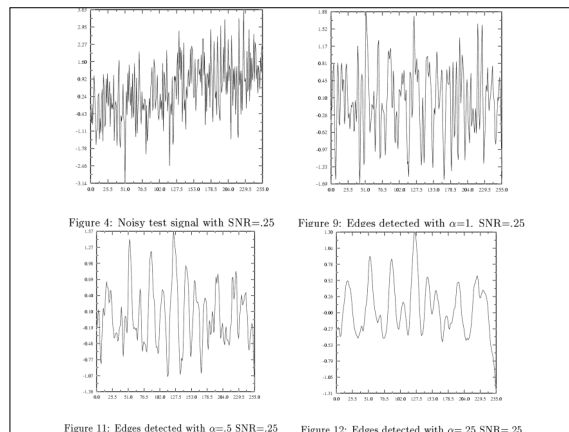
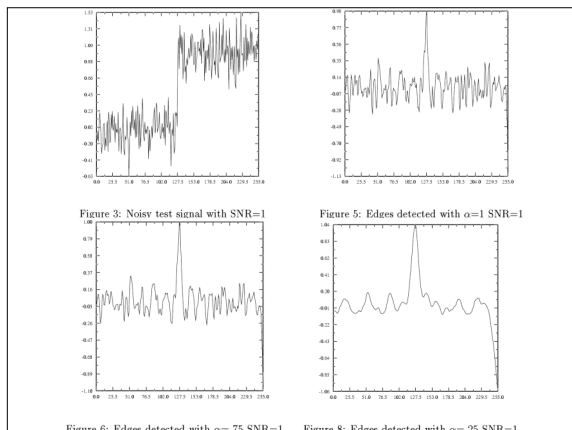
$$I_o(i, j) = \arctan\left(\frac{I_y(i, j)}{I_x(i, j)}\right)$$

- Norme du gradient :

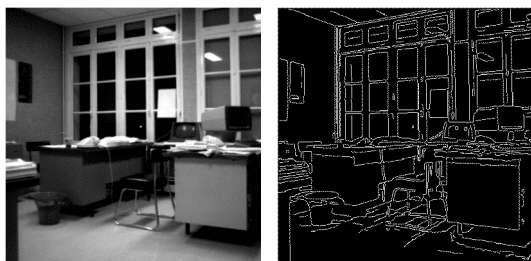
$$I_g(i, j) = \sqrt{I_x(i, j)^2 + I_y(i, j)^2}$$

Exemples





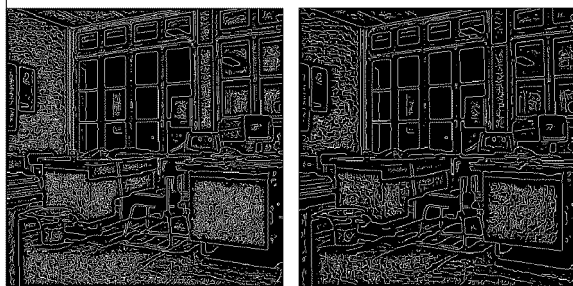
Contours par gradient+NMS



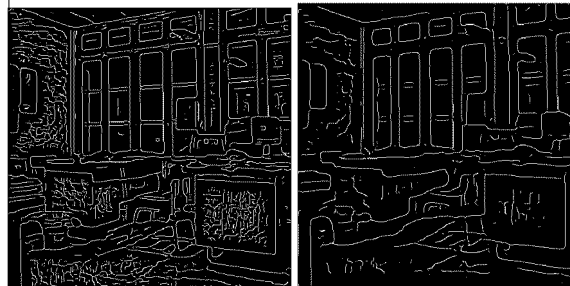
Contours par laplacien+ZCR



$\alpha = 1; \alpha = 0,75$



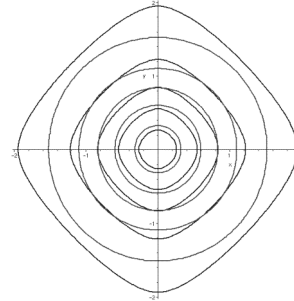
$\alpha = 0,5; \alpha = 0,25$



Filtres isotropes

- La réponse impulsionnelle doit être isotrope:
 $f(x,y) = f_r(x^2 + y^2)$
- Nécessaires pour dérivées directionnelles ou d'ordre élevé (courbures)
- Le filtre gaussien est séparable et isotrope
- On peut obtenir une approximation du filtre gaussien par un filtre récursif d'ordre 4 (Deriche 1993)

