

## 1) Problème

Dans une image monochrome  $I: [0,1]^2 \rightarrow [0,1]$   
(en pratique de  $[0,w] \times [0,h] \rightarrow [0,255]$ )

les contours d'un objet ont un fort gradient:

$$\| \nabla I \| \text{ \u00e9lev\u00e9 }$$

## 2) Seuillage par hyst\u00e9resis

(a) - D\u00e9br\u00fbter l'image, par exemple en convoluant avec une gaussienne (en pratique, appliquer le noyau  $\frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$ )

(b) - Calculer les d\u00e9riv\u00e9es par diff\u00e9rence finie  
(appliquer  $(-\frac{1}{2} \ 0 \ \frac{1}{2})$  pour  $I_x = \frac{\partial I}{\partial x}, \dots$ )

NB: On r\u00e9alise (a) et (b) en m\u00eame tps avec par exemple Sobel  $\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$ . Mieux on utilise des filtres r\u00e9cursifs (Deriche) impl\u00e9mentant rapidement et exactement des filtres "optimaux" c'est-\u00e0-dire adapt\u00e9s \u00e0 un profil id\u00e9al de contour (Canny)

(c) Calculer  $\| \nabla I \|^2$

(d) NMS: ne conserver dans la suite que les points max. dans la direction du gradient. de  $\| \nabla I \|^2$

(e) Choisir deux seuils  $s_1 > s_2$  et:

- extraire les points tq  $\| \nabla I \|^2 > s_1$  (graine)

- extraire les points de  $\| \nabla I \|^2 > s_2$  reli\u00e9s aux grains par d'autres points de  $\| \nabla I \|^2 > s_2$

### 3) Snakes

Inconvénient de la méthode précédente: pas d'information globale sur le bord d'un objet (des contours à l'intérieur, pas de contour complet au bord, etc...)

→ on modélise maintenant un contour comme une courbe  $C: [0,1] \rightarrow [0,1]^2$  dépendant de paramètres  $(d_i)_{1 \leq i \leq N}$  (en pratique, B-spline) et on cherche:

$$\max_{(d_i)} E(c) \text{ avec } \boxed{E(c) = \int \|\nabla I(c(p))\| dp + \text{lissage}}$$

(en général, lissage =  $\int \|c'\|$  ou  $\int \|c''\|$ )

→ on obtient ce maximum par descente de gradient

→ Problèmes: 1) Non intrinsèque (dépend de la forme de  $C$  mais aussi de la paramétrisation  $p$ )  
2) Minima locaux → commencer la desc. de gradient près de la solution.

### 4) Contours actifs

$C(p)$  est maintenant fermé ( $C(0) = C(1)$ ) et ne dépend plus de paramètres  $(d_i)$ . On minimise  $E(c)$

avec  $\boxed{E(c) = \int g(\|\nabla I(c(p))\|) d\sigma(p)}$

avec  $g$   $f^{\circ}$  positive décroissante (par ex  $\frac{1}{1+c_1^2}$ )

$\sigma$  = abscisse curviligne de  $C$

La descente de gradient devient:

$$\begin{cases} C(p, 0) = C^0(p) \\ \frac{\partial C}{\partial t}(p, t) = -\nabla_c E(c)(p, t) \end{cases}$$

4.1) Gradient de forme

Soit  $\vec{w}(p)$  un champ de vecteurs défini sur  $C(p)$

$$\text{alors } D_{\vec{w}} = \lim_{\lambda \rightarrow 0} \frac{E(C(p) + \lambda \vec{w}(p)) - E(C(p))}{\lambda}$$

(Dérivée de Gâteaux)

Le gradient de  $E$  en  $C$  est le champ  $\nabla_C E$  tel que:

$$\forall \vec{w}, D_{\vec{w}} = \langle \nabla_C E | \vec{w} \rangle$$

$$\text{avec } \langle \vec{u} | \vec{v} \rangle = \int_C \vec{u} \cdot \vec{v} \, dp$$

Exemples

• Si  $F(C) = \int_{\Omega_i} f$  avec  $\Omega_i$  intérieur de  $C$  alors  $\nabla_C F = f \vec{N}$  ( $\vec{N}$  normale <sup>extérieure</sup> à  $C$ )

• Si  $F(C) = \int_C ds$  (= longueur de  $C$ !) alors  $\nabla_C F = \kappa \vec{N}$  ( $\kappa$  courbure signée)

4.2) Cas des contours actifs

Pour  $E(C) = \int g(\|\nabla I\|) ds$  on trouve

$$\boxed{\nabla_C E = +g(\|\nabla I\|) \kappa \vec{N} + (\vec{\nabla} g \cdot \vec{N}) \vec{N}}$$

