

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 Ω_i intérieur de C alors

$$\nabla_C F = f \vec{N} \quad (\vec{N} \text{ normale } \overset{\text{extérieure}}{\text{à } C})$$

• Si $F(C) = \int_C ds$ (= longueur de C !) alors

$$\nabla_C F = \kappa \vec{N} \quad (\kappa \text{ courbure signée})$$

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 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 Ω_i (resp Ω_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 :

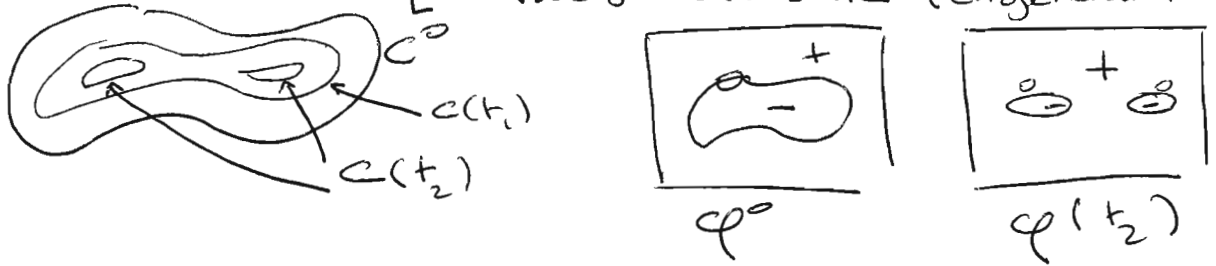
$$\left[\begin{array}{l} \varphi(x,y,0) = \varphi^0(x,y) \\ \frac{\partial \varphi}{\partial t}(x,y,t) = -\vec{v} \cdot \nabla \varphi \end{array} \right] (2)$$

(Si φ^0 est une rep. implicite de C^0)

Il suffit donc de discrétiser ϕ sur une grille régulière et d'implémenter un schéma numérique pour (2) adapté aux 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)



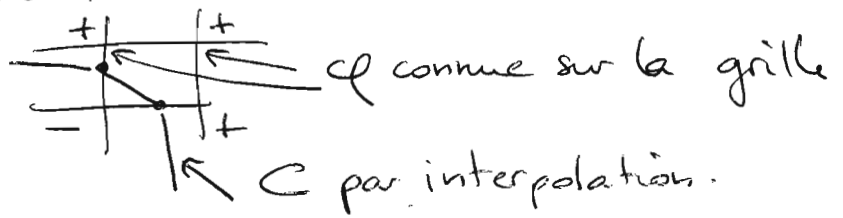
NB : il faut savoir passer de C à ϕ et réciproq⁺

1) $C \rightarrow \phi$ (normalement, seul^t pour C^0)

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

2) $\phi \rightarrow C$ (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 ϕ de ϕ avec une précision supérieure à la grille :



2) Graph Cuts

4

2.1) Coupes minimales

$G = (V, E)$ graphe orienté pondéré
 s, t deux sommets particuliers de V (source et puits)
 $w(p, q)$ poids de l'arête $(p, q) \in E$ ($w \geq 0$)

- une s, t -coupe C est une partition de V
en deux sous ensembles S et T avec $s \in S, t \in T$

- coût de la coupe:
$$c(C) = \sum_{\substack{p \in S \\ q \in T \\ (p, q) \in E}} w(p, q)$$

NB: on compte les arêtes de S vers T

→ Il existe des algorithmes polynomiaux pour calculer la coupe de coût minimal. Ils se ramènent à calculer le flot maximal de s vers t en identifiant $w(p, q)$ à des capacités (Ford-Fulkerson). Leur complexité au pire est en $O(n^3)$ mais linéaire en pratique, en particulier sur les graphes utilisés en computer vision (algorithme des "GraphCut" (Boykov et al. 7))

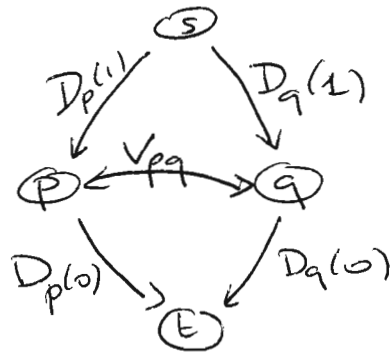
2.2) Segmentation binaire

Le problème des régions actives peut se modéliser comme l'étiquetage des pixels p à $f_p = 0$ ou 1 ($0 = \text{intérieur}$, $1 = \text{extérieur}$) qui minimise

$$E(f) = \sum_p D_p(f_p) + \sum_{p, q \text{ voisins}} V_{pq} \mathbb{1}\{f_p \neq f_q\}$$

$$\text{avec } D_p(0) = g_i(p), D_p(1) = g_e(p), \quad V_{pq} = g\left(\frac{p+q}{2}\right)$$

Or, minimiser $E(f)$ revient à trouver la min-cut dans ce graphe :



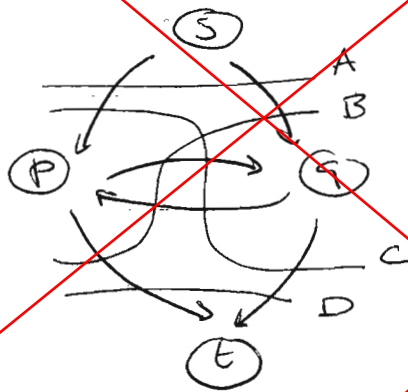
puis à faire : $f_p = 0$ si $p \in S$, $f_p = 1$ si $p \in T$

→ minimum global et rapide pour les régions actives ! (NB : le terme $\int g ds$ n'est que grossièrement approché. on peut l'approcher mieux avec des arcs supplémentaires)

2.3) Autres cas

a) Il est possible de construire un graphe pour un cas plus général $E(f) = \sum_p D_p(f_p) + \sum_{pq} V_{pq}(f_p, f_q)$ à condition que V_{pq} soit submodulaire :

$$V_{pq}(0,0) + V_{pq}(1,1) \leq V_{pq}(1,0) + V_{pq}(0,1) \quad (1)$$



Il faut trouver les bons poids pour que les coupes A, B, C et D contiennent les valeurs d'énergie adéquates → ce n'est possible avec des poids ≥ 0 que si on a (1)