Active Contours

Ivano Calabrese Nicola Carlon

04 Marzo 2010

Sommario **- Presentazione ed implementazione MATLab di un modello di contorno attivo deformabile, applicato alla segmentazione di depth maps.**

1 Introduzione

Gli *snakes*[1] o *contorni attivi* sono un modello deformabile multidimensionale utili per segmentare immagini con efficienza ed accuratezza.
I modelli deformabili si basano sui concetti di geometria, fisica e di teoria dell'ottimizzazione e sono in generale robusti al rumore e ai contorni disconnessi.
In particolare gli snakes sono spline a energia minima ossia curve polinomiali che, da un punto iniziale, vengono deformate da forze interne ed esterne in modo da adattarsi iterativamente al contorno della regione da segmentare. La natura dinamica del metodo li rende utili anche nella segmentazione di video e di immagini 3D.
Questo particolare modello basato sulla teoria dell'elasticità permette di inserire eventuale conoscenza apriori dell'oggetto da segmentare, definendo opportunamente il coefficiente di elasticità dello snake e la funzione di energia potenziale esterna che attrae lo snake verso le caratteristiche della regione da fittare.

2 Formulazione matematica

Lo snake è una curva parametrica nel piano immagine che nella sua rappresentazione discreta è definita da una serie di *corner points* connessi tramite relazioni di vicinanza di pixel.

$$s(p) = (x(p), y(p)) \tag{1}$$

dove $p \in [0,1]$ è la lunghezza normalizzata dell'arco lungo il contorno.
Alla curva è associato un funzionale:

$$E^*_{snake} = \int_0^1 E_{interna}(s(p)) + E_{esterna}(s(p))dp$$

L'energia interna regola la forza parallela alla lunghezza dello snake (tensione) e quella perpendicolare (rigidità):

$$E_{interna}(s) = \frac{1}{2}[\alpha \left|s'\right|^2 + \beta \left|s''\right|^2] \tag{2}$$

Più grande è il parametro α e più lo snake potrà allungarsi facilmente; più piccolo è il parametro β più lo snake sarà flessibile e si adatterà maggiormente alle features dell'immagine.

L'energia esterna è definita in modo che le features dell'immagine attirino ad esse lo snake.
Ad esempio se si ha una $E_{esterna} = -\gamma|\triangle I(x,y)|^2$ lo snake verrà attirato da contorni con alto valore di gradiente, cioè dove è sicuramente presente un bordo.
In questa implementazione viene considerata la radice quadrata della distanza euclidea dai punti di bordo:

$$E_{xy} = \sqrt{d_{b,i}} \tag{3}$$

con $d_{b,i} = \sqrt{(x_b - x_i)^2 + (y_b - y_i)^2}$ dove x_i, y_i sono le coordinate del punto considerato e x_b, y_b le coordinate del bordo più vicino.
In questo modo si forma una mappa grande come tutta l'immagine dove i punti di minimo saranno proprio i bordi mentre nel resto dell'immagine l'energia aumenterà con la radice quadrata della distanza come una sorta di forza gravitazionale che regola la velocità dello snake vicino al bordo. L'oggetto viene segmentato quando lo snake raggiunge un minimo locale di energia, quindi il problema si traduce nella minimizzazione del funzionale $E^*_{snake}(u)$
Per farlo si utilizza l'equazione di Eulero-Lagrange.

$$\alpha s'' - \beta s'''' - \bigtriangledown E_{ext}(s(p)) = 0 \tag{4}$$

Dato che si cerca lo snake che minimizza la (4) si considera lo snake come una funzione nel tempo. Si vuole quindi che la sua derivata sia zero al raggiungimento di tale minimo:

$$\frac{\partial s}{\partial t} = \alpha s'' - \beta s'''' - \bigtriangledown E_{ext}(s(p)) \tag{5}$$

