Segmentation and tracking of mitral valve leaflets in echocardiographic sequences: active contours guided by optical flow estimates

Mikic, Ivana, Krucinski, Slawomir, Thomas, James

Ivana Mikic, Slawomir Krucinski, James D. Thomas, "Segmentation and tracking of mitral valve leaflets in echocardiographic sequences: active contours guided by optical flow estimates," Proc. SPIE 2710, Medical Imaging 1996: Image Processing, (16 April 1996); doi: 10.1117/12.237934

SPIE.

Event: Medical Imaging 1996, 1996, Newport Beach, CA, United States

Segmentation and tracking of mitral valve leaflets in echocardiographic sequences: active contours guided by optical flow estimates

Ivana Mikić[1], Slawomir Krucinski[2], James D. Thomas[2]

Biomedical Engineering Center, The Ohio State University, Columbus, Ohio[1];
The Cleveland Clinic Foundation, Cleveland, Ohio[2]

ABSTRACT

In this paper we present a technique for detecting and tracking the mitral leaflet boundaries in echocardiographic sequences of images covering a complete cardiac cycle. The developed method is based on the active contour framework. This approach requires initial placement of the contour in vicinity of a feature of interest (a leaflet boundary in this case). Contour shape and position are then modified by the energy minimization process. Active contour can be used for tracking by selecting the optimal solution from the previous frame as an initial position in the present one. Such approach fails when displacements of the feature between two consecutive frames are too large as is the case with mitral valve leaflets. We propose a technique for tracking leaflets and other fast moving objects that takes advantage of the velocity information obtained by optical flow estimation. Singh's multiresolution optical flow estimation technique is used since it can estimate large velocities. The algorithm has been tested on several sequences from transesophageal echocardiography. The advantage of utilizing velocity information proves to be important in frames where leaflet motion is large. The contour inertia and stiffness suppress sporadic inaccuracies in velocity estimates which results in a well-behaved contour.

Keywords: active contours, snakes, optical flow, tracking, mitral valve, echocardiography

1. INTRODUCTION

Ultrasonic echocardiography has been recognized as a nontraumatic method for diagnosis and understanding of pathophysiology of heart valve diseases. A sequence of ultrasound images, acquired over a cardiac cycle, enables examination of cardiac structure and assessment of function. A common approach to the interpretation of the ultrasound sequence starts with manual or automated identification of feature points in images taken at two subsequent time frames. Feature points are extracted with some low level image processing algorithms and then the correspondence between feature points obtained at the different frames is established by a knowledge-based processing. A quantitative evaluation of information provided by ultrasound images requires more automated and objective segmentation and tracking technique. In the field of computer vision active contour models have been proposed as an interactive approach for locating features of interest in images. The active contour, a deformable curve, finds the position of minimal energy in the image and identifies feature of interest. In this paper we propose a modification of the active contour formulation that incorporates optical flow information as an estimate of initial velocities of the contour points in frame-to-frame motion. Although additional research is required to determine the optimal active contour parameters, we have demonstrated that this approach enables user-independent tracking of mitral valve leaflets in echocardiographic sequences. User input is reduced to the initial placement of the contour in the first frame of the sequence.

2. ACTIVE CONTOUR MODEL

Active contour models known also as snakes have been proposed by Kass et al.[1] The contour is defined as a continuous deformable curve $\mathbf{v}(s) = \langle x(s), y(s) \rangle$ that is attracted toward features of interest by image forces. The model assigns to the contour an energy functional that includes two components: internal and external energy terms (1).

$$E_{snake} = \int_0^1 \left(E_{\text{int}}(\mathbf{v}(s)) + E_{img}(\mathbf{v}(s)) \right) ds \tag{1}$$

E_{int} represents the internal energy of the contour due to bending and elongation/contraction and E_{img} is the energy associated with image forces. If the features of interest are edges, E_{img} can be defined as:

$$E_{img} = -|\nabla I| \tag{2}$$

Position and shape of the contour are modified by the energy minimization process, which means that snake is attracted to large image gradients (I is the image gray level). Model relies on initial placement of the contour in the vicinity of the desired feature, since the contour settles in local energy minima.

