Motion-based Post Processing of Deformable Contours *

Min Li
CIS Department, University of Delaware
mli@cis.udel.edu

Chandra Kambhamettu
CIS Department, University of Delaware
chandra@cis.udel.edu

Abstract

We present a framework for tracking contours in 2D image sequences by taking into account the motion information and correspondence between contour points. With this framework, The tracking problem can be divided into two steps: i) Contour detection using general internal and external energies in separate frames, and ii) Tracking and refining contours using motion information of the entire image sequence. In addition to the traditional internal and external energies, we introduce two new constraints in the temporal domain which we call temporal internal energy and temporal external energy in the post processing stage. The temporal internal energy smoothes the contours by minimizing the velocity difference between neighboring contour points and the temporal external energy pushes the contour points toward their best location according to their corresponding points in the preceding image. We show the performance of the new constraints with synthetic image sequences and also present the post processing results for medical image sequences.

1. Introduction

Snake is an energy minimizing model which is popularly used for automatic extraction and tracking of image contours. As an active curve, snake moves under the control of image forces and the curve properties. The image forces, usually related to the image gradient, push the snake toward the boundaries in images. The curve properties influence the shape of snake and usually the goal is to get smooth curves.

Since the snake model was introduced by Kass [10], many works have been done to improve the performance of this model. Cohen [8] [7] proposed the balloon model, Gunn [9] introduced the dual active contour. Both these models make the snake insensitive to the initial guess and prevent it from stopping at local minima. Wang [15] introduced the B-Spline representation of snake which is a multistage active contour model. Leymarie [12] used the extracted contour of the preceding frame as the initial guess for the next frame to track contour through a sequence of images. In [2][3], in addition to the extracted contour of preceding frame, optic flow is also used to find the candidate contour points for the adjacent frame.

In [12] [2][3], tracking in every frame is a separate 2D problem except for the propagation of the result from the preceding frame to the next frame. The advantage of this scheme is that the selection of the initial position is quite simple. The selection is only necessary for the first frame. But if one frame contains missing boundaries, the error will also propagate to the sequence. To address this problem, multiple snakes in different frames are necessary. With 2D contour model applied to each frame, temporal information from neighboring frames can be used to achieve robust results. Akgul [4] constrains the smoothness between snakes in different frames and Chalana [6] also uses the temporal smoothness in addition to the constraint of distance between snakes in successive frames. However, the temporal smoothness is applied to points at different frames. These points should correspond to each other but the correspondence can not be guaranteed without additional constraints. Chalana's distance constraint also needs the assumption of monotonic motion.

In [2][3], the extracted contour in an individual frame is refined according to the contours of neighboring frames with a novel dual snake technique. Contour correspondence is used and the hard constraint is not considered in this post processing. While in our approach, image correspondence, motion information and user feedback will be used to do the post processing.

2 Why Postprocessing?

Due to the inappropriate smoothness between non-corresponding points in different frames and the difficulty of motion estimation, no one model is perfect to track the boundaries through a sequence of images. Generally, one can always find that the results from the snake model are good in some frames but not satisfying in other frames. Correcting the results in every frame by hand is non-trivial and can not fully utilize the information from correctly tracked contours in other frames. Sometimes the user of an active

*Research funding was provided by the National Institutes of Health Grant.

contour system refines the tracked contour in a single frame by hand and expects that the correction be propagated to the whole image sequence. Post processing is necessary in these situations. Post processing should refine the results by correcting the incorrectly tracked contours in some frames according to the correctly tracked contours in other frames, including the corrections given by the user. Our framework deals with the post processing by using information from motion, image correlation constraints and user feedback.

Peterfreund [13] applied the velocity control and contour position prediction with general external and internal energies in a single framework. By introducing the post-processing stage, we divide contour tracking in 2D image sequences into two steps: i) Contour detection using general internal and external energies in separate frames, and ii) Tracking and refining contours using motion information of the entire image sequence. This strategy separates the snake initialization problem and motion-based tracking in two steps, which makes the tracking results more robust. At the post processing step, the requirement of contour position prediction is removed because the motion estimation of contour points can be extracted from the position difference of contour points at different frames. Different with [13], which uses variational approach, dynamic programming is used in our approach. Dynamic programming ensures global optimality of the solution and allows for hard constraints to be enforced on the behavior of the solution [5].

