TP7 – Contours actifs

Formulation variationnelle

Un « contour actif », ou *snake*, est un modèle de courbe déformable qui évolue au cours du temps pour se fixer sur les contours présents dans une image, ce qui peut aider l'utilisateur à segmenter des objets visibles dans une scène. Ce modèle est très utilisé, notamment en imagerie médicale (segmentation de tumeurs, ...).

Modélisation

Le *snake* est représenté par une courbe paramétrique $P(s) = [x(s), y(s)]^\top \in \mathbb{R}^2$, où $s \in [0; 1]$ est une abscisse curviligne. Pour s'assurer d'obtenir une courbe fermée, on impose de plus $P(1) = P(0)$. Le but est d'obtenir une courbe vérifiant les propriétés suivantes :

1. La courbe doit « coller » aux contours de l'image $\mathcal{I}$.
2. Elle doit être aussi courte que possible.
3. Elle doit osciller le moins possible.

La première de ces propriétés correspond à l'*attache aux données*, les deux autres sont des *régularisations* nécessaires pour assurer une certaine robustesse de la méthode au bruit et aux données manquantes. Chacune de ces propriétés peut être assurée par une énergie bien choisie. Dans le modèle original [Kass, Witkins et Terzopoulos, 1988], la régularité de la courbe est assurée par l'énergie suivante :

$$E_{\text{int}}(P(s)) = \frac{1}{2}\left(\alpha \left\|P'(s)\right\|^2 + \beta \left\|P''(s)\right\|^2\right) \tag{1}$$

où $\alpha > 0$ et $\beta > 0$ sont des paramètres choisis par l'utilisateur.

Quant à l'adéquation aux contours, elle est assurée en pénalisant localement les zones où le gradient de l'image est faible :

$$E_{\text{ext}}(P(s)) = -\left\|\nabla \mathcal{I}\left(P(s)\right)\right\|^2 \tag{2}$$

La fonctionnelle à minimiser s'écrit donc finalement :

$$E(P(s)) = \int_0^1 \left[E_{\text{ext}}(P(s)) + \frac{1}{2}\left(\alpha \left\|P'(s)\right\|^2 + \beta \left\|P''(s)\right\|^2\right)\right] ds \tag{3}$$

Optimisation

D'après le calcul des variations, l'équation d'Euler-Lagrange associée à cette fonctionnelle s'écrit :

$$\forall s \in [0,1], \quad \nabla E_{\text{ext}}(P(s)) - \alpha\, P^{(2)}(s) + \beta\, P^{(4)}(s) = 0 \tag{4}$$

En notant $\nabla E_{\text{ext}} = [F_x, F_y]^\top$, la solution de cette équation est atteinte lorsque l'itération suivante converge ($\gamma > 0$ est un paramètre fixé par l'utilisateur) :

$$\forall s \in [0,1], \quad \begin{cases} x^{n+1}(s) = x^n(s) - \gamma\left[F_x(x^n(s), y^n(s)) - \alpha\,(x^n)^{(2)}(s) + \beta\,(x^n)^{(4)}(s)\right] \\ y^{n+1}(s) = y^n(s) - \gamma\left[F_y(x^n(s), y^n(s)) - \alpha\,(y^n)^{(2)}(s) + \beta\,(y^n)^{(4)}(s)\right] \end{cases} \tag{5}$$

Pour implémenter cette itération, on utilise la contrainte $P(1) = P(0)$ pour imposer à la courbe d'être fermée, et les schémas de différences finies suivants ($h \ll 1$) :

$$\begin{cases} (P^n)^{(2)}(s) = P^n(s+h) - 2\,P^n(s) + P^n(s-h) \\ (P^n)^{(4)}(s) = P^n(s+2\,h) - 4\,P^n(s+h) + 6\,P^n(s) - 4\,P^n(s-h) - P^n(s-2\,h) \end{cases} \tag{6}$$

