TP Contours déformables: Snakes
CPE

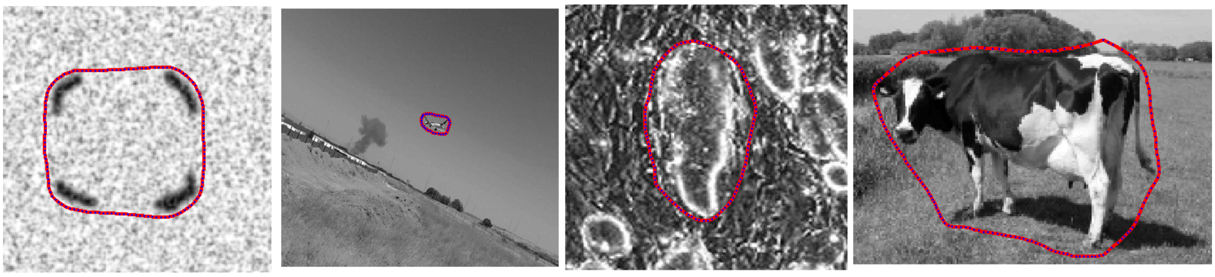

FIGURE 1 – Figures d'exemples

1 But du TP

Ce TP propose de mettre en oeuvre la méthode des contours déformables (ou Snakes) afin de réaliser la segmentation d'un image.

2 Rappel

On rappelle que les Snakes consistent à trouver une courbe *optimale* γ tel que γ minimise une énergie E.
On peut considérer dans le cas classique que l'énergie E est donnée par :

$$E(\gamma) = \int \lambda_1 \|\gamma'\|^2 + \lambda_2 \|\gamma''\|^2 - \lambda_3 \|\nabla I\|^2(\gamma) \tag{1}$$

On peut alors montrer, qu'une courbe discrète dénotée par $v = (v_x, v_y)$, avec

$$\left\{ \begin{array}{l} v_x = (v_x[0], v_x[1], v_x[2], \dots) \\ v_y = (v_y[0], v_y[1], v_y[2], \dots) \end{array} \right. ,$$

converge vers un minimum de cette énergie si on applique la relation suivante de manière itérative :

$$v_x^{k+1} = \mathrm{A}^{-1} \left(v_x^k + \Delta t \, \lambda_3 \, P_x^k \right) , \tag{2}$$

avec

$$\left\{ \begin{array}{l} \mathrm{A} = \mathrm{Id} + 2\Delta t(-\lambda_1 \, \mathrm{D}_2 + \lambda_2 \, \mathrm{D}_4) \\ \\ P_x^k = -\nabla_x(\|\nabla I\|^2)(v_x^k, v_y^k) \, . \end{array} \right.$$

Et de manière identique pour la composante y.

3 Initialisation

Dans un premier temps, on vient initialiser la courbe à déformer.

Question 1 *Chargez une image d'exemple simple et affichez la.*

On pourra se servir des appels suivants :

```
% lecture d'un fichier image
I=imread('fichier');

% conversion de valeurs en double
I2=double(I);

%affichage d'une image avec normalisation du niveau de gris
imagesc(I);

%modification de la table de couleur en noir et blanc
colormap gray;

%effacer l'image courante
clf;
```

Question 2 *Créez un cercle autour de l'objet à déformer. Affichez celui-ci.*
On prendra soin de centrer correctement ce cercle et d'éviter de sortir des frontières de l'image.

Pour afficher le cercle par dessus l'image, on pourra s'inspirer de l'approche suivante :

```
%efface l'image preexistante
clf;

%affiche une image
imagesc(I);

%genere une courbe
t=[0:N-1]/N;t=t';
vx=R*cos(2*pi*t)+Cx;
vy=R*sin(2*pi*t)+Cy;

%permet l'affichage en sur-impression
hold on;

%affiche le cercle en rouge
plot([vx;vx(1)],[vy;vy(1)],'r');
```

4 Minimisation de la norme des dérivées

Dans un premier temps, on s'interesse à la matrice A de l'eq.(2).