Seguendo l'algoritmo di Kass [1] con il metodo alle differenze finite (DFM) si può costruire la matrice delle forze interne A:

$$\begin{bmatrix} -2\alpha-6\beta & \alpha+4\beta & -\beta & 0 & 0 & \cdots & -\beta & \alpha+4\beta \\ \alpha+4\beta & -2\alpha-6\beta & \alpha+4\beta & -\beta & 0 & \cdots & 0 & -\beta \\ -\beta & \alpha+4\beta & -2\alpha-6\beta & \alpha+4\beta & -\beta & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -\beta & 0 & 0 & 0 & 0 & \cdots & -2\alpha-6\beta & \alpha+4\beta \\ \alpha+4\beta & -\beta & 0 & 0 & 0 & \cdots & \alpha+4\beta & -2\alpha-6\beta \end{bmatrix}$$

Scrivendo la (5) in notazione matriciale si ottiene la legge di variazione di energia dello snake:

$$\begin{aligned} \frac{\partial X}{\partial t} &= A \cdot X + f_x(X,Y) \\ \frac{\partial Y}{\partial t} &= A \cdot Y + f_y(X,Y) \end{aligned}$$

Dove f_x e f_y sono le componenti del gradiente dell'energia esterna mentre X e Y sono le due componenti dello snake.
A questo punto è necessario discretizzare il tempo con

passo γ assumendo però che il passo sia abbastanza piccolo in modo che la forza esterna non cambi molto da un passo al suo successivo.

$$X_t = (I - \gamma A)^{-1} \cdot (X_{t-1} + \gamma f_x(X_t, Y_t)) \quad (6)$$
$$Y_t = (I - \gamma A)^{-1} \cdot (Y_{t-1} + \gamma f_y(X_t, Y_t)) \quad (7)$$

Questa approssimazione permette una risoluzione semplice del sistema ma rende instabili soluzioni con γ elevati.

3 Implementazione

Lo snake viene inizializzato dall'utente con una serie di click sull'immagine in modo da formare una spezzata che viene chiusa successivamente:

```
xyPoly=[xyPoly;[xmouse,ymouse]];
...
xyPoly=[xyPoly;xyPoly(1,:)];
```

Questa curva viene interpolata in modo da permettere il calcolo dello spostamento dovuto alle forze anche lungo i suoi segmenti e non solo sui punti definiti dall'utente. Il sovracampionamento dei segmenti dello snake deve seguire la struttura quantizzata dell'immagine in modo che il calcolo delle forze elastiche sia coerente.

```
img = poly2mask(xyPoly(:,1),xyPoly(:,2),h,w);
boundary=bwboundaries(img);
```

Per realizzare la funzione di energia esterna esposta in (3) è necessario estrarre i bordi dell'immagine.
Nel caso di *depth map* si chiede all'utente di cliccare sul punto più chiaro (più vicino) e su quello più scuro (più lontano) dell'oggetto da segmentare. Si procede quindi sogliando l'immagine in modo da considerare solo oggetti compresi in quel range di distanze.
Sucessivamente si esegue un adjustment dell'immagine e si estraggono i bordi con il metodo di Canny per ottenere un'immagine binaria:

```
I=im2double(imadjust(I));
padded=padarray(I,[h w],'replicate');
T=graythresh(padded);
BW = edge(padded,'canny',T);
BW=BW(h+1:2*h,w+1:2*w);
```

Si è scelto di usare questo metodo sul padding dell'immagine per evitare l'effetto dei bordi dell'immagine.
La funzione di energia esterna coincide con la radice quadrata della distanza euclidea di ogni punto dell'immagine dal punto di bordo più vicino.

```
distancemap=sqrt(bwdist(BW));
...
smoothed=imfilter(I, fspecial('gaussian', [3 3], 1),
                  'replicate');
gradiente=distancemap+distancemap.*smoothed;
```

L'algoritmo di Canny elimina le differenze tra bordi di regioni diverse e per ripristinarle è necessario sommare l'immagine originale al risultato solo dove sono presenti bordi.
Il calcolo esplicito delle componenti della forza esterna lungo i due assi X e Y viene effettuato calcolando il gradiente della funzione di energia appena definita:

```
[FX,FY] = gradient(gradiente/max(max(gradiente)));
```