Snakes can also be interpreted[2] as contours composed of a deformable material, placed on the potential surface H, which is proportional to E_{img} (2). Regions with highest intensity gradients correspond to the valleys of H. Contour is assigned a weight, and through the action of gravitational forces, it is forced to fall down the slopes of the potential surface H. In the valley, contour reaches local energy minimum which results in locking onto the corresponding edge in the image.

A discrete contour model has been proposed by Miller[3]. In this model, the contour consists of a set of vertices, connected by line segments. The length of connecting segments defines the resolution of such discrete model. Internal forces are based on the local geometric constraints. In a similar model proposed by Lobregt and Viergever[4], the deformation process is performed in a number of discrete steps where acceleration, velocity and position are updated for each of the vertices. Position of a vertex and its neighbors determines internal and external forces that act on that vertex. This deformation process is given by the following equations:

$$\begin{aligned} \mathbf{X}_i(t+\Delta t) &= \mathbf{X}_i(t) + \dot{\mathbf{X}}_i(t)\Delta t \\ \dot{\mathbf{X}}_i(t+\Delta t) &= \dot{\mathbf{X}}_i(t) + \ddot{\mathbf{X}}_i(t)\Delta t \\ \ddot{\mathbf{X}}_i(t) &= \frac{1}{\mu_i}\mathbf{f}_i(t) \end{aligned} \tag{3}$$

where $\mathbf{X}_i(t) = \langle x_i(t), y_i(t) \rangle$, $\mathbf{f}_i$ and μ_i represent the position, the resultant of image and internal forces and the mass of the i-th vertex respectivelly and, t stands for physical time.

3. TRACKING ALGORITHM

Active contour models rely on a user's interaction in placing the contour close to the feature of interest. If initially placed too far, the contour can be out of reach of image forces or it can be attracted by some other, undesired image features. The snake model can be used for tracking if the deformations and displacements of the feature to be tracked are small enough. In such case, an optimal position from the previous frame can be used as an initial position in the present one. Obviously, this method is not applicable for large frame-to-frame displacements.

In the algorithm presented in this paper, optical flow velocity estimates are used to yield a better initial position of the snake in subsequent frames, limiting the user interaction to defining the initial contour only in the first frame of the sequence. The contour is defined as in[3,4] as a set of M vertices connected by line segments. The equation of motion for such contour model in matrix form is:

$$\mathbf{M}\ddot{\mathbf{X}}(t) - \mathbf{C}\dot{\mathbf{X}}(t) - \mathbf{K}\mathbf{X}(t) = \mathbf{F}(t) \tag{4}$$

where

$$\mathbf{X} = \begin{bmatrix} x_0 & y_0 \\ x_1 & y_1 \\ \cdots & \cdots \\ x_{M-1} & y_{M-1} \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} f_x^0 & f_y^0 \\ f_x^1 & f_y^1 \\ \cdots & \cdots \\ f_x^{M-1} & f_y^{M-1} \end{bmatrix}$$

$\mathbf{X}$ is the matrix of x and y vertex coordinates, $\mathbf{F}$ is the matrix of x and y components of image forces at each vertex, $\mathbf{M} = \mu \cdot \mathbf{I}_M$ where μ is the mass assigned to each vertex, $\mathbf{C} = \gamma \cdot \mathbf{I}_M$ where γ is the constant damping density and $\mathbf{K}$ is a MxM stiffness matrix.

To integrate the equation (4), two initial conditions are needed: $\mathbf{X}(0)$ and $\dot{\mathbf{X}}(0)$. For the first frame we use the user defined contour as $\mathbf{X}(0)$ and $\dot{\mathbf{X}}(0)$ is set to zero. In subsequent frames, $\mathbf{X}(0)$ is given by the optimal position of the contour in the previous frame, and $\dot{\mathbf{X}}(0)$ by the optical flow estimates. We use a single-step time marching scheme[5] (5) to solve for the optimal contour position. The time between two successive frames is assigned a value of 1 and divided into N_t intervals. In each time interval n, the contour position and velocity, $\mathbf{X}_{n+1}$ and $\dot{\mathbf{X}}_{n+1}$ are calculated and used as first guess for the following interval. Steps (i), (ii), (iii) are repeated for each of the intervals.