3 Snakes for image sequence and improvements

Active contour, snake, can be defined as a set of points [4][1].

$$M = \begin{bmatrix} m_{11} & m_{12} & ... & m_{1N} \\ m_{21} & m_{22} & ... & m_{2N} \\ ... & ... & ... & ... \\ m_{F1} & m_{F2} & ... & m_{FN} \end{bmatrix} \quad (1)$$

where $(m_{i1}, m_{i2}, m_{i3}, m_{iN})$ is an active contour in frame i and $(m_{1j}, m_{2j}, m_{3j}, ...m_{Fj})$ represents corresponding contour points from frame 1 to frame F for the j^{th} contour point.

Given the contour representation as the above matrix for an image sequence, the internal energy in one frame can be defined as

$$E_{Int}(f) = \sum_i (E_{SpaSmo}(m_{if}) + E_{TemSmo}(m_{if})) \quad (2)$$

where

$$E_{SpaSmo}(m_{if}) = 1 - \frac{\vec{m_{i-1f}m_{if}} . \vec{m_{if}m_{i+1f}}}{|\vec{m_{i-1f}m_{if}}| . |\vec{m_{if}m_{i+1f}}|} \quad (3)$$

$$E_{TemSmo}(m_{if}) = 1 - \frac{\vec{m_{if-1}m_{if}} . \vec{m_{if}m_{if+1}}}{|\vec{m_{if-1}m_{if}}| . |\vec{m_{if}m_{if+1}}|}. \quad (4)$$

The E_{SpaSmo} corresponds to the bending force of the smoothness term in the original snake formulation. The E_{TemSmo} is the bending force which constrains the bending of corresponding contour points in different frames. With this bending force, the motion of the contour points in the temporal domain is smoothed. The tracking problem is not limited to a single 2D image anymore. Multiple active contours in different frames converge to the boundaries using both spatial and temporal energies.

The external energy can be defined as the negative of the gradient of the image intensity and minimization of the external energy will push the active contours toward high spatial image gradient areas which are the positions of the boundaries. In a single frame

$$E_{Ext}(f) = -\sum_i \nabla I(m_{if}). \quad (5)$$

The total external energy of a sequence is just the sum of external energies of all frames. The total energy associated with the snake in one frame is,

$$E_{s1}(f) = \alpha_1 E_{Int}(f) + \beta_1 E_{Ext}(f) \quad (6)$$

where α_1 and β_1 are the weighting parameters.

With the temporal smoothness energy defined in Equation (4), contours in successive frames will affect each other. Minimizing the temporal smoothness energy should give a smoothly moving contour in time and this temporal smoothness should be between the corresponding contour points. However, without an additional constraint, the correspondence cannot be guaranteed and the performance of the constraint in the temporal domain is unsatisfied.

To improve the performance of the constraint in the temporal domain, more information of the images such as the image correspondence should be taken into account. By introducing the temporal external energy, we combine the contour correspondence and the image correspondence in a single framework.

Another constraint we introduced in the temporal domain comes from the motion information. In the general active contour model, neighboring points control each other by minimizing E_{SpaSmo} in Equation(3). With the temporal internal energy that we will talk about in the next section, the position of one contour point is affected not only by the positions of its neighbors but also by the motion of

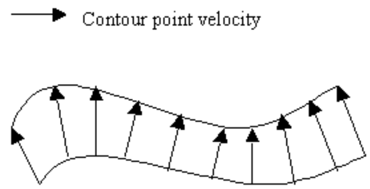

Figure 1: The motion of neighboring contour points is close to each other

its neighbors. Our framework does not need pre-estimated motion since we can get the contour motion from the position difference of corresponding contour points in successive frames.