En notant $\mathbf{x} = [x(0), x(h), \ldots, x(1-h)]^\top$ et $\mathbf{y} = [y(0), y(h), \ldots, y(1-h)]^\top$ la discrétisation de la courbe, et en remplaçant les opérateurs de dérivation par leurs approximations discrètes (6), l'itération (5) se réécrit :

$$\begin{cases} \mathbf{x}^{n+1} = (I + \gamma A)\,\mathbf{x}^n + B_x(\mathbf{x}^n, \mathbf{y}^n) \\ \mathbf{y}^{n+1} = (I + \gamma A)\,\mathbf{y}^n + B_y(\mathbf{x}^n, \mathbf{y}^n) \end{cases} \tag{7}$$

où I est la matrice identité (attention à ne pas confondre cette matrice avec la matrice des niveaux de gris !). Dans (7), la matrice A s'écrit :

$$A = \alpha D_2 - \beta {D_2}^\top D_2 \tag{8}$$

où D_2 est une matrice de différences finies qui constitue une approximation de la dérivée seconde :

$$D_2 = \begin{bmatrix} -2 & 1 & & & & 1 \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ 1 & & & & 1 & -2 \end{bmatrix} \tag{9}$$

(notez le cas particulier des deux premières et des deux dernières lignes), ${D_2}^\top D_2$ représente l'opérateur de dérivation à l'ordre 4, et $B_x(\mathbf{x}^n, \mathbf{y}^n)$ et $B_y(\mathbf{x}^n, \mathbf{y}^n)$ sont définis par :

$$\begin{cases} B_x(\mathbf{x}^n, \mathbf{y}^n) = -\gamma \begin{bmatrix} F_x(x^n(0), y^n(0)) \\ \vdots \\ F_x(x^n(1-h), y^n(1-h)) \end{bmatrix} \\ \\ B_y(\mathbf{x}^n, \mathbf{y}^n) = -\gamma \begin{bmatrix} F_y(x^n(0), y^n(0)) \\ \vdots \\ F_y(x^n(1-h), y^n(1-h)) \end{bmatrix} \end{cases} \tag{10}$$

Pour implémenter un *snake*, il faut donc se donner une forme initiale $\mathbf{P}^0 = [\mathbf{x}^0, \mathbf{y}^0]$, puis effectuer les mises à jour (7) jusqu'à convergence de l'algorithme.

Exercice 1 : étude de l'énergie externe

Étudiez dans un premier temps l'énergie externe définie par l'équation (2). Complétez le script `exercice_1.m` de manière à afficher cette énergie, ainsi que le champ de forces $\nabla E_{\text{ext}} = [F_x, F_y]^\top$ associé. Que peut-on dire de ce champ loin des contours présents dans l'image ? Quelle conséquence cela peut-il avoir sur l'initialisation du *snake* ?

Afin d'« attirer » davantage le *snake* vers les contours de l'image, il est d'usage d'appliquer un filtre de convolution gaussien à l'image avant d'en calculer le gradient, i.e. de remplacer l'énergie externe (2) par :

$$E_{\text{ext}}(P(s)) = -\left\|\nabla \left(G_{\sigma,T} * \mathcal{I}\right)(P(s))\right\|^2 \tag{11}$$

où $G_{\sigma,T}$ est un noyau gaussien de taille $T \times T$ et d'écart-type σ.

En utilisant les fonctions `fspecial` et `conv2` (avec l'option `'same'`), calculez et affichez la nouvelle énergie externe et le champ de forces externes qui en dérive. En faisant varier les paramètres du noyau gaussien, essayez de prédire le comportement d'un *snake* selon la façon dont il est initialisé. Que pensez-vous enfin de ce champ de forces lorsque l'on s'approche des bords externes de l'image ? Comment pourrait-on corriger cela ?

Exercice 2 : implémentation du *snake*

En utilisant l'énergie externe (11), complétez le script `exercice_2.m` de façon à implémenter le *snake* tel qu'il est décrit ci-dessus. Il faut, dans l'ordre :

1. Calculer l'énergie externe (cf. exercice 1).
2. Définir un *snake* initial (pour simplifier, il vous est proposé de cliquer les points de contrôle du polygone initial).
3. Définir A en utilisant l'équation (9) (utilisez la fonction `spdiags` de Matlab plutôt que des boucles `for` !).
4. Réaliser l'itération (7).

