Medical Image Analysis

Final Report
CAP 6516 - Spring 2009

Level-set method for tracking Objects using density matching

Tao Zhang and Daniel Freedman

Prepared by
S M Shahed Nejhum
(8589-1199)
Introduction:

Tao Zhang and Daniel Freedman propose a new level set based tracking algorithm which finds the target using foreground matching and background mismatching. Foreground density matching based trackers only uses estimated foreground density. These types of trackers often get stuck into local optima and do not produce good tracking result. Not only this, a good initialization close to target contour is also required. Tao Zhang showed how background density mismatching can be used to overcome these problems. They showed when tracker uses mismatching of background density in the energy function; it gives more robust, improved result.

They also showed how the tracker can be formulated as PDE-based curve evolution and then level set method can be applied to solve the PDE. In their other work, they showed it is also possible to include shape prior in the curve evolution equation.

Formulation of the Tracker

In this paper, two different types of density measurement have been used.

- Kullback-Leibler divergence, $K(\omega)$,

\[
K(\omega) = \int q(x) \log \left( \frac{q(x)}{p(x; \omega)} \right) dx
\]

- Bhattacharyya measure, $B(\omega)$

\[
B(\omega) = \int \sqrt{q(x) p(x; \omega)} dx
\]
Here, \( q(x) \) is the model density learned prior to running the tracker, \( \omega \) is current estimate of the target region and \( p(x, \omega) \) is current foreground density. Note that background density estimate can be represented by similar notation, \( p(x; \omega^c) \).

Energy function of the tracker, using KL divergence measure, is defined as –

\[
E(\omega) = \lambda_1 K(\omega) - \lambda_2 K(\omega^c) \\
= \int \lambda_1 q(x) \log \left( \frac{q(x)}{p(z; \omega)} \right) dx - \int \lambda_2 q(x) \log \left( \frac{q(x)}{p(z; \omega^c)} \right) dx
\]

Similarly, if we use Bhattacharyya measure,

\[
E(\omega) = -\lambda_1 B(\omega) + \lambda_2 B(\omega^c)  \\
= -\int \lambda_1 \sqrt{q(x) p(x; \omega)} dx + \int \lambda_2 \sqrt{q(x) p(x; \omega^c)} dx
\]

First term in the energy function is the foreground density matching term and the other one is background density mismatching term.

If we let, \( c = \partial \omega \), then we can write the curve evolution equations.

\[
\frac{\partial c}{\partial t} = \left[ \lambda_1 \frac{q(Z(c)) - p(Z(c); \omega)}{N(Z(c); \omega)} + \lambda_2 \frac{q(Z(c)) - p(Z(c); \Omega \setminus \omega)}{N(Z(c); \Omega \setminus \omega)} \right] n
\]

\[
\frac{\partial c}{\partial t} = \left[ \frac{\lambda_1}{2A(\omega)} \left( \sqrt{\frac{q(Z(c))}{p(Z(c); \omega)}} - B(\omega) \right) + \frac{\lambda_2}{2A(\Omega \setminus \omega)} \left( \sqrt{\frac{q(Z(c))}{p(Z(c); \Omega \setminus \omega)}} - B(\Omega \setminus \omega) \right) \right] n
\]

In general, we can write,

\[
\frac{\partial c}{\partial t} = F(c) n,
\]

Where, \( F(c) \) can be thought as image based velocity.
Implementation Details

These curve flow equations are implemented using level set methods. I have implemented the tracker in MATLAB. A graphical user interface is developed for initialization of the tracker. Figure 1 shows the GUI for this project. Using this GUI interface user can load an image sequence, outline the initial contour and load model distribution function. Also, several important parameters of the tracker can be set using the same interface. After initialization, user should click **Track** button at the bottom of GUI. This invokes the tracking sub-routine. Tracking result is updated every 5-10 iteration and displayed in GUI.

![GUI for the project](image)
Results

I have tested the tracker using several different image sequences. Some of the image sequences are synthetic and some of them are real image sequence downloaded from web. For each sequences I run the program several times with different initialization and parameter setting. This helped me to find the suitable values of the parameters.

First I present a tracking result on a very simple synthetic image sequence where an irregular shape is going mainly through translational motion. Tracking result in Figure 2 shows that the tracker is able to find the contour of the target.

Figure 2: Tracking result on synthetic image sequence. From left to right, top row shows Frame 2, 5 and 9 of the sequence. Bottom row shows frame 11, 13 and 14 of the sequence.
Next we show a tracking result on tracking a tennis ball. This sequence has been downloaded from internet. The tracker is able to track the ball reasonably well. At every frame it gave almost correct boundary (figure 3).

Figure 3: Tennis ball tracking result.

Next, we present hand tracking result of a table tennis player. Result shows that it is locking on the hand of the target but contour is not smooth. Result is given in figure 4.

Figure 4: Tracking hand of a table tennis player.
Tracking result on Echo Ultra Sound data is shown in figure 5.

Figure 5: Echo Ultra Sound image sequence. Top row: Left: initialization, Right: final result. Bottom row: tracking result on successive frames.

Tracking result on heart image sequence.
As a general comment, not all the images in the image sequence, implemented tracker produced good results. There were some frames, where the tracker gave very zigzag boundary. I believe by playing more with tunable parameters improved tracking result can be achieved.

Reference