4 Tracking Framework With Temporal Constraints

4.1. Temporal Internal Energy

When tracking contours from an image sequence, we can approximate the motion of a contour point with the difference between its position and the position of its corresponding point in the successive frame. If two neighboring contour points in the same frame are close to each other, which is necessary in order to not lose the details of the contour, we can assume their motion is very similar. That is, the velocity of the neighboring points along the contour should be close as in Figure 1. This assumption does not require the small motion between successive frames in time. It induces the motion smoothness of neighboring points in space.

For every contour point m_{if}, we can get its estimated velocity $\vec{V_{if}}$ as:

$$\vec{V}_{if} = \vec{X}_{if+1} - \vec{X}_{if} \tag{7}$$

where $\vec{X}_{if+1}$ and $\vec{X}_{if}$ are the positions of points m_{if+1} and m_{if} respectively.

Based on the observation above, we propose the temporal internal energy for one contour point:

$$E_{TemInt}(m_{if}) = \alpha' \left|(\vec{V}_{if} - \vec{V}_{i-1f})\right| + \beta' \left|(\vec{V}_{if} - \vec{V}_{if+1})\right| \tag{8}$$

where α' and β' are the weighting parameters. Minimizing the E_{TemInt} will give the similar motion to neighboring points along the contour. If the contour in one frame is smooth, the contour smoothness will be propagated to the successive frame and the relative positions of the contour points in the successive frame will not change very much. That means with this temporal internal energy, not only the contour smoothness but also the similarity between contours in successive frames is incorporated.

4.2. Temporal External Energy

External energy is the only term that attaches the contour to the image in a snake model. It is usually related to the image gradient. Sometimes it can also be defined as a function of the distance to some edge points. The external energy is usually limited to one single frame as in [4] [6]. Image information from other frames in the sequence does not constrain the shape and location of a single contour. For instance, one frame in the sequence may be blurred and the boundary in it may not have a strong potential to attract contour points. As a result, the active contour of this frame will lock into an incorrect boundary. Thus, we need to have a framework where other frames that will potentially have correct boundaries to influence the incorrect snakes.

Motivated by the temporal internal energy which applies contour information from neighboring frames to the current frame, we propose the temporal external energy. It uses the correlation between corresponding points in successive frames. Even though the image force in one single frame is not strong enough, the corresponding points in successive frames still have a high correlation. Since this constraint comes from the previous frame and is decided by the image information, we call it temporal external energy.

Optic flow is also a good choice to be the external force to drive the snakes in continuous frames [13]. But we are only interested in the motion of contour points and need to consider the hard constraints. Correlation is more suitable and efficient to pick up the best position from the neighborhood for a point, as it is well suited for a dynamic programming implementation.

For point m_{if} which is the corresponding point in frame f of point m_{if-1} in frame f-1, its temporal external energy can be defined as:

$$E_{TemExt}(m_{if}) = 1 - \frac{\sum_{j,k}(A_{jk}-\overline{A}_{if})(B_{jk}-\overline{B}_{if-1})}{\left[\sum_{j,k}(A_{jk}-\overline{A}_{if})^2\right]^{1/2}\left[\sum_{j,k}(B_{jk}-\overline{B}_{if-1})^2\right]^{1/2}} \tag{9}$$

where $\overline{A}_{if}$ and $\overline{B}_{if-1}$ are the intensity mean values of the region centered around the point m_{if} and m_{if-1}. A_{jk} and B_{jk} are the intensities of the elements in these two regions. For some noisy image sequences, we can also use the correlation between their gradient images which are obtained from the original images' Gauss blurred images.

With the temporal external energy, the active contours get more information from images. Contour correspondence and image correspondence are combined together. Minimizing $E_{TemExt}(m_{if})$ will guide the contour point m_{if} move to a better position according to the image information it can get from its corresponding contour point m_{if-1}. Since the correspondence between contour points in successive frames can be guaranteed with this temporal external energy, the temporal smoothness in Equation(4) can affect the contour shape in a more reasonable way.