Isotropie/Anisotropie : exemple



Courbes de niveau de la réponse impulsionnelle du filtre 2D exponentiel (bleu) et gaussien (rouge)

Détection de contours : conclusion

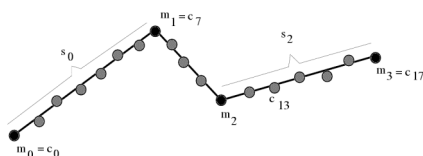
- Deux approches : laplacien ou gradient
- Les filtres de Sobel sont très simples mais ne permettent pas de gérer le bruit
- Les filtres de Deriche permettent la sélection d'une échelle α , ils sont séparables et récursifs
- Chaînage par ZCR ou NMS + hysteresis
- Rapide et isotrope = gaussien récursif

Références

- Radu Horaud et Olivier Monga, *Vision par ordinateur*, Éditions Hermès, chapitre 2.
- Rachid Deriche, *Fast Algorithms For Low-Level Vision*, Pattern Analysis and Machine Intelligence, 1988
- Rachid Deriche, *Recursively Implementing the Gaussian and its Derivatives*, Rapport de recherche No 1893, INRIA.

Segmentation de contours

- Chaîne = liste ordonnée de points $C = \{c_i\}$
- Points de cassure $M \subset C$, $M = \{m_k\} = \{c_{ik}\}$
- Segmentation = partitionner en segments s_k , et trouver la meilleure approximation de s_k (droite, arc de cercle...)



Approximer un segment par une droite

- Distance point i à la droite :

$$e_i = |x_i \sin \phi + y_i \cos \phi - d|$$

- On minimise : $E = \sum_{i=1}^n e_i^2$
- Solution :

$$d = V_x \sin \phi + V_y \cos \phi$$

$$\phi = \frac{1}{2} \arctan \frac{2V_{xy}}{V_{yy} - V_{xx}}$$

- avec :

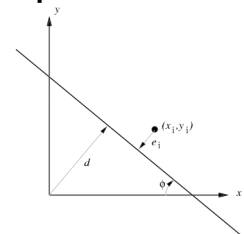
$$V_x = \frac{1}{n} \sum_{i=1}^n x_i$$

$$V_y = \frac{1}{n} \sum_{i=1}^n y_i$$

$$V_{xx} = \frac{1}{n} \sum_{i=1}^n (x_i - V_x)^2$$

$$V_{yy} = \frac{1}{n} \sum_{i=1}^n (y_i - V_y)^2$$

$$V_{xy} = \frac{1}{n} \sum_{i=1}^n (x_i - V_x)(y_i - V_y)$$



Découpage récursif

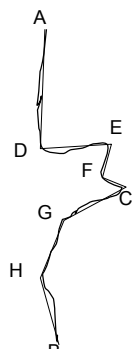
Est-ce que la chaîne considérée est une droite ($e_{\max} < \epsilon$) ?

- Si oui, fin
- Si non, diviser en deux sous-chaînes (au point le + éloigné des deux extrémités) et répéter l'algo pour chaque sous chaîne

Implantation non récursive

- Un tableau CLOSED initialisé à A
- Un tableau OPEN initialisé à B
- Segment considéré : (dernier de CLOSED, dernier de OPEN)
- Le segment est une droite: le dernier de OPEN est placé en fin de CLOSED
- Création d'un point de cassure : Le point est ajouté en fin de OPEN

Exemple



CLOSED	OPEN	Segment considéré	Sommet créé
A	B	AB	C
A	B, C	AC	D
A	B, C, D	AD	-
A, D	B, C	DC	E
A, D	B, C, E	DE	-
A, D, E	B, C	EC	F
A, D, E	B, C, F	EF	-
A, D, E, F	B, C	FC	-
A, D, E, F, C	B	CB	G
A, D, E, F, C	B, G	CG	-
A, D, E, F, C, G	B	GB	H
A, D, E, F, C, G	B, G	GH	-
A, D, E, F, C, G, H	B	HB	-
A, D, E, F, C, G, H, B	-	-	-

Problème : est-ce que CH n'est pas une droite ?

Split and Merge (Pavlidis & Horowitz 1974)

- L'algorithme de découpage récursif crée parfois des sommets en trop
- À chaque étape "Split", on ajoute une étape "Merge" qui teste si chacun des deux segments créés peut être fusionné avec le segment voisin, en utilisant le même seuil de distance

Détection de points d'intérêt

Qu'est-ce qu'un point d'intérêt ?