Successivamente vengono normalizzati i coefficienti di tensione e rigidità introdotti dall'utente, in modo da evitare degenerazioni dello snake a causa di una forza elastica maggiore della forza esterna. Si considera anche il fatto che una forza esterna debole induce un rallentamento nella convergenza e una relativa perdominanza delle forze interne.

```
gamma = (100*delta/resolution)/max(max(FX));
alpha=alpha/(100*delta)*max(max(FX));
beta=beta/(100*delta)*max(max(FX));
```

A questo punto si può procedere con il calcolo vero e proprio della deformazione dove viene calcolata la forza esterna che agisce sullo snake, utilizzata per il calcolo definito in (6):

```
fex= interpn(double(FX),double(y),double(x),'linear');
fey= interpn(double(FY),double(y),double(x),'linear');
...
x = P*(x+gamma*fex);
y = P*(y+gamma*fey);
```

con $P = (I - \gamma A)^{-1}$.
Per implementare il criterio di fermata dell'algoritmo e al tempo stesso raggiungere una minimizzazione adeguata della (4) si è scelto di verificare la distanza tra i punti dei due snake di passi temporali consecutivi.
A causa della natura discreta degli snake la distanza corretta è definita dalla distanza di Hausdroff che viene però realizzata con un algoritmo particolarmente oneroso. Per questo si è scelto di considerare la distanza euclidea tra i punti delle due curve. L'algoritmo si comporta bene se si considera anche la distanza massima e la si impone, insieme alla distanza media, minore di una soglia calcolata sperimentalmente.
Lo snake minimo risultante viene poi utilizzato per la creazione della maskera ROI che viene moltiplicata per l'immagine originale:

```
mask = poly2mask(xf,yf,size(I,1),size(I,2));
imshow(im2double(I).*mask);
```

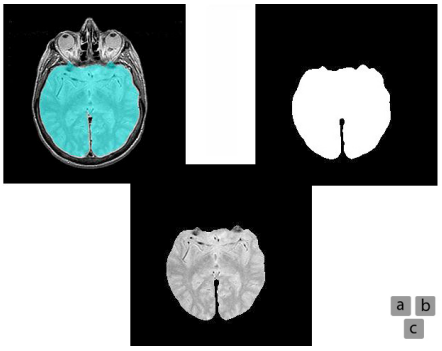

Figura 3.1: a)Risultato della minimizzazione b)creazione ROI c)segmentazione.

4 Risultati

I test sono stati eseguiti su un set di 7 immagini diverse:

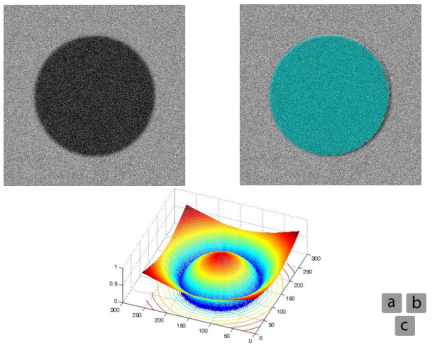

Figura 4.1: a) Cerchio con rumore gaussiano ($\sigma = 20$) b) segmentazione c) energia esterna

Come si vede nonostante il rumore utilizzando un coefficiente di rigidità elevato si ottengono buoni risultati.

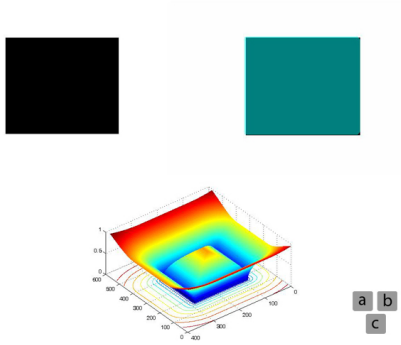

Figura 4.2: a) Rettangolo nero su sfondo bianco b) segmentazione c) energia esterna

In questo caso se la tensione è troppo elevata gli spigoli vengono smussati, per contro la segmentazione avviene più lentamente perchè è necessaria una maggiore risoluzione.