En pratique, on accélère la convergence en rajoutant ~~on~~ une force dépendant d'une constante  $C_2$  et en partant d'un  $C^0$  entourant l'objet:

$$\begin{cases} C(t=0) = C^0 \\ \frac{\partial C}{\partial t} = (g(\kappa + C_2) - \vec{\nabla} g \cdot \vec{N}) \vec{N} \end{cases}$$

## 5) Régions actives

(4)

Lorsqu'on dispose d'un modèle de l'intérieur ou de l'extérieur du contour (par ex un a-priori sur les couleurs ou la texture de l'objet à segmenter ou du fond):  $E = \int_C g d\sigma + \int_{\Omega_i} g_i dx + \int_{\Omega_e} g_e dx$

avec  $\Omega_i$  (resp  $\Omega_e$ ) intérieur (resp. extérieur) de  $C$  et  $g_i$   $g_e$  fonctions modèles.

Le gradient devient:

$$\left[ \frac{\nabla E}{C} = g \kappa \vec{N} + (\vec{\nabla} g \cdot \vec{N}) \vec{N} + (g_i - g_e) \vec{N} \right]$$

NB: alors que le minimum global des contours actifs est la courbe vide, celui des régions actives ne l'est plus  $\rightarrow$  importance des termes de région!

## 6) Simulation

x Le pb se ramène à simuler  $\left\{ C(t=0) = C^0; \frac{\partial C}{\partial t} = \vec{v} \right\}$  (1)

x Discrétiser  $C(p,t)$  par un ensemble de sommets et faire  $C[p,t+\Delta t] \leftarrow C[p,t] + \Delta t \vec{v}[p,t]$   
n'est ni précis, ni stable.

x Level Sets :

- On représente  $C$  par une fonction implicite  $\varphi(x,y)$  nulle sur  $C$ , négative à l'intérieur et positive à l'ext.

$$\left[ C(\cdot, t) = \{ (x,y) \mid \varphi(x,y,t) = 0 \} \right]$$

On peut montrer que (1) est équivalent à :

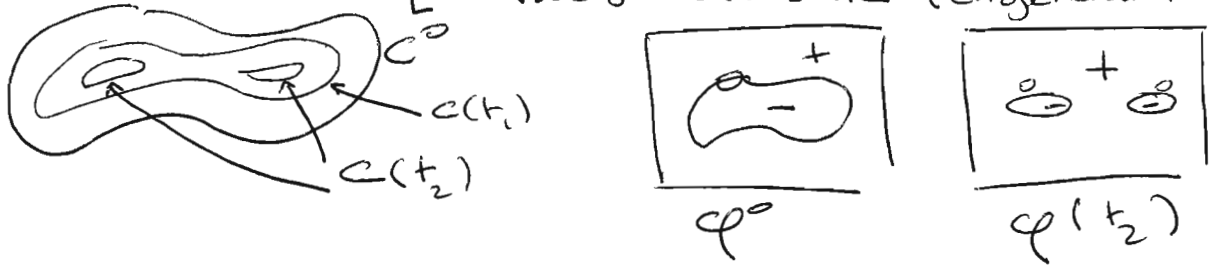
$$\left[ \begin{array}{l} \varphi(x,y,0) = \varphi^0(x,y) \\ \frac{\partial \varphi}{\partial t}(x,y,t) = -\vec{v} \cdot \nabla \varphi \end{array} \right] (2)$$

(Si  $\varphi^0$  est une rep. implicite de  $C^0$ )

Il suffit donc de discrétiser  $\varphi$  sur une grille régulière et d'implémenter un schéma numérique pour (2) adapté aux  $\neq$ ts choix de  $\vec{v}$ . (5)

La méthode est :

- précise et stable
- gère les changements de topologie (i.e. split ou merge du contour)
- Valable dans  $\mathbb{R}^d$  (en général  $\mathbb{R}^3$ )



NB : il faut savoir passer de  $C$  à  $\varphi$  et réciproq<sup>+</sup>

1)  $C \rightarrow \varphi$  (normalement, seul<sup>t</sup> pour  $C^0$ )

On choisit  $\varphi^0 = \underline{\text{distance signée à } C^0}$

2)  $\varphi \rightarrow C$  (pas à chaque pas de temps, mais quand on veut récupérer  $C$  pour l'afficher ou après convergence)

Algorithme des marching cubes qui fournit un maillage du niveau  $\varphi$  de  $\varphi$  avec une précision supérieure à la grille :