- Contour = discontinuité dans une direction de la fonction d'intensité ou de ses dérivées
- Point d'intérêt = dans 2 directions



- Coins, jonctions, discontinuités de texture...
- Avantages :
 - Contour = 1 contrainte, Pt d'intérêt = 2 contraintes
 - Visibilité totale ou nulle, jamais partielle
 - Présents dans toutes les images
 - Pas de chaînage ou de segmentation nécessaire

Deux approches

- À partir des niveaux de gris de l'image, trouver un opérateur qui est maximal aux points d'intérêt
- Effectuer une extraction de contours, suivie d'une recherche des points de courbure maximale

A partir des niveaux de gris

- L'opérateur doit être invariant par rotation

Beaudet '78: Déterminant de la hessienne $DET = I_{xx}I_{yy} - I_{xy}^2$

Kitchen & Rosenfeld: courbure de la surface image

multipliée par norme du gradient : $K = \frac{I_{xx}I_{yy}^2 + I_{yy}I_{xx}^2 - 2I_{xy}I_{xx}I_{yy}}{I_{xx}^2 + I_{yy}^2}$

Basés sur $\hat{C} = \begin{bmatrix} \hat{I}_x^2 & I_x \hat{I}_y \\ I_x \hat{I}_y & \hat{I}_y^2 \end{bmatrix}$:

Noble '88 : $C_p = \frac{Trace^2(\hat{C})}{Det(\hat{C})}$

Harris & Stephens '88 $R(x, y) = Det(\hat{C}) - kTrace^2(\hat{C})$
where $k \approx 0.04$

Lequel choisir ?

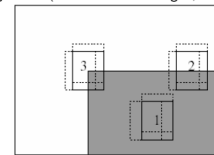
- Tous ces détecteurs sont basés sur un modèle de coin (de même qu'on avait le modèle "marche" pour les contours)
- Les points détectés ne correspondent pas forcément à un point physique :
 - tous les points ne sont pas des coins
 - le lissage pour la dérivation peut "déplacer" les coins

Détecteur de coins de Moravec (1980)

- Variation moyenne de l'intensité pour un petit déplacement (x, y) :

$$E(x, y) = \sum_{u, v} w(u, v) |I(x + u, y + u) - I(u, v)|^2$$

- w : fenêtre rectangulaire (1 dans le rectangle, 0 sinon)



- 1 - $E(x, y) \approx 0$
- 2 - $E(x, y) > 0$ pour $y \neq 0$, ≈ 0 sinon
- 3 - $E(x, y) > 0$ pour tout $(x, y) \neq (0, 0)$
- Coins: Maxima locaux du minimum de E pour $(x, y) \neq (0, 0)$

De Moravec à Harris

- D.L. de I : $I(x + u, y + v) = I(u, v) + x \frac{\delta I}{\delta x} + y \frac{\delta I}{\delta y} + o(x^2, y^2)$

- D'où : $E(x, y) = \sum_{u, v} w(u, v) [x \frac{\delta I}{\delta x} + y \frac{\delta I}{\delta y} + o(x^2, y^2)]^2$

- On écrit : $E(x, y) = (x, y) \cdot M \cdot (x, y)^t$

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix} \quad \begin{aligned} A &= \frac{\delta I^2}{\delta x^2} \otimes w \\ B &= \frac{\delta I^2}{\delta y^2} \otimes w \\ C &= \left(\frac{\delta I}{\delta x} \frac{\delta I}{\delta y} \right) \otimes w \end{aligned}$$

- On choisit w = gaussienne (+isotrope)
- M sym. définie positive $\Rightarrow$ orthogonalisable (décomposition en valeurs propres)

Harris et Stephen (1988)

- Coin : v.p. de M toutes deux élevées

$$R = Det(M) - kTrace(M)^2$$

- Rappel : Det = produit, Trace = somme des v.p.
- k est déterminé par l'expérience, en maximisant la réponse sur un modèle de coin idéal : $k = 0.04$
- Coins/Points d'intérêt = max. locaux de R

Corriger le déplacement des coins détectés du au lissage (Deriche & Giraudon)

- Utilise à la fois l'approche "intensité" et l'approche "contour" pour détecter les coins
- On calcule les maxima de l'opérateur à deux échelles (=lissages) différentes
- On prend l'intersection de la droite joignant les des points et des passages par zéro du laplacien

Exemples