Question 3 *Construisez les matrices de dérivées 2 (D_2) et 4 (D_4). Vérifiez celles-ci en l'affichant dans le cas où le nombre de points est faible.*

Question 4 *Calculez et affichez $\mathrm{D}_2\, v_x$, $\mathrm{D}_2\, v_y$ et $\mathrm{D}_4\, v_x$, $\mathrm{D}_4\, v_y$ (on pourra cette fois prendre une courbe échantillonnée sur 50 à 100 points). Est-ce que ce résultat était prévisible ?*

Dans la suite, on pourra considérer $\Delta t = 0.05$, $\lambda_1 = 5$, et $\lambda_2 = 5$.

Question 5 *Précalculez la matrice A ainsi que* A^{-1} *une fois. Notez que celle-ci ne dépendent pas des coordonnées de* v*, mais seulement de la dimension de la courbe.*

Question 6 *Dans une boucle, appliquez l'itération suivante et visualisez le résultat de la déformation de la courbe. Est-ce bien le résultat attendu ?*

$$\begin{cases} v_x^{k+1} = \mathrm{A}^{-1}\ v_x^k \\ v_y^{k+1} = \mathrm{A}^{-1}\ v_y^k \end{cases}$$

Pour l'affichage, on pourra s'inspirer du procédé suivant :

```
for k=[1:500]

  %iteration
  vx=inverse_A*vx;
  vy=inverse_A*vy;

  %affichage une fois sur 2
  if(mod(k,2)==0)

    %supprime l'image et en affiche une nouvelle
    clf
    imagesc(I);
    hold on
    plot([vx;vx(1)],[vy;vy(1)]);

    %force reactualisation de l'image
    pause(0.01);
  end

end
```

Question 7 *Visualisez l'évolution de l'énergie interne de la courbe.*

5 Force totale

On ajoute désormais la force externe empêchant la courbe de traverser les bords de l'image.

Question 8 *Calculez et visualiser en tant qu'image* $\|\nabla I\|^2$*.*

Question 9 *Calculez* $[Kx, Ky] = -\nabla(\|\nabla I\|^2)$*. Notez que ce gradient contient deux composantes, et que chacune d'elle peut être vue en tant qu'image.*

Question 10 *Dans la boucle, calculez les termes* P_x *et* P_y *(voir eq. 2) en tant qu'échantillonnée de* $-\nabla(\|\nabla I\|^2)$ *aux positions* (v_x, v_y)*.*

On pourra s'aider des fonctions d'interpolations de Matlab suivant la syntaxe suivante :

```
%interpolation lineaire calculant P=K(vx,vy);
Px=interp2(Kx,vx,vy);
Py=interp2(Ky,vx,vy);
```

Question 11 *Appliquez désormais l'iteration complète de l'eq. (2). Commentez.*

On pourra considérer $\lambda_3 = 20$ dans ce cas.

Question 12 *Visualisez l'évolution de l'énergie interne et externe.*

6 Analyse supplémentaire

Question 13 *Appliquez votre algorithme sur une image bruitée. Comment devez vous faire évoluer les paramètres* λ *?*

Pour vous aider, vous pourrez également calculer le terme P sur une version lissée de l'image (utilisation d'une convolution par un noyau Gaussien avant calcul du gradient).

Question 14 *Réalisez des tests sur différents types d'images à segmenter, et commentez les résultats obtenues.*

Question 15 (Supplément) *Implémentez une extension (ballon, GVF, ...) et commentez l'amélioration obtenue.*

7 Aide Matlab supplémentaire

7.1 Autres fonctions potentiellement utiles

— *gradient* : Calcule le gradient d'une image.
— *conv2* : Réalise une convolution 2D.
— *diag* : Construit une matrice diagonale (options pour choisir un offset sur la position).
— *ginput* : Permet la saisie manuelle des coordonnées d'une position à la souris.