Le jeu de paramètres fourni devrait permettre d'obtenir des résultats raisonnables sur l'image `coins.png`. Essayez de visualiser l'influence de chacun des paramètres sur la segmentation, et de travailler sur d'autres images (la commande `fileparts(which('cameraman.tif'))` permet de localiser les images internes à Matlab). Essayez également de « détourer » la tumeur sur l'image `IRM.png`.

Testez ensuite votre script avec différentes initialisations (à l'extérieur d'un objet, à l'intérieur, autour, ...). Que pensez-vous de l'énergie externe utilisée, notamment dans les zones homogènes ?

Il semble que le *snake* converge chaque fois « à l'intérieur » de l'objet à détourer. Pourquoi ? Une solution pour éviter ce phénomène est de faire « diffuser » l'énergie vers les contours de l'image (cf. exercice 3). Voyez-vous d'autres parades ?

Exercice 3 (facultatif) : diffusion vers les contours

Afin d'améliorer la robustesse à l'initialisation et aux concavités, il est nécessaire de changer le champ de forces externes $[F_x, F_y]^\top$ (qui attire le *snake* vers les contours de l'image). Un bon modèle de forces externes est celui de la diffusion vers les contours (*gradient vector flow*, cf. [Xu et Prince, 1998]), défini comme l'argument du minimum (argmin) de la fonctionnelle suivante :

$$\mathcal{E}(F_x, F_y) = \iint_{x,y} \left\{ \|\nabla E_{\text{ext}}(x,y)\|^2 \, \|[F_x, F_y]^\top(x,y) - \nabla E_{\text{ext}}(x,y)\|^2 + \mu_{\text{GVF}} \left[\|\nabla F_x(x,y)\|^2 + \|\nabla F_y(x,y)\|^2 \right] \right\} dxdy \tag{12}$$

où $\nabla E_{\text{ext}} = [\partial_x E_{\text{ext}}, \partial_y E_{\text{ext}}]^\top$ désigne le champ de forces externes de base (sans filtrage gaussien).

D'après le calcul des variations, l'équation d'Euler-Lagrange associée s'écrit, pour (x, y) quelconque :

$$\begin{cases} \|\nabla E_{\text{ext}}(x,y)\|^2 \, [F_x(x,y) - \partial_x E_{\text{ext}}(x,y)] - \mu_{\text{GVF}} \, \Delta F_x(x,y) = 0 \\ \|\nabla E_{\text{ext}}(x,y)\|^2 \, [F_y(x,y) - \partial_y E_{\text{ext}}(x,y)] - \mu_{\text{GVF}} \, \Delta F_y(x,y) = 0 \end{cases} \tag{13}$$

où Δ désigne l'opérateur laplacien.

Pour résoudre ces deux équations, connues sous le nom d'*équations de diffusion généralisées*, on peut utiliser l'itération suivante :

$$\begin{cases} F_x^{n+1}(x,y) = F_x^n(x,y) - \gamma_{\text{GVF}} \left\{ \|\nabla E_{\text{ext}}(x,y)\|^2 \, [F_x^n(x,y) - \partial_x E_{\text{ext}}(x,y)] - \mu_{\text{GVF}} \, \Delta F_x^n(x,y) \right\} \\ F_y^{n+1}(x,y) = F_y^n(x,y) - \gamma_{\text{GVF}} \left\{ \|\nabla E_{\text{ext}}(x,y)\|^2 \, \left[F_y^n(x,y) - \partial_y E_{\text{ext}}(x,y)\right] - \mu_{\text{GVF}} \, \Delta F_y^n(x,y) \right\} \end{cases} \tag{14}$$

Complétez le script `exercice_3.m` de façon à calculer le nouveau champ de forces externes (utilisez la fonction `del2` pour calculer le laplacien). Que pensez-vous de ce nouveau champ de forces ? Quelle conséquence cela va-t-il avoir sur le comportement du *snake* ? Observez également le comportement du champ près des bords de l'image. Que pourrait-on faire pour améliorer les résultats sur les bords de l'image ?

Effectuez une copie du script `exercice_2.m`, de nom `exercice_3_bis.m`, de manière à implémenter le *snake* avec le nouveau champ de forces externes. Essayez de mettre en évidence différents comportements du *snake*. Quelles autres améliorations vous sembleraient-elles pertinentes ?