$$
\begin{aligned}
&\text{(i)} \quad \tilde{\mathbf{X}}_{n+1} = \mathbf{X}_n + \dot{\mathbf{X}}_n \Delta t \theta_1 \\
&\qquad \dot{\tilde{\mathbf{X}}}_{n+1} = \dot{\tilde{\mathbf{X}}}_n \\
&\text{(ii)} \quad \alpha_n = \left(\mathbf{M} + \Delta t \theta_1 \mathbf{C} + \frac{\Delta t^2}{2} \theta_2 \mathbf{K} \right)^{-1} \left(\mathbf{F} - \mathbf{C} \dot{\tilde{\mathbf{X}}}_{n+1} - \mathbf{K} \tilde{\mathbf{X}}_{n+1} \right) \\
&\text{(iii)} \quad \mathbf{X}_{n+1} = \mathbf{X}_n + \Delta t \dot{\mathbf{X}}_n + \alpha_n \frac{\Delta t^2}{2} \\
&\qquad \dot{\mathbf{X}}_{n+1} = \dot{\mathbf{X}}_n + \alpha_n \Delta t
\end{aligned}
\tag{5}
$$

In the first interval of each frame, $\mathbf{X}_0$ is the final contour position on the previous frame and $\dot{\mathbf{X}}_0$ is obtained from the optical flow estimates. The unconditional stability of the integration is guaranteed[5] if $\theta_1 \geq 0.5$ and $\theta_2 \geq \theta_1$. Results presented in this paper were obtained with $\theta_1 = \theta_2 = 1$.

The set of equations of motion (4) is nonlinear since both the image forces and stiffness matrix depend on the current position of the contour. For such problems, the solution is found iteratively in steps *(ii)* and *(iii)* of the procedure (5). Image forces and stiffness matrix are updated in each iteration. For this iterative procedure, an equilibrium criterion has to be introduced. As equilibrium error measure ε, we use the ratio of total deformation energy at current and first iteration (6), where total energy is a sum of energies at each vertex i.e.,

$$
\begin{aligned}
\varepsilon &= \frac{\sum_{i=0}^{M-1} d\mathbf{F}_i d\mathbf{X}_i}{\sum_{i=0}^{M-1} \left(d\mathbf{F}_i d\mathbf{X}_i \right)_0} \\
d\mathbf{F} &= \mathbf{M}\alpha + \mathbf{C}\dot{\mathbf{X}} + \mathbf{K}\mathbf{X} - \mathbf{F}
\end{aligned}
\tag{6}
$$

and $d\mathbf{X}_i$ is the position increment at current iteration. We assume that the contour has reached the equilibrium if error ε is below 1%.

To avoid the attraction by noise and other undesired image features that may be in the way before snake comes close to the new leaflet position, we chose a non-uniform time division. First interval Δt_0 is chosen to be 0.95 and the remaining 0.05 is divided into N_t equal intervals Δt_n. In calculations of $\mathbf{X}_1$ and $\dot{\mathbf{X}}_1$ (position and velocity after Δt_0) image forces are set to zero. This process is illustrated in Fig. 1: the beginning of the first interval Δt_0 is shown in (a), snake is in the position that corresponds to the optimal position from the previous frame; after Δt_0 contour is positioned close to the leaflet (b); after all steps are completed, the contour is adjusted to the new leaflet position as a result of action of image forces (c).