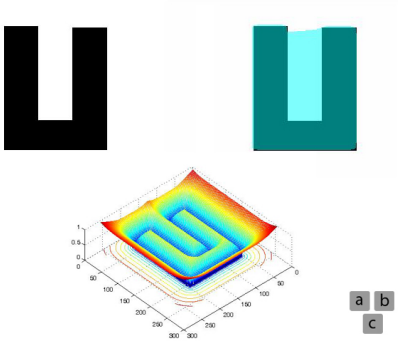

Figura 4.3: Figura a 'U' b) segmentazione c) energia esterna

Il fallimento di questa segmentazione è dovuto alla definizione di energia basata sulla distanza dai punti di bordo che, in questo caso, risultano essere paralleli. Per risolvere questo problema è necessario inizializzare lo snake in modo che segua la concavità oppure modificare il modello utilizzando per esempio il Gradient Vector Flow [2].

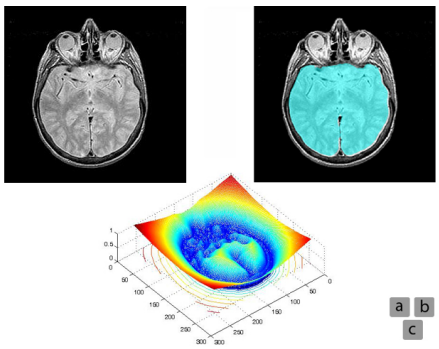

Figura 4.4: a) Risonanza magnetica b) segmentazione c) energia esterna

In questo caso l'inizializzazione è più critica a causa di zone ristrette di bassa energia dovute della complessità dell'immagine.

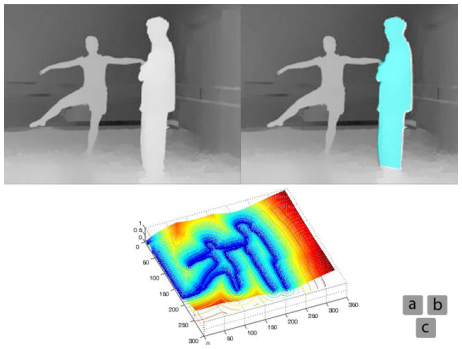

Figura 4.5: a) Depth map b) segmentazione c) energia esterna

La segmentazione risulta essere efficiente grazie alla correzione dei bordi presentata nella Sezione 3, infatti si può notare come nella segmentazione della figura in primo piano non risulta esserci parte della figura in secondo piano nonostante l'intersezione dei loro bordi.

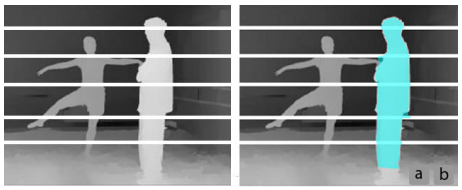

Figura 4.6: a) Depth map con occlusioni b) segmentazione

L'algoritmo è abbastanza robusto alle occlusioni se si mantengono elevate le forze interne.

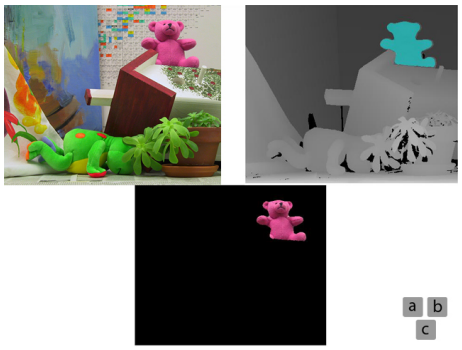

Figura 4.7: a) Immagine originale b) segmentazione della depth map con selezione manuale della soglia c) risultato

Utilizzando la sogliatura manuale sulla distanza nella depth map è possibile segmentare oggetti a distanze arbitrarie anche se a basso contrasto e senza che gli altri interferiscano a casua di bordi con energie più elevate.

5 Conclusioni

Trattandosi di un modello di analisi locale i risultati dipendono dall'inizializzazione dello snake che deve essere effettuata da un operatore umano.

Riferimenti bibliografici

[1] *'Snakes: Active contour models'*, Michael Kass, Andrew Witkin and Demetri Terzopoulos, Schlumberger Palo Alto Research, 1987.

[2] *'Snakes, Shapes, and Gradient Vector Flow'*,Chenyang Xu, Jerry L. Prince, IEEE.