4.3. Energy Minimization

Combining the temporal internal and external energy together, the temporal energy associated with the snake in one frame is:

$$E_{s2}(f) = \alpha_2 E_{TemInt}(f) + \beta_2 E_{TemExt}(f). \qquad (10)$$

Where

$$E_{TemInt}(f) = \sum_i E_{TemInt}(m_{if}) \qquad (11)$$

$$E_{TemExt}(f) = \sum_i E_{TemExt}(m_{if}). \qquad (12)$$

The energy of the snake in one frame now is:

$$E_{Snake}(f) = \alpha E_{s1}(f) + \beta E_{s2}(f) \qquad (13)$$

where $E_{s1}(f)$ and $E_{s2}(f)$ are defined in Equations (6) and (10) respectively.

Given all the energies definition above, we can now write the optimized total energy of an image sequence as:

$$E_{Total} = min \sum_f E_{Snake}(f). \qquad (14)$$

To reduce the computation complexity, based on the dynamic programming [5], we approximate Equation (14) as:

$$E_{Total} = \sum_f min E_{Snake}(f). \qquad (15)$$

For an image sequence with k frames, assume that there are n contour points $[m_{1f}, m_{2f}, ..., m_{nf}]$ in one frame and the search window size for a point is l. The Equation(15) reduces the computation cost of Equation(14) from $O(l^{nk})$ to $O(k * l^n)$. But the cost is still an exponentially increasing number with respect to the contour point number in one frame. Fortunately, the energy formula Equation(13) can be written in terms of separate energies $E_1, E_2, ..., E_n$ such that the energy term E_i only depends on three contour points $m_{i-1f}, m_{if}, m_{i+1f}$. The dependence of E_i on other contour points in continuous frames has been removed from Equation(15).

$$E_{snake}(f) = \sum_i E_i(m_{i-1f}, m_{if}, m_{i+1f}). \qquad (16)$$

The energy of end points can be approximated as their neighbor points' energy. Based on the above definition, energy minimization in one frame is decomposed to n separated steps. Total computation cost for Equation(15) is reduced to $O(k*n*l^3)$. The details of the decomposition can be found in [11].

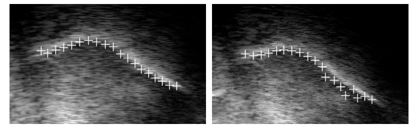

Figure 2: Frames with global affine transformation before post processing

However, the above optimization is only performed in separate frames. The global optimal solution defined by Equation (14) cannot be guaranteed. To overcome this problem, the optimized result of Equation (14) is calculated iteratively over the the whole image sequence: 1) minimize E_{Snake} individually in every frames with dynamic programming; 2) calculate the total energy E_{Total} over the whole image sequence; 3) if E_{Total} is decreased, go back to step 1). Otherwise the optimization process is stopped.

With this iterative minimization algorithm, in addition to the reduction in the computation cost, the temporal constraints are also applied to neighbor frames in an iterative way. The correctly tracked result in one reference frame can be propagated to the whole image sequence.

5 Experiments with medical image sequences

We have applied our algorithm to medical ultrasound image sequences and tested the performance of our framework for post processing. We selected a sequence of ultrasound images of the tongue during speech. Two pairs of images are also created according to a synthetic motion using an ultrasound image as the reference frame.

Ultrasound images are produced by a Head and Transducer Support System(HATS) [14] and are very noisy. In the image producing process, structures such as tendons and blood vessels within the tongue and noise echo artifacts can cause high contrast edges unrelated to the interest in ultrasound images. In some poor images, the tongue contours are even discontinuous. These problems make it difficult to track the contour in the ultrasound image.

5.1 Medical images with synthetic motion

To evaluate the performance of our framework with precise ground truth, we test it with images which are created according to the synthetic motion.

5.1.1 Medical images with synthetic global motion

We first tested the performance of our framework with a pair of images in which a global affine transformation is applied.

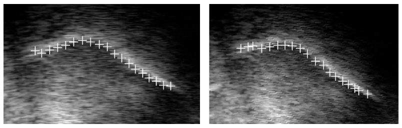

Figure 3: Frames with global affine transformation after post processing

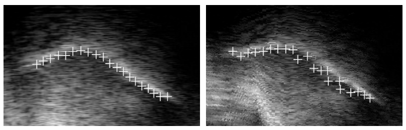

Figure 4: Frames with local affine transformation before post processing

We made the second frame of the pair by applying the same affine transformation to every pixel in the reference frame. The reference image is a frame of an ultrasound image sequence. Since we know the ground truth of the contour in the reference frame, the ground truth of the synthetic frame can be calculated. We intentionally changed some contour points in the synthetic frame to incorrect positions and only used the temporal energy to do the post processing.