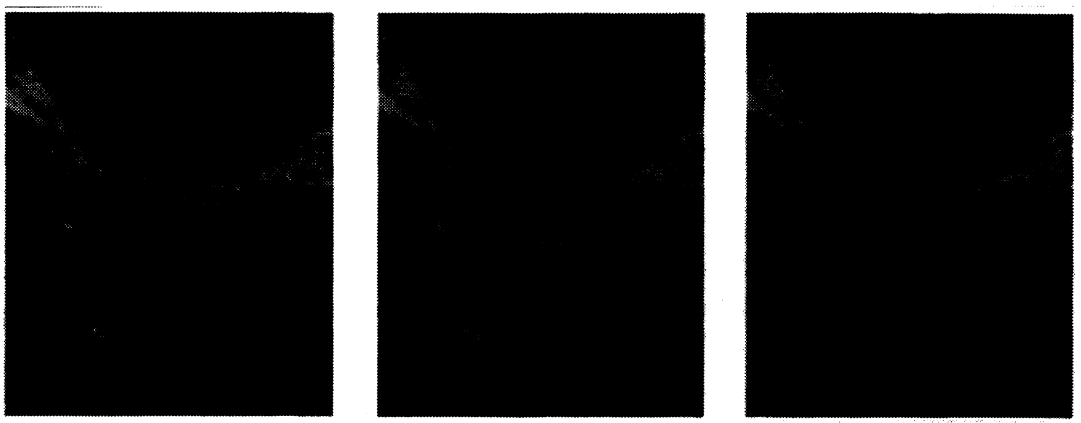


Figure 1. Stages in the tracking algorithm: (a) new frame with the optimal snake position from the previous frame shown, (b) after first time subinterval, contour is automatically placed close to the leaflet, (c) final position after the action of image forces

If velocity estimates are noisy and somewhat inaccurate, it would be desirable to further divide Δt_0 to increase the role of contour stiffness and inertia in filtering those estimates. We chose to preprocess velocity estimates instead, since our choice of stiffness matrix was somewhat limited as will be explained later.

3.1. Contour stiffness and internal forces

In the case of the discrete contour, internal forces depend on positions of vertices. Internal forces can be decomposed into the product of the stiffness matrix and the position matrix $\mathbf{F}_{int} = \mathbf{KX}$. In cases where this is not possible (as with internal forces used by Lobregt and Viergever[4]), these forces can always be added to image forces **F**. During the cardiac cycle, the appearance of the leaflet in ultrasound images changes drastically in shape and length. Assigning contour elastic properties that try to preserve these features results in failure to detect sudden changes especially in the leaflet length. The only leaflet shape feature that seems to be somewhat preserved throughout the ultrasound sequence is thickness. That lead us to introduce elastic forces that act to preserve leaflet thickness.

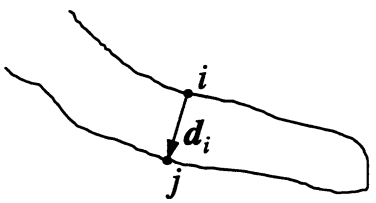


Figure 2. Internal force acts to preserve the length of each vector $\mathbf{d}_i$ throughout the sequence. Force at vertex *i* will be in the direction of $\mathbf{d}_i$ and its length (positive or negative) will depend on the degree of change in the length of $\mathbf{d}_i$

For each point *i* on the contour, the distance d_i from the matching point on the opposite side is calculated (Fig. 2). Also, the distance d_i^0 calculated at the optimal position in the first frame is stored and considered to be optimal. The internal force that acts on the vertex *i* is then calculated as an elastic force proportional to the relative change in length d_i:

$$
\begin{aligned}
\mathbf{f}_{\text{int}}^{i} &= k\frac{|\mathbf{d}_i| - |\mathbf{d}_i^0|}{|\mathbf{d}_i^0|}\frac{\mathbf{d}_i}{|\mathbf{d}_i|} \\
&= k\left(\frac{1}{d_i^0} - \frac{1}{d_i}\right)\mathbf{d}_i \\
&= k\left(\frac{1}{d_i^0} - \frac{1}{d_i}\right)\left(\mathbf{X}^j - \mathbf{X}^i\right) \\
&= k\left(\frac{1}{d_i^0} - \frac{1}{d_i}\right)\left\langle x_j - x_i, y_j - y_i\right\rangle
\end{aligned}
\tag{7}
$$

where k is the constant elastic coefficient. The form of eq. (7) indicates that in the i-th row of the stiffness matrix, all elements are zero, except $\mathbf{K}_{ii}$ and $\mathbf{K}_{ij}$. If we define $k_i = k\left(\frac{1}{d_i^0} - \frac{1}{d_i}\right)$, then $\mathbf{K}_{ii} = -k_i$ and $\mathbf{K}_{ij} = k_i$.

3.2. Image forces

As mentioned earlier, the active contour can be interpreted as a deformable curve placed on the potential surface H. Valleys of the surface H correspond to the edges in the original image. H is defined as:

$$
\mathbf{H}[i,j] = -\left|\nabla \mathbf{I}[i,j]\right| \tag{8}
$$

where I is the image intensity. Image forces pull the contour down the slopes of H towards the bottom of the valley and can, therefore, be calculated as the gradient of the potential surface. Components of the image force at each vertex are calculated as partial derivatives of H in corresponding directions:

$$
f_x^i = g\frac{\partial H[x_i, y_i]}{\partial x},\ f_y^i = g\frac{\partial H[x_i, y_i]}{\partial y} \tag{9}
$$

Sobel operator was used to calculate both H and F. Parameter g determines the amplitude of image forces. H is not blurred because of very small thickness of the leaflet, but for other applications, this operation would make the contour sensitive to more distant edges.

3.3. Velocity estimation

The optical flow field is defined as the distribution of pixel's velocities in an image sequence. We implemented three well-known optical flow estimation algorithms: Horn and Schunck[6,7], Lucas and Kanade[7,8] and Singh[9]. Singh's method gave best results on ultrasound images, it gives good confidence measures and importantly, allows a straightforward multiresolution extension, necessary for detection of high velocities present in ultrasound images of valve leaflets.

Singh's technique is based on the assumption that gray level distribution in the local neighborhood of each pixel is preserved throughout the sequence. The best match for the local pattern in one frame is found in the next frame, and the corresponding displacement is stored as the velocity of the central pixel. This calculation alone is sufficient for highly textured regions or sharp corners, but for regions with highly oriented intensity gradients multiple, equally good, matches for the local neighborhood of one pixel can be found in the succeeding frame (Figure 3). This is termed the aperture problem and to be resolved it requires additional constraints. An often used additional constraint is the smoothness of the velocity field.

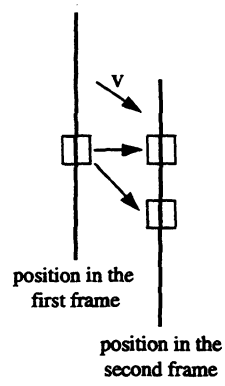


Figure 3. For a moving bar, multiple equally good matches for the local neighborhood of a pixel in on frame are found in the second frame. Component of velocity parallel to the line cannot be computed without additional information.

The algorithm consists of two parts. In the first part, velocity estimates are calculated using the image intensity information, and in the second part are further refined using the estimates obtained in the local neighborhood of each pixel.

First step: A (2n+1)×(2n+1) window W_p is formed around a pixel (x,y) in the first image and a (2N+1)×(2N+1) search window W_s is formed around the same location (x,y) in the second image. The similarity measures (10) between W_p and such window around pixels in W_s are calculated.

$$E_c(u,v)=\sum_{i=-n}^{n}\sum_{j=-n}^{n}\left(I(x+i,y+j,t)-I(x+u+i,y+v+j,t+1)\right)^2,\ -N\le u,v\le N \tag{10}$$

This similarity measure can be used to assign a probability distribution to each position in W_s representing the likelihood of the corresponding displacement. This distribution is defined as:

$$R_c(u,v)=e^{-aE_c(u,v)} \tag{11}$$

Using this approach, velocity estimate can be calculated as expected value of (u,v):

$$u_{cc}=\frac{\sum_u\sum_v R_c(u,v)u}{\sum_u\sum_v R_c(u,v)},\ v_{cc}=\frac{\sum_u\sum_v R_c(u,v)v}{\sum_u\sum_v R_c(u,v)},\ -N\le u,v\le N \tag{12}$$

and the corresponding covariance matrix S_{cc} is given by:

$$S_{cc}=\begin{bmatrix}\dfrac{\sum_u\sum_v R_c(u,v)(u-u_{cc})^2}{\sum_u\sum_v R_c(u,v)} & \dfrac{\sum_u\sum_v R_c(u,v)(u-u_{cc})(v-v_{cc})}{\sum_u\sum_v R_c(u,v)}\\ \dfrac{\sum_u\sum_v R_c(u,v)(u-u_{cc})(v-v_{cc})}{\sum_u\sum_v R_c(u,v)} & \dfrac{\sum_u\sum_v R_c(u,v)(v-v_{cc})^2}{\sum_u\sum_v R_c(u,v)}\end{bmatrix},-N\le u,v\le N \tag{13}$$

For window sizes we used n=1 and N=2, which means that maximal displacement that can be estimated is 2 pixels. For a, Singh suggests such number that maximum response R_c is close to 1. In our tests, this did not give satisfying results. Instead, we chose a such that minimum response is very small. This gives larger value of a and therefore provides better differentiation between more and less likely displacements.

Second step: In this part, a different "probability distribution" is introduced that gives rise to neighborhood information. In the (2w+1)×(2w+1) window W_n around the current pixel (w=1 is used), a Gaussian mask is applied, giving smaller weights to estimates that correspond to more distant pixels. These weights are denoted $R_n(u_i, v_i)$ where i is the pixel number in W_n. Using the same approach as in the first step, velocity estimates $(\overline{u}, \overline{v})$ and the corresponding covariance matrix S_n can be calculated.

Total squared error of a velocity estimate can be written as:

$$\varepsilon = (U - \overline{U})^T S_n^{-1} (U - \overline{U}) + (U - U_{cc})^T S_{cc} (U - U_{cc}) \tag{14}$$

Setting the gradient of this expression to zero and solving for U gives:

$$U = \left(S_{cc}^{-1} + S_n^{-1}\right)^{-1} \left(S_{cc}^{-1} U_{cc} + S_n^{-1} \overline{U}\right) \tag{15}$$

The solution is found using the Gauss-Seidel relaxation:

$$U^{k+1} = \left(S_{cc}^{-1} + S_n^{-1}\right)^{-1} \left(S_{cc}^{-1} U_{cc} + S_n^{-1} \overline{U}^k\right), \; U^0 = U_{cc} \tag{16}$$

Coarse-to-fine strategy: When velocities are known to be large, instead of using larger search windows, one can use multiscale approach. Velocities are first estimated at coarse resolution and at the next finer level, those estimates are used as a first guess. This means that instead of searching in the window around (x, y) in the second image, the search is performed around $(x+\tilde{u}, y+\tilde{v})$ where $\tilde{u}$ and $\tilde{v}$ are velocity estimates propagated from the coarser level. For these calculations we used both Gaussian and Laplacian multiscale pyramids[10]. In the sequences we tested, results obtained using Gaussian pyramid were slightly better. In each level of the pyramid the image is twice bigger than in previous level. Velocity estimated at one level is, therefore, multiplied by a factor 2 when propagated to the next finer level. Maximal velocity U_{max} that can be estimated at a particular level l is:

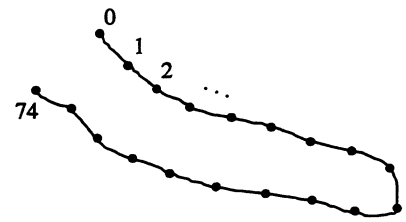
$$U_{\max}(l) = 2U_{\max}(l-1) + N \tag{17}$$

Where $U_{max}(0)=N$ is the size of the search window W_s

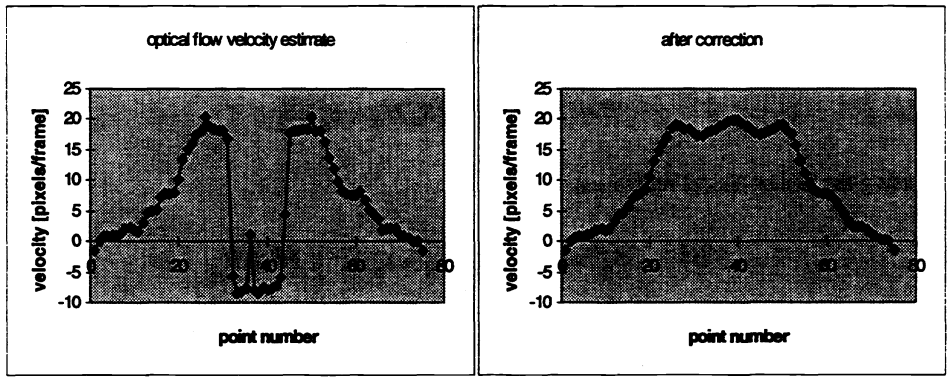
3.4. Velocity preprocessing

Even though we are using a fairly sophisticated optical flow estimation technique, velocities obtained are not always good enough and some corrections have to be made before they can be used for the tracking purposes. The main problem with the mitral valve leaflets occurs during valve opening. When it is closed, the tips of two leaflets overlap. Once the valve opens, tips move to opposite sides. Both these regions that are now moving in different directions belonged to the same region in the previous frame. Out of two different velocities, optical flow estimation algorithm chooses the one that is lower in intensity and for the case of the anterior leaflet that velocity is the wrong one. Fortunately, this only happens at the tip, therefore it is possible to calculate velocities of points at the tip from known values at other points. The motion of the leaflet can be approximated as a rotation around the axis placed at the valve annulus. In this case, velocities increase linearly as we move from the annulus toward the tip.

Velocities are preprocessed by calculating the current and average velocity derivative as we move down the leaflet. If a drastic change in the current derivative is found, that velocity value is discarded and new one is calculated from the previous increased by the average derivative. Snake is resampled at this time, so distance between neighboring points is constant. In Fig. 4b and 4c, y components of optical flow velocities are shown for contour points before and after this preprocessing. Points are numbered from 0 to 74 as in Fig. 4a.



(a)



(b) (c)

Figure 4. Velocity preprocessing results (same frame as in Fig. 1): (a) numbering of points on the contour; valve annulus is at points 0 and 74 and points around the middle belong to the tip, (b) y component of optical flow velocity estimates at each vertex, (c) results after preprocessing

4. RESULTS

The results presented in Fig. 5 were obtained with the following set of parameters:

- number of time intervals per frame N_t=5
- mass of a vertex μ=10^2
- damping factor γ=5
- elastic coefficient k=5
- image force coefficient g=10^4

All values were chosen empirically. The reason for such large value of g is that image force acts only for t=0.05, while internal and damping forces act for t=1 in each frame.

Fig. 5 shows a sequence of images, with the contour tracking the leaflet. Out of 21 frames present in this sequence, opening and closing are each shown only in two frames. This is due to a very fast leaflet motion. During these phases, displacements of points at the tip of the leaflet reach values of 35 pixels per frame, while the thickness of the leaflet itself is less than 5 pixels in the same region. Problems are sometimes occurring in the opening phase, where the contour does not fully extend to the tip of the leaflet.

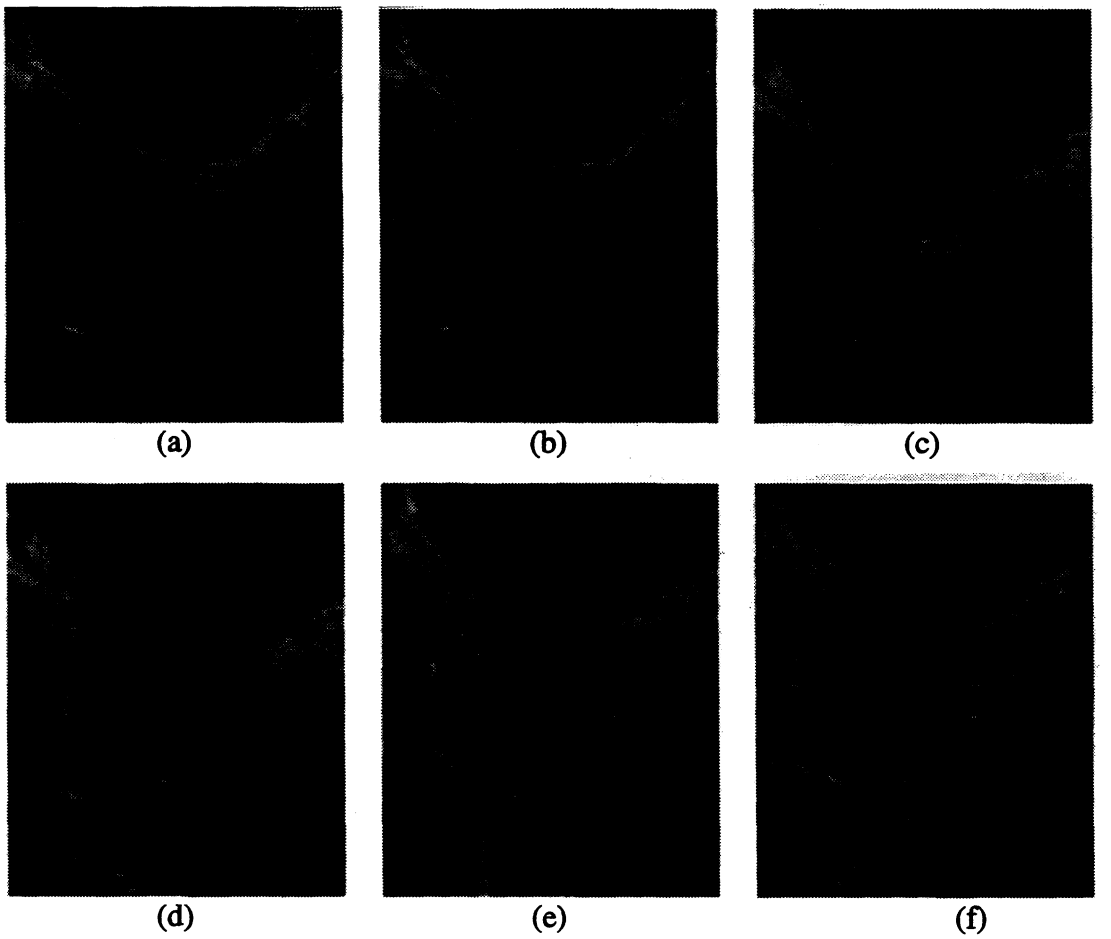


Figure 5. Images of the mitral valve leaflets from one transesophageal echocardiographic sequence with active contour tracking the leaflet edges. Results were obtained using 5 time steps per frame. At each step, several iterations are performed until the equilibrium criterion is met. (a) user-defined contour, (b) final solution in the first frame, (c)-(f) results of automatic tracking

5. CONCLUSIONS AND FUTURE DIRECTIONS

We have presented a discrete active contour model able to track fast moving objects. The model has been developed for echocardiographic sequences, but can be used in various other applications. Results obtained so far are encouraging. It is important to emphasize that leaflet motion is sometimes extremely rapid (frame-to-frame displacements are sometimes an order of magnitude larger than the leaflet thickness). Also, the shape of the leaflet changes significantly during the cardiac cycle. Only better results should be expected for more rigid motion or somewhat smaller velocities.

The algorithm is computationally intensive and we plan further efforts to increase its speed. Testing of the influence of parameter settings on snake behavior could reveal better guidelines for choice of the parameter values. Final goal of this project is extension of presented contour model to three dimensions, which would enable more complete analysis of cardiac structures and provide a powerful image analysis tool.

[1] M. Kass, A. Witkin and D. Terzopoulos, "Snakes: active contour models", *Proc. International Conference on Computer Vision*, pp. 259-268, London, 1987.
[2] F. Leymarie and M. D. Levine, "Tracking deformable objects in the plane using an active contour model", *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no 6, pp. 617-634, 1993.
[3] J. V. Miller, D. E. Breen, W. E. Lorensen, R. M. O'Bara and M. J. Wozny, "Geometrically deformed models: a method for extracting closed geometric models from volume data", *Computer Graphics,* vol. 25, no 4, pp. 217-226, 1991.
[4] S. Lobregt and M. A. Viergever, "A discrete dynamic contour model", *IEEE Trans. Medical Imaging,* vol. 14, no 1, pp. 12-24, 1995.
[5] O. C. Zienkiewicz, W. L. Wood and N. W. Hine, "A unified set of single step algorithms", *International Journal for Numerical Methods in Engineering,* vol. 20, pp. 1529-1552, 1984.
[6] B. K. P. Horn and B. G. Schunck, "Determining optical flow", *Artificial Intelligence,* vol. 17, pp. 185-203, 1981.
[7] J. L. Barron, D. J. Fleet, S. S. Beauchemin and T. A. Burkitt, "Performance of optical flow techniques", *International Journal of Computer Vision,* 12:1, pp. 43-77, 1994.
[8] E. P. Simoncelli, E. H. Andelson, D. J. Heeger, "Probability distributions of optical flow", *IEEE Proc. CVPR,* pp. 310-315, 1991.
[9] A. Singh, *Optic Flow Computation: A Unified Perspective,* pp. 1-85, IEEE Computer Society Press, Los Alamitos, 1991.
[10] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code", *IEEE Trans. Communications,* vol. com-31, no 4, pp. 532-540, 1983.