This pair of images is showed in Figure 2. The contour of the synthetic frame of this pair has some manually created errors, which average to a 4.83 pixel difference from the ground truth. Figure 3 shows the result after we apply only the temporal constraints to the synthetic frame. The errors now average to only a 2.06 pixel difference from the ground truth.

5.1.2 Medical images with synthetic local motion

we tested another image pair in which a local affine transformation is applied. The second frame of this pair is the result of applying different affine transformations to different regions of the reference frame.

To get reasonable local affine transformations for different regions, we first analyzed the motion of the tongue contour in an ultrasound image sequence for which we know the positions of the contours in all frames. Because the ultrasound transducers of HATS are employed in a fan-like configuration, it is natural to represent every pixel of these 2D ultrasound images with (r, θ) in a polar coordinate system.

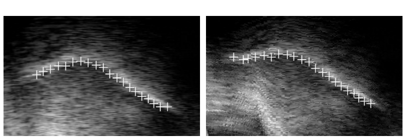

Figure 5: Frames with local affine transformation after post processing

For every ultrasound image frame we are only interested in the part which contains the tongue contour. We divided the interesting part of every ultrasound image frame to 40 regions according to the angle θ in the polar coordinate system. In each region, we selected a contour point as the sample point to be analyzed.

The affine transformation we applied to all regions is:

$$\begin{aligned} R_i^{j+1} &= a_i r_i^j + b_i \theta_i^j \\ \theta_i^{j+1} &= \theta_i^j \end{aligned} \qquad (17)$$

where r_i^j and θ_i^j are the radius and angle of the sample point for the i^{th} region in the j^{th} frame respectively. R_i^{j+1} and θ_i^{j+1} are the transformation results. a_i and b_i are the motion parameters for the i^{th} region. Because of the fan-like configuration of transducers in HATS, we constrain the transformation to be along the radius direction in Equation (17). The error of the affine transformation for the i^{th} region is:

$$err_i = \sum_{j=1}^{n-1} [(a_i r_i^j + b_i \theta_i) - r_i^{j+1}]^2 \qquad (18)$$

where r_i^{j+1} is the radius of the i^{th} sample point in the $(j+1)^{th}$ frame.

After getting the local motion parameters for every region by minimizing the error function in Equation (18), we created the second frame for the image pair by moving different regions in the reference frame with different affine motion. Since we know the contour position in the reference frame, we can calculate the contour positions for the synthetic frame according to the local affine transformation.

The pair of images with synthetic local motion(affine transformation) is showed in Figure 4. The contour of the synthetic frame has some manually created errors, which average to a 3.71 pixel difference from the ground truth. Figure 5 shows the result after we apply only the temporal constraints to the synthetic frame. The errors now average to only a 0.98 pixel difference from the ground truth.

5.2 Image sequence with real motion

We also tested the post processing with the medical image sequence in which real motion is presented. The post processing is based on the tracking results of tonTrak which is developed by Akgul [4]. The tonTrak system [4] works well for the tongue contour tracking problem. This system applies contour smoothness constraint in the temporal domain in addition to the traditional constraints applied in every single image frame. But due to the problems we talked about

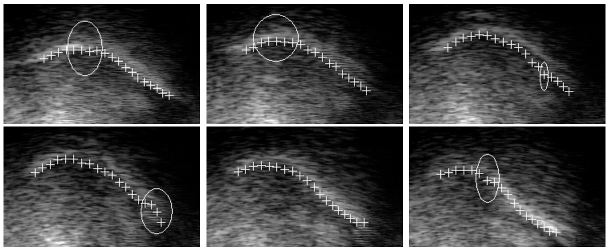

Figure 6: Tracking result before post processing. Circled contour points are incorrectly located at the tongue boundaries

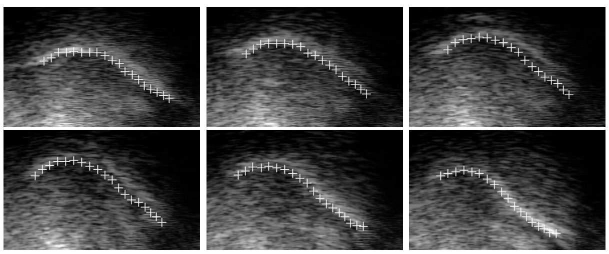

Figure 7: Tracking result after post processing. Incorrectly located contour points have been pushed to the tongue boundaries

in Section 2 and the noise of ultrasound images, there are still some contour points that are not tracked correctly. Figure 6 shows some frames of an ultrasound image sequence with the tracking results of tonTrak [4]. We circled contour points that were not in the correct positions. These points incorrectly converged before they reached the correct boundaries, due to the poor image quality. The image information in one single image was not enough to push these points anymore.

With more information from the tongue motion and the image correlation, our post processing can push these incorrectly located contour points to the tongue boundaries. Figure 7 shows the post processing results for image frames in Figure 6. The positions of these circled contour points have been corrected.

6 Conclusion

We divide the tracking problem into two steps: i) Contour detection using general internal and external energies in separate frames, and ii) Tracking and refining contours using motion information of the entire image sequence. A temporal internal energy and a temporal external energy are introduced in our framework.

Experiments performed on synthetic images and ultrasound image sequences showed that our framework works well and properly corrects positions of contour points. Based on this framework, we are now trying to develop a tracking framework which can fully utilize the information from the temporal domain and is insensitive to the initialization of contours.

References

[1] Y. Akgul and C. Kambhamettu. A new multi-level framework for deformable contour optimization. In *CVPR99*, pages II:465–470, 1999.

[2] Y. Akgul, C. Kambhamettu, and M. Stone. Extraction and tracking of the tongue surface from ultrasound image sequences. In *CVPR98*, pages 298–303, 1998.

[3] Y. Akgul, C. Kambhamettu, and M. Stone. Automatic Extraction and Tracking of The Tongue Contours. *IEEE Transactions on Medical Imaging*, 18(10):1035–1045, October. 1999.

[4] Y. Akgul, C. Kambhamettu, and M. Stone. A Task-Specific Contour Tracker for Ultrasound. *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis, Hilton Head Island, South CarolinaI*, pages 135–142, 2000.

[5] A. Amini, T. Weymouth, and R. Jain. Using dynamic programming for solving variational problems in vision. *PAMI*, 12(9):855–867, September 1990.

[6] V. Chalana and D. Linker. A Multiple Active Contour Model for Cardiac Boundary Detection on Echocardiographic Sequences. *IEEE Transactions on Medical Imaging*, 15(3):290–298, June. 1996.

[7] L. Cohen. On active contour models and balloons. *CVGIP*, 53(2):211–218, March 1991.

[8] L. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for 2-d and 3-d images. *PAMI*, 15(11):1131–1147, November 1993.

[9] S. Gunn and M. Nixon. Robust snake implementation: A dual active contour. *PAMI*, 19(1):63–68, January 1997.

[10] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *IJCV*, 1(4):321–331, January 1988.

[11] K. Lai. Deformable contours: Modeling, extraction, detection and classification. In *PH.D. Thesis Univ. of Wisconsin*, 1994.

[12] F. Leymarie and M. Levine. Tracking deformable objects in the plane using an active contour model. *PAMI*, 15(6):617–634, June 1993.

[13] N. Peterfreund. The velocity snake: Deformable contour for tracking in spatio-velocity space. *CVIU*, 73(3):346–356, March 1999.

[14] M. Stone and E. Davis. A head and transducer support system for making ultrasound images of tongue/jaw movement. *The Journal of The Acoustical Society of America*, 6:3107–3112, December. 1995.

[15] M. Wang, J. Evans, L. Hassebrook, and C. Knapp. A multistage, optimal active contour model. *PAMI*, 5(11):1586–1591, November